Snowflake.ARA-R01.v2024-10-29.q82

Exam Code:	ARA-R01
Exam Name:	SnowPro Advanced: Architect Recertification Exam
Certification Provider:	Snowflake
Free Question Number:	82
Version:	v2024-10-29
# of views:	602
# of Questions views:	820
https://www.freeqas.com/qa/Snowflake/ARA-R01/Snowflake.ARA-R01.v2024-10-29.q82.html	

NEW QUESTION: 1
What is a valid object hierarchy when building a Snowflake environment?
A. Account --> Database --> Schema --> Warehouse
B. Organization --> Account --> Database --> Schema --> Stage
C. Account --> Schema > Table --> Stage
D. Organization --> Account --> Stage --> Table --> View
Answer: (SHOW ANSWER)
This is the valid object hierarchy when building a Snowflake environment, according to the Snowflake documentation and the web search results. Snowflake is a cloud data platform that supports various types of objects, such as databases, schemas, tables, views, stages, warehouses, and more. These objects are organized in a hierarchical structure, as follows:
Organization: An organization is the top-level entity that represents a group of Snowflake accounts that are related by business needs or ownership. An organization can have one or more accounts, and can enable features such as cross-account data sharing, billing and usage reporting, and single sign-on across accounts12.
Account: An account is the primary entity that represents a Snowflake customer. An account can have one or more databases, schemas, stages, warehouses, and other objects. An account can also have one or more users, roles, and security integrations. An account is associated with a specific cloud platform, region, and Snowflake edition34.
Database: A database is a logical grouping of schemas. A database can have one or more schemas, and can store structured, semi-structured, or unstructured data. A database can also have properties such as retention time, encryption, and ownership56.
Schema: A schema is a logical grouping of tables, views, stages, and other objects. A schema can have one or more objects, and can define the namespace and access control for the objects. A schema can also have properties such as ownership and default warehouse .

Stage: A stage is a named location that references the files in external or internal storage. A stage can be used to load data into Snowflake tables using the COPY INTO command, or to unload data from Snowflake tables using the COPY INTO LOCATION command. A stage can be created at the account, database, or schema level, and can have properties such as file format, encryption, and credentials .
The other options listed are not valid object hierarchies, because they either omit or misplace some objects in the structure. For example, option A omits the organization level and places the warehouse under the schema level, which is incorrect. Option C omits the organization, account, and stage levels, and places the table under the schema level, which is incorrect. Option D omits the database level and places the stage and table under the account level, which is incorrect.
References:
Snowflake Documentation: Organizations
Snowflake Blog: Introducing Organizations in Snowflake
Snowflake Documentation: Accounts
Snowflake Blog: Understanding Snowflake Account Structures
Snowflake Documentation: Databases
Snowflake Blog: How to Create a Database in Snowflake
[Snowflake Documentation: Schemas]
[Snowflake Blog: How to Create a Schema in Snowflake]
[Snowflake Documentation: Stages]
[Snowflake Blog: How to Use Stages in Snowflake]

NEW QUESTION: 2
What built-in Snowflake features make use of the change tracking metadata for a table? (Choose two.)
A. The MERGE command
B. The UPSERT command
C. The CHANGES clause
D. A STREAM object
E. The CHANGE_DATA_CAPTURE command
Answer: C,D (LEAVE A REPLY)
The built-in Snowflake features that make use of the change tracking metadata for a table are the CHANGES clause and a STREAM object. The CHANGES clause enables querying the change tracking metadata for a table or view within a specified interval of time without having to create a stream with an explicit transactional offset1. A STREAM object records data manipulation language (DML) changes made to tables, including inserts, updates, and deletes, as well as metadata about each change, so that actions can be taken using the changed data. This process is referred to as change data capture (CDC)2. The other options are incorrect because they do not make use of the change tracking metadata for a table. The MERGE command performs insert, update, or delete operations on a target

table based on the results of a join with a source table3. The UPSERT command is not a valid Snowflake command. The CHANGE_DATA_CAPTURE command is not a valid Snowflake command. References: CHANGES | Snowflake Documentation, Change Tracking Using Table Streams | Snowflake Documentation, MERGE | Snowflake Documentation

NEW QUESTION: 3
Is it possible for a data provider account with a Snowflake Business Critical edition to share data with an Enterprise edition data consumer account?
A. A Business Critical account cannot be a data sharing provider to an Enterprise consumer. Any consumer accounts must also be Business Critical.
B. If a user in the provider account with role authority to create or alter share adds an Enterprise account as a consumer, it can import the share.
C. If a user in the provider account with a share owning role sets share_restrictions to False when adding an Enterprise consumer account, it can import the share.
D. If a user in the provider account with a share owning role which also has override share restrictions privilege share_restrictions set to False when adding an Enterprise consumer account, it can import the share.
Answer: B (LEAVE A REPLY)
In Snowflake, data sharing capabilities allow a Business Critical edition account to share data with an Enterprise edition consumer account. The ability to share data is contingent upon the role permissions within the provider account. If a user has the necessary role authority (like ACCOUNTADMIN or a role with similar privileges to create or manage shares), they can add an Enterprise edition account as a consumer. This feature enables flexibility in data sharing across different Snowflake account editions, facilitating broader data collaboration and accessibility.References: Snowflake's data sharing documentation and the specifics of edition-based capabilities discussed in SnowPro Advanced: Architect certification materials.

NEW QUESTION: 4
Files arrive in an external stage every 10 seconds from a proprietary system. The files range in size from 500 K to 3 MB. The data must be accessible by dashboards as soon as it arrives.
How can a Snowflake Architect meet this requirement with the LEAST amount of coding? (Choose two.)
A. Use Snowpipe with auto-ingest.
B. Use a COPY command with a task.
C. Use a materialized view on an external table.
D. Use the COPY INTO command.
E. Use a combination of a task and a stream.
Answer: A,E (LEAVE A REPLY)

The requirement is for the data to be accessible as quickly as possible after it arrives in the external stage with minimal coding effort.
Option A: Snowpipe with auto-ingest is a service that continuously loads data as it arrives in the stage. With auto-ingest, Snowpipe automatically detects new files as they arrive in a cloud stage and loads the data into the specified Snowflake table with minimal delay and no intervention required. This is an ideal low-maintenance solution for the given scenario where files are arriving at a very high frequency.
Option E: Using a combination of a task and a stream allows for real-time change data capture in Snowflake.
A stream records changes (inserts, updates, and deletes) made to a table, and a task can be scheduled to trigger on a very short interval, ensuring that changes are processed into the dashboard tables as they occur.

NEW QUESTION: 5
Which data models can be used when modeling tables in a Snowflake environment? (Select THREE).
A. Graph model
B. Dimensional/Kimball
C. Data lake
D. Inmon/3NF
E. Bayesian hierarchical model
F. Data vault
Answer: (SHOW ANSWER)
Snowflake is a cloud data platform that supports various data models for modeling tables in a Snowflake environment. The data models can be classified into two categories: dimensional and normalized. Dimensional data models are designed to optimize query performance and ease of use for business intelligence and analytics. Normalized data models are designed to reduce data redundancy and ensure data integrity for transactional and operational systems. The following are some of the data models that can be used in Snowflake:
* Dimensional/Kimball: This is a popular dimensional data model that uses a star or snowflake schema to organize data into fact and dimension tables. Fact tables store quantitative measures and foreign keys to dimension tables. Dimension tables store descriptive attributes and hierarchies. A star schema has a single denormalized dimension table for each dimension, while a snowflake schema has multiple normalized dimension tables for each dimension. Snowflake supports both star and snowflake schemas, and allows users to create views and joins to simplify queries.
* Inmon/3NF: This is a common normalized data model that uses a third normal form (3NF) schema to organize data into entities and relationships. 3NF schema eliminates data duplication and ensures data consistency by applying three rules: 1) every column in a table must depend on the primary key, 2) every column in a table must depend on the

whole primary key, not a part of it, and 3) every column in a table must depend only on the primary key, not on other columns. Snowflake supports 3NF schema and allows users to create referential integrity constraints and foreign key relationships to enforce data quality.
* Data vault: This is a hybrid data model that combines the best practices of dimensional and normalized data models to create a scalable, flexible, and resilient data warehouse. Data vault schema consists of three types of tables: hubs, links, and satellites. Hubs store business keys and metadata for each entity.
Links store associations and relationships between entities. Satellites store descriptive attributes and historical changes for each entity or relationship. Snowflake supports data vault schema and allows users to leverage its features such as time travel, zero-copy cloning, and secure data sharing to implement data vault methodology.
References: What is Data Modeling? | Snowflake, Snowflake Schema in Data Warehouse Model - GeeksforGeeks, [Data Vault 2.0 Modeling with Snowflake]

NEW QUESTION: 6
A Snowflake Architect created a new data share and would like to verify that only specific records in secure views are visible within the data share by the consumers.
What is the recommended way to validate data accessibility by the consumers?
A. Create reader accounts as shown below and impersonate the consumers by logging in with their credentials.
create managed account reader_acctl admin_name = userl , adroin_password 'Sdfed43da! 44T , type = reader;
B. Create a row access policy as shown below and assign it to the data share.
create or replace row access policy rap_acct as (acct_id varchar) returns boolean -> case when
'acctl_role' = current_role() then true else false end;
C. Set the session parameter called SIMULATED_DATA_SHARING_C0NSUMER as shown below in order to impersonate the consumer accounts.
alter session set simulated_data_sharing_consumer - 'Consumer Acctl*
D. Alter the share settings as shown below, in order to impersonate a specific consumer account.
alter share sales share set accounts = 'Consumerl' share restrictions = true
Answer: (SHOW ANSWER)
The SIMULATED_DATA_SHARING_CONSUMER session parameter allows a data provider to simulate the data access of a consumer account without creating a reader account or logging in with the consumer credentials. This parameter can be used to validate the data accessibility by the consumers in a data share, especially when using secure viewsor secure UDFs that filter data based on the current account or role. By setting this parameter to the name of a consumer account, the data provider can see the same data as the consumer would see when querying the shared database. This is a

convenient and efficient way to test the data sharing functionality and ensure that only the intended data is visible to the consumers.
References:
Using the SIMULATED_DATA_SHARING_CONSUMER Session Parameter
SnowPro Advanced: Architect Exam Study Guide

NEW QUESTION: 7
Consider the following COPY command which is loading data with CSV format into a Snowflake table from an internal stage through a data transformation query.
This command results in the following error:
SQL compilation error: invalid parameter 'validation_mode'
Assuming the syntax is correct, what is the cause of this error?
A. The VALIDATION_MODE parameter supports COPY statements that load data from external stages only.
B. The VALIDATION_MODE parameter does not support COPY statements with CSV file formats.
C. The VALIDATION_MODE parameter does not support COPY statements that transform data during a load.
D. The value return_all_errors of the option VALIDATION_MODE is causing a compilation error.
Answer: C (LEAVE A REPLY)
The VALIDATION_MODE parameter is used to specify the behavior of the COPY statement when loading data into a table. It is used to specify whether the COPY statement should return an error if any of the rows in the file are invalid or if it should continue loading the valid rows. The VALIDATION_MODE parameter is only supported for COPY statements that load data from external stages1.
The query in the question uses a data transformation query to load data from an internal stage. A data transformation query is a query that transforms the data during the load process, such as parsing JSON or XML data, applying functions, or joining with other tables2.
According to the documentation, VALIDATION_MODE does not support COPY statements that transform data during a load. If the parameter is specified, the COPY statement returns an error1.
Therefore, option C is the correct answer.
References: : COPY INTO <table> : Transforming Data During a Load

NEW QUESTION: 8
A healthcare company wants to share data with a medical institute. The institute is running a Standard edition of Snowflake; the healthcare company is running a Business Critical edition.
How can this data be shared?

A. The healthcare company will need to change the institute's Snowflake edition in the accounts panel.
B. By default, sharing is supported from a Business Critical Snowflake edition to a Standard edition.
C. Contact Snowflake and they will execute the share request for the healthcare company.
D. Set the share_restriction parameter on the shared object to false.
Answer: D (LEAVE A REPLY)
By default, Snowflake does not allow sharing data from a Business Critical edition to a non-Business Critical edition. This is because Business Critical edition provides enhanced security and data protection features that are not available in lower editions. However, this restriction can be overridden by setting the share_restriction parameter on the shared object (database, schema, or table) to false. This parameter allows the data provider to explicitly allow sharing data with lower edition accounts. Note that this parameter can only be set by the data provider, not the data consumer. Also, setting this parameter to false may reduce the level of security and data protection for the shared data.
References:
* Enable Data Share:Business Critical Account to Lower Edition
* Sharing Is Not Allowed From An Account on BUSINESS CRITICAL Edition to an Account On A Lower Edition
* SQL Execution Error: Sharing is Not Allowed from an Account on BUSINESS CRITICAL Edition to an Account on a Lower Edition
* Snowflake Editions | Snowflake Documentation

NEW QUESTION: 9
Which Snowflake data modeling approach is designed for BI queries?
A. 3 NF
B. Star schema
C. Data Vault
D. Snowflake schema
Answer: B (LEAVE A REPLY)
A star schema is a Snowflake data modeling approach that is designed for BI queries. A star schema is a type of dimensional modeling that organizes data into fact tables and dimension tables. A fact table contains the measures or metrics of the business process, such as sales amount, order quantity, or profit margin. A dimension table contains the attributes or descriptors of the business process, such as product name, customer name, or order date. A star schema is called so because it resembles a star, with one fact table in the center and multiple dimension tables radiating from it. A star schema can improve the performance and simplicity of BI queries by reducing the number of joins, providing fast access to aggregated data, and enabling intuitive query syntax. A star schema can also support various types of analysis, such as trend analysis, slice and dice, drill down, and roll up12.

References:
Snowflake Documentation: Dimensional Modeling
Snowflake Documentation: Star Schema

NEW QUESTION: 10
An Architect runs the following SQL query:
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.
Answer: (SHOW ANSWER)
* 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

NEW QUESTION: 11
A table contains five columns and it has millions of records. The cardinality distribution of the columns is shown below:
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
Answer: C (LEAVE A REPLY)
According to the Snowflake documentation, the following are some considerations for choosing clustering for a table1:
Clustering is optimal when either:

You require the fastest possible response times, regardless of cost.
Your improved query performance offsets the credits required to cluster and maintain the table.
Clustering is most effective when the clustering key is used in the following types of query predicates:
Filter predicates (e.g. WHERE clauses)
Join predicates (e.g. ON clauses)
Grouping predicates (e.g. GROUP BY clauses)
Sorting predicates (e.g. ORDER BY clauses)
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

NEW QUESTION: 12
A Data Engineer is designing a near real-time ingestion pipeline for a retail company to ingest event logs into Snowflake to derive insights. A Snowflake Architect is asked to define security best practices to configure access control privileges for the data load for auto-ingest to Snowpipe.
What are the MINIMUM object privileges required for the Snowpipe user to execute Snowpipe?
A. OWNERSHIP on the named pipe, USAGE on the named stage, target database, and schema, and INSERT and SELECT on the target table
B. OWNERSHIP on the named pipe, USAGE and READ on the named stage, USAGE on the target database and schema, and INSERT end SELECT on the target table
C. CREATE on the named pipe, USAGE and READ on the named stage, USAGE on the target database and schema, and INSERT end SELECT on the target table

D. USAGE on the named pipe, named stage, target database, and schema, and INSERT and SELECT on the target table

Answer: B (LEAVE A REPLY)

According to the SnowPro Advanced: Architect documents and learning resources, the minimum object privileges required for the Snowpipe user to execute Snowpipe are:

* OWNERSHIP on the named pipe. This privilege allows the Snowpipe user to create, modify, and drop the pipe object that defines the COPY statement for loading data from the stage to the table1.
* USAGE and READ on the named stage. These privileges allow the Snowpipe user to access and read the data files from the stage that are loaded by Snowpipe2.
* USAGE on the target database and schema. These privileges allow the Snowpipe user to access the database and schema that contain the target table3.
* INSERT and SELECT on the target table. These privileges allow the Snowpipe user to insert data into the table and select data from the table4.

The other options are incorrect because they do not specify the minimum object privileges required for the Snowpipe user to execute Snowpipe. Option A is incorrect because it does not include the READ privilege on the named stage, which is required for the Snowpipe user to read the data files from the stage. Option C is incorrect because it does not include the OWNERSHIP privilege on the named pipe, which is required for the Snowpipe user to create, modify, and drop the pipe object. Option D is incorrect because it does not include the OWNERSHIP privilege on the named pipe or the READ privilege on the named stage, which are both required for the Snowpipe user to execute Snowpipe. References: CREATE PIPE | Snowflake Documentation, CREATE STAGE | Snowflake Documentation, CREATE DATABASE | Snowflake Documentation, CREATE TABLE | Snowflake Documentation

NEW QUESTION: 13

Based on the architecture in the image, how can the data from DB1 be copied into TBL2? (Select TWO).

A.

B.

C.

D.

E.

Answer: B,E (LEAVE A REPLY)

* The architecture in the image shows a Snowflake data platform with two databases, DB1 and DB2, and two schemas, SH1 and SH2. DB1 contains a table TBL1 and a stage STAGE1. DB2 contains a table TBL2. The image also shows a snippet of code written in SQL language that copies data from STAGE1 to TBL2 using a file format FF PIPE 1.
* To copy data from DB1 to TBL2, there are two possible options among the choices given:

* Option B: Use a named external stage that references STAGE1. This option requires creating an external stage object in DB2.SH2 that points to the same location as STAGE1 in DB1.SH1. The external stage can be created using the CREATE STAGE command with the URL parameter specifying the location of STAGE11. For example:
SQLAI-generated code. Review and use carefully. More info on FAQ.
use database DB2;
use schema SH2;
create stage EXT_STAGE1
url = @DB1.SH1.STAGE1;
* Then, the data can be copied from the external stage to TBL2 using the COPY INTO command with the FROM parameter specifying the external stage name and the FILE FORMAT parameter specifying the file format name2. For example:
SQLAI-generated code. Review and use carefully. More info on FAQ.
copy into TBL2
from @EXT_STAGE1
file format = (format name = DB1.SH1.FF PIPE 1);
* Option E: Use a cross-database query to select data from TBL1 and insert into TBL2. This option requires using the INSERT INTO command with the SELECT clause to query data from TBL1 in DB1.SH1 and insert it into TBL2 in DB2.SH2. The query must use the fully-qualified names of the tables, including the database and schema names3. For example:
SQLAI-generated code. Review and use carefully. More info on FAQ.
use database DB2;
use schema SH2;
insert into TBL2
select * from DB1.SH1.TBL1;
* The other options are not valid because:
* Option A: It uses an invalid syntax for the COPY INTO command. The FROM parameter cannot specify a table name, only a stage name or a file location2.
* Option C: It uses an invalid syntax for the COPY INTO command. The FILE FORMAT parameter cannot specify a stage name, only a file format name or options2.
* Option D: It uses an invalid syntax for the CREATE STAGE command. The URL parameter cannot specify a table name, only a file location1.
References:
* 1: CREATE STAGE | Snowflake Documentation
* 2: COPY INTO table | Snowflake Documentation
* 3: Cross-database Queries | Snowflake Documentation

NEW QUESTION: 14
Is it possible for a data provider account with a Snowflake Business Critical edition to share data with an Enterprise edition data consumer account?

A. A Business Critical account cannot be a data sharing provider to an Enterprise consumer. Any consumer accounts must also be Business Critical.
B. If a user in the provider account with role authority to create or alter share adds an Enterprise account as a consumer, it can import the share.
C. If a user in the provider account with a share owning role sets share_restrictions to False when adding an Enterprise consumer account, it can import the share.
D. If a user in the provider account with a share owning role which also has override share restrictions privilege share_restrictions set to False when adding an Enterprise consumer account, it can import the share.
Answer: (SHOW ANSWER)
Data sharing is a feature that allows Snowflake accounts to share data with each other without the need for data movement or copying1. Data sharing is enabled by creating shares, which are collections of database objects (tables, views, secure views, and secure UDFs) that can be accessed by other accounts, called consumers2.
By default, Snowflake does not allow sharing data from a Business Critical edition account to a non-Business Critical edition account. This is because Business Critical edition offers higher levels of data protection and encryption than other editions, and sharing data with lower editions may compromise the security and compliance of the data3.
However, Snowflake provides the OVERRIDE SHARE RESTRICTIONS global privilege, which allows a user to override the default restriction and share data from a Business Critical edition account to a non-Business Critical edition account. This privilege is granted to the ACCOUNTADMIN role by default, and can be granted to other roles as well4.
To enable data sharing from a Business Critical edition account to an Enterprise edition account, the following steps are required34:
A user in the provider account with the OVERRIDE SHARE RESTRICTIONS privilege must create or alter a share and add the Enterprise edition account as a consumer. The user must also set the share_restrictions parameter to False when adding the consumer. This parameter indicates whether the share is restricted to Business Critical edition accounts only. Setting it to False allows the share to be imported by lower edition accounts.
A user in the consumer account with the IMPORT SHARE privilege must import the share and grant access to the share objects to other roles in the account. The user must also set the share_restrictions parameter to False when importing the share. This parameter indicates whether the consumer account accepts shares from Business Critical edition accounts only. Setting it to False allows the consumer account to import shares from lower edition accounts.
References:
1: Introduction to Secure Data Sharing | Snowflake Documentation
2: Creating Secure Data Shares | Snowflake Documentation
3: Enable Data Share:Business Critical Account to Lower Edition | Medium

4: Enabling sharing from a Business critical account to a non-business ... | Snowflake Documentation

NEW QUESTION: 15
An Architect Is designing a data lake with Snowflake. The company has structured, semi-structured, and unstructured data. The company wants to save the data inside the data lake within the Snowflake system. The company is planning on sharing data among Its corporate branches using Snowflake data sharing.
What should be considered when sharing the unstructured data within Snowflake?
A. A pre-signed URL should be used to save the unstructured data into Snowflake in order to share data over secure views, with no time limit for the URL.
B. A scoped URL should be used to save the unstructured data into Snowflake in order to share data over secure views, with a 24-hour time limit for the URL.
C. A file URL should be used to save the unstructured data into Snowflake in order to share data over secure views, with a 7-day time limit for the URL.
D. A file URL should be used to save the unstructured data into Snowflake in order to share data over secure views, with the "expiration_time" argument defined for the URL time limit.
Answer: (SHOW ANSWER)
According to the Snowflake documentation, unstructured data files can be shared by using a secure view and Secure Data Sharing. A secure view allows the result of a query to be accessed like a table, and a secure view is specifically designated for data privacy. A scoped URL is an encoded URL that permits temporary access to a staged file without granting privileges to the stage. The URL expires when the persisted query result period ends, which is currently 24 hours. A scoped URL is recommended for file administrators to give scoped access to data files to specific roles in the same account. Snowflake records information in the query history about who uses a scoped URL to access a file, and when. Therefore, a scoped URL is the best option to share unstructured data within Snowflake, as it provides security, accountability, and control over the data access. References:
Sharing unstructured Data with a secure view
Introduction to Loading Unstructured Data

NEW QUESTION: 16
Role A has the following permissions:
. USAGE on db1
. USAGE and CREATE VIEW on schemal in db1
. SELECT on tablel in schemal
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 schemal;
create view v1 as select * from db2.schema2.table2;
B. use database db2;
use schema schema2;
create view v2 as select * from dbl.schemal. tablel;
C. use database db2;
use schema schema2;
select * from db1.schemal.tablel union select * from table2;
D. use database db1;
use schema schemal;
select * from db2.schema2.table2;
Answer: (SHOW ANSWER)
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.schemal.table1 with Role B.
Since Role A, which has SELECT permission on db1.schemal.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.schemal.table1 to create the view. Snowflake's security model requires that the active role have all necessary permissions to execute the command.

NEW QUESTION: 17
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 tables exceed the 1 TB limit for data recovery.
D. The staging tables are of the TRANSIENT type.
E. The DevOps role should be granted ALLOW_RECOVERY privilege on the staging schema.
Answer: B,D (LEAVE A REPLY)
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 numberof 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: 18
A retail company has over 3000 stores all using the same Point of Sale (POS) system. The company wants to deliver near real-time sales results to category managers. The stores

operate in a variety of time zones and exhibit a dynamic range of transactions each minute, with some stores having higher sales volumes than others.
Sales results are provided in a uniform fashion using data engineered fields that will be calculated in a complex data pipeline. Calculations include exceptions, aggregations, and scoring using external functions interfaced to scoring algorithms. The source data for aggregations has over 100M rows.
Every minute, the POS sends all sales transactions files to a cloud storage location with a naming convention that includes store numbers and timestamps to identify the set of transactions contained in the files. The files are typically less than 10MB in size.
How can the near real-time results be provided to the category managers? (Select TWO).
A. All files should be concatenated before ingestion into Snowflake to avoid micro-ingestion.
B. A Snowpipe should be created and configured with AUTO_INGEST = true. A stream should be created to process INSERTS into a single target table using the stream metadata to inform the store number and timestamps.
C. A stream should be created to accumulate the near real-time data and a task should be created that runs at a frequency that matches the real-time analytics needs.
D. An external scheduler should examine the contents of the cloud storage location and issue SnowSQL commands to process the data at a frequency that matches the real-time analytics needs.
E. The copy into command with a task scheduled to run every second should be used to achieve the near-real time requirement.
Answer: B,C (LEAVE A REPLY)
To provide near real-time sales results to category managers, the Architect can use the following steps:
* Create an external stage that references the cloud storage location where the POS sends the sales transactions files. The external stage should use the file format and encryption settings that match the source files2
* Create a Snowpipe that loads the files from the external stage into a target table in Snowflake. The Snowpipe should be configured with AUTO_INGEST = true, which means that it will automatically detect and ingest new files as they arrive in the external stage. The Snowpipe should also use a copy option to purge the files from the external stage after loading, to avoid duplicate ingestion3
* Create a stream on the target table that captures the INSERTS made by the Snowpipe. The stream should include the metadata columns that provide information about the file name, path, size, and last modified time. The stream should also have a retention period that matches the real-time analytics needs4
* Create a task that runs a query on the stream to process the near real-time data. The query should use the stream metadata to extract the store number and timestamps from the file name and path, and perform the calculations for exceptions, aggregations, and scoring using external functions. The query should also output the results to another table

or view that can be accessed by the category managers. The task should be scheduled to run at a frequency that matches the real-time analytics needs, such as every minute or every 5 minutes.
The other options are not optimal or feasible for providing near real-time results:
* All files should be concatenated before ingestion into Snowflake to avoid micro-ingestion. This option is not recommended because it would introduce additional latency and complexity in the data pipeline.
Concatenating files would require an external process or service that monitors the cloud storage location and performs the file merging operation. This would delay the ingestion of new files into Snowflake and increase the risk of data loss or corruption. Moreover, concatenating files would not avoid micro-ingestion, as Snowpipe would still ingest each concatenated file as a separate load.
* An external scheduler should examine the contents of the cloud storage location and issue SnowSQL commands to process the data at a frequency that matches the real-time analytics needs. This option is not necessary because Snowpipe can automatically ingest new files from the external stage without requiring an external trigger or scheduler. Using an external scheduler would add more overhead and dependency to the data pipeline, and it would not guarantee near real-time ingestion, as it would depend on the polling interval and the availability of the external scheduler.
* The copy into command with a task scheduled to run every second should be used to achieve the near-real time requirement. This option is not feasible because tasks cannot be scheduled to run every second in Snowflake. The minimum interval for tasks is one minute, and even that is not guaranteed, as tasks are subject to scheduling delays and concurrency limits. Moreover, using the copy into command with a task would not leverage the benefits of Snowpipe, such as automatic file detection, load balancing, and micro-partition optimization. References:
* 1: SnowPro Advanced: Architect | Study Guide
* 2: Snowflake Documentation | Creating Stages
* 3: Snowflake Documentation | Loading Data Using Snowpipe
* 4: Snowflake Documentation | Using Streams and Tasks for ELT
* : Snowflake Documentation | Creating Tasks
* : Snowflake Documentation | Best Practices for Loading Data
* : Snowflake Documentation | Using the Snowpipe REST API
* : Snowflake Documentation | Scheduling Tasks
* : SnowPro Advanced: Architect | Study Guide
* : Creating Stages
* : Loading Data Using Snowpipe
* : Using Streams and Tasks for ELT
* : [Creating Tasks]
* : [Best Practices for Loading Data]
* : [Using the Snowpipe REST API]

* : [Scheduling Tasks]

NEW QUESTION: 19
Which query will identify the specific days and virtual warehouses that would benefit from a multi-cluster warehouse to improve the performance of a particular workload?
A.
B.
C.
D.
Answer: B (LEAVE A REPLY)
The correct answer is option B. This query is designed to assess the need for a multi-cluster warehouse by examining the queuing time (AVG_QUEUED_LOAD) on different days and virtual warehouses. When the AVG_QUEUED_LOAD is greater than zero, it suggests that queries are waiting for resources, which can be an indicator that performance might be improved by using a multi-cluster warehouse to handle the workload more efficiently. By grouping by date and warehouse name and filtering on the sum of the average queued load being greater than zero, the query identifies specific days and warehouses where the workload exceeded the available compute resources. This information is valuable when considering scaling out warehouses to multi-cluster configurations for improved performance.

NEW QUESTION: 20
An Architect is designing a solution that will be used to process changed records in an orders table.
Newly-inserted orders must be loaded into the f_orders fact table, which will aggregate all the orders by multiple dimensions (time, region, channel, etc.). Existing orders can be updated by the sales department within 30 days after the order creation. In case of an order update, the solution must perform two actions:
1. Update the order in the f_0RDERS fact table.
2. Load the changed order data into the special table ORDER _REPAIRS.
This table is used by the Accounting department once a month. If the order has been changed, the Accounting team needs to know the latest details and perform the necessary actions based on the data in the order_repairs table.
What data processing logic design will be the MOST performant?
A. Useone stream and one task.
B. Useone stream and two tasks.
C. Usetwo streams and one task.
D. Usetwo streams and two tasks.
Answer: B (LEAVE A REPLY)
The most performant design for processing changed records, considering the need to both update records in the f_orders fact table and load changes into the order_repairs table, is

to use one stream and two tasks. The stream will monitor changes in the orders table, capturing both inserts and updates. The first task would apply these changes to the f_orders fact table, ensuring all dimensions are accurately represented. The second task would use the same stream to insert relevant changes into the order_repairs table, which is critical for the Accounting department's monthly review. This method ensures efficient processing by minimizing the overhead of managing multiple streams and synchronizing between them, while also allowing specific tasks to optimize for their target operations.References: Snowflake's documentation on streams and tasks for handling data changes efficiently.

NEW QUESTION: 21
How does a standard virtual warehouse policy work in Snowflake?
A. It conserves credits by keeping running clusters fully loaded rather than starting additional clusters.
B. It starts only if the system estimates that there is a query load that will keep the cluster busy for at least 6 minutes.
C. It starts only f the system estimates that there is a query load that will keep the cluster busy for at least 2 minutes.
D. It prevents or minimizes queuing by starting additional clusters instead of conserving credits.
Answer: D (LEAVE A REPLY)
A standard virtual warehouse policy is one of the two scaling policies available for multi-cluster warehouses in Snowflake. The other policy is economic. A standard policy aims to prevent or minimize queuing by starting additional clusters as soon as the current cluster is fully loaded, regardless of the number of queries in the queue. This policy can improve query performance and concurrency, but it may also consume more credits than an economic policy, which tries to conserve credits by keeping the running clusters fully loaded before starting additional clusters. The scaling policy can be set when creating or modifying a warehouse, and it can be changed at any time.
References:
Snowflake Documentation: Multi-cluster Warehouses
Snowflake Documentation: Scaling Policy for Multi-cluster Warehouses

NEW QUESTION: 22
A company's client application supports multiple authentication methods, and is using Okta.
What is the best practice recommendation for the order of priority when applications authenticate to Snowflake?
A. 1) OAuth (either Snowflake OAuth or External OAuth)
2) External browser
3) Okta native authentication

4) Key Pair Authentication, mostly used for service account users
5) Password
B. 1) External browser, SSO
2) Key Pair Authentication, mostly used for development environment users
3) Okta native authentication
4) OAuth (ether Snowflake OAuth or External OAuth)
5) Password
C. 1) Okta native authentication
2) Key Pair Authentication, mostly used for production environment users
3) Password
4) OAuth (either Snowflake OAuth or External OAuth)
5) External browser, SSO
D. 1) Password
2) Key Pair Authentication, mostly used for production environment users
3) Okta native authentication
4) OAuth (either Snowflake OAuth or External OAuth)
5) External browser, SSO
Answer: (SHOW ANSWER)
This is the best practice recommendation for the order of priority when applications authenticate to Snowflake, according to the Snowflake documentation and the web search results. Authentication is the process of verifying the identity of a user or application that connects to Snowflake. Snowflake supports multiple authentication methods, each with different advantages and disadvantages. The recommended order of priority is based on the following factors:
Security: The authentication method should provide a high level of security and protection against unauthorized access or data breaches. The authentication method should also support multi-factor authentication (MFA) or single sign-on (SSO) for additional security.
Convenience: The authentication method should provide a smooth and easy user experience, without requiring complex or manual steps. The authentication method should also support seamless integration with external identity providers or applications.
Flexibility: The authentication method should provide a range of options and features to suit different use cases and scenarios. The authentication method should also support customization and configuration to meet specific requirements.
Based on these factors, the recommended order of priority is:
OAuth (either Snowflake OAuth or External OAuth): OAuth is an open standard for authorization that allows applications to access Snowflake resources on behalf of a user, without exposing the user's credentials. OAuth provides a high level of security, convenience, and flexibility, as it supports MFA, SSO, token-based authentication, and various grant types and scopes. OAuth can be implemented using either Snowflake OAuth or External OAuth, depending on the identity provider and the application12.

External browser: External browser is an authentication method that allows users to log in to Snowflake using a web browser and an external identity provider, such as Okta, Azure AD, or Ping Identity.
External browser provides a high level of security and convenience, as it supports MFA, SSO, and federated authentication. External browser also provides a consistent user interface and experience across different platforms and devices34.
Okta native authentication: Okta native authentication is an authentication method that allows users to log in to Snowflake using Okta as the identity provider, without using a web browser. Okta native authentication provides a high level of security and convenience, as it supports MFA, SSO, and federated authentication. Okta native authentication also provides a native user interface and experience for Okta users, and supports various Okta features, such as password policies and user management56.
Key Pair Authentication: Key Pair Authentication is an authentication method that allows users to log in to Snowflake using a public-private key pair, without using a password. Key Pair Authentication provides a high level of security, as it relies on asymmetric encryption and digital signatures. Key Pair Authentication also provides a flexible and customizable authentication option, as it supports various key formats, algorithms, and expiration times. Key Pair Authentication is mostly used for service account users, such as applications or scripts that connect to Snowflake programmatically7 .
Password: Password is the simplest and most basic authentication method that allows users to log in to Snowflake using a username and password. Password provides a low level of security, as it relies on symmetric encryption and is vulnerable to brute force attacks or phishing. Password also provides a low level of convenience and flexibility, as it requires manual input and management, and does not support MFA or SSO. Password is the least recommended authentication method, and should be used only as a last resort or for testing purposes .
References:
Snowflake Documentation: Snowflake OAuth
Snowflake Documentation: External OAuth
Snowflake Documentation: External Browser Authentication
Snowflake Blog: How to Use External Browser Authentication with Snowflake Snowflake Documentation: Okta Native Authentication Snowflake Blog: How to Use Okta Native Authentication with Snowflake Snowflake Documentation: Key Pair Authentication
[Snowflake Blog: How to Use Key Pair Authentication with Snowflake]
[Snowflake Documentation: Password Authentication]
[Snowflake Blog: How to Use Password Authentication with Snowflake]

NEW QUESTION: 23
Consider the following scenario where a masking policy is applied on the CREDICARDND column of the CREDITCARDINFO table. The masking policy definition Is as follows:
Sample data for the CREDITCARDINFO table is as follows:

NAME EXPIRYDATE CREDITCARDNO
JOHN DOE 2022-07-23 4321 5678 9012 1234
if the Snowflake system rotes have not been granted any additional roles, what will be the result?
A. The sysadmin can see the CREDICARDND column data in clear text.
B. The owner of the table will see the CREDICARDND column data in clear text.
C. Anyone with the PI_ANALYTICS role will see the last 4 characters of the CREDICARDND column data in dear text.
D. Anyone with the PI_ANALYTICS role will see the CREDICARDND column as*** 'MASKED* **'.
Answer: (SHOW ANSWER)
* The masking policy defined in the image indicates that if a user has the PI_ANALYTICS role, they will be able to see the last 4 characters of the CREDITCARDNO column data in clear text. Otherwise, they will see 'MASKED'. Since Snowflake system roles have not been granted any additional roles, they won't have the PI_ANALYTICS role and therefore cannot view the last 4 characters of credit card numbers.
* To apply a masking policy on a column in Snowflake, you need to use the ALTER TABLE ... ALTER COLUMN command or the ALTER VIEW command and specify the policy name. For example, to apply the creditcardno_mask policy on the CREDITCARDNO column of the CREDITCARDINFO table, you can use the following command:
ALTER TABLE CREDITCARDINFO ALTER COLUMN CREDITCARDNO SET MASKING POLICY creditcardno_mask;
* For more information on how to create and use masking policies in Snowflake, you can refer to the following resources:
CREATE MASKING POLICY: This document explains the syntax and usage of the CREATE MASKING POLICY command, which allows you to create a new masking policy or replace an existing one.
Using Dynamic Data Masking: This guide provides instructions on how to configure and use dynamic data masking in Snowflake, which is a feature that allows you to mask sensitive data based on the execution context of the user.
ALTER MASKING POLICY: This document explains the syntax and usage of the ALTER MASKING POLICY command, which allows you to modify the properties of an existing masking policy.
References: 1: https://docs.snowflake.com/en/sql-reference/sql/create-masking-policy 2: https://docs.snowflake.com/en/user-guide/security-column-ddm-use 3: https://docs.snowflake.com/en/sql-reference/sql/alter-masking-policy

NEW QUESTION: 24
In a managed access schema, what are characteristics of the roles that can manage object privileges? (Select TWO).

A. Users with the SYSADMIN role can grant object privileges in a managed access schema.
B. Users with the SECURITYADMIN role or higher, can grant object privileges in a managed access schema.
C. Users who are database owners can grant object privileges in a managed access schema.
D. Users who are schema owners can grant object privileges in a managed access schema.
E. Users who are object owners can grant object privileges in a managed access schema.
Answer: B,D (LEAVE A REPLY)
In a managed access schema, the privilege management is centralized with the schema owner, who has the authority to grant object privileges within the schema. Additionally, the SECURITYADMIN role has the capability to manage object grants globally, which includes within managed access schemas. Other roles, such as SYSADMIN or database owners, do not inherently have this privilege unless explicitly granted.
References: The verified answers are based on Snowflake's official documentation, which outlines the roles and privileges associated with managed access schemas12.

NEW QUESTION: 25
An Architect is designing a file ingestion recovery solution. The project will use an internal named stage for file storage. Currently, in the case of an ingestion failure, the Operations team must manually download the failed file and check for errors.
Which downloading method should the Architect recommend that requires the LEAST amount of operational overhead?
A. Use the Snowflake Connector for Python, connect to remote storage and download the file.
B. Use the get command in SnowSQL to retrieve the file.
C. Use the get command in Snowsight to retrieve the file.
D. Use the Snowflake API endpoint and download the file.
Answer: B (LEAVE A REPLY)
The get command in SnowSQL is a convenient way to download files from an internal stage to a local directory. The get command can be used in interactive mode or in a script, and it supports wildcards and parallel downloads. The get command also allows specifying the overwrite option, which determines how to handle existing files with the same name2
The Snowflake Connector for Python, the Snowflake API endpoint, and the get command in Snowsight are not recommended methods for downloading files from an internal stage, because they require more operational overhead than the get command in SnowSQL. The Snowflake Connector for Python and the Snowflake API endpoint require writing and maintaining code to handle the connection, authentication, and file transfer. The get command in Snowsight requires using the web interface and manually selecting the files to download34 References:

1: SnowPro Advanced: Architect | Study Guide
2: Snowflake Documentation | Using the GET Command
3: Snowflake Documentation | Using the Snowflake Connector for Python
4: Snowflake Documentation | Using the Snowflake API
: Snowflake Documentation | Using the GET Command in Snowsight
: SnowPro Advanced: Architect | Study Guide
: Using the GET Command
: Using the Snowflake Connector for Python
: Using the Snowflake API
: [Using the GET Command in Snowsight]

NEW QUESTION: 26
The following table exists in the production database:
A regulatory requirement states that the company must mask the username for events that are older than six months based on the current date when the data is queried.
How can the requirement be met without duplicating the event data and making sure it is applied when creating views using the table or cloning the table?
A. Use a masking policy on the username column using a entitlement table with valid dates.
B. Use a row level policy on the user_events table using a entitlement table with valid dates.
C. Use a masking policy on the username column with event_timestamp as a conditional column.
D. Use a secure view on the user_events table using a case statement on the username column.
Answer: (SHOW ANSWER)
A masking policy is a feature of Snowflake that allows masking sensitive data in query results based on the role of the user and the condition of the data. A masking policy can be applied to a column in a table or a view, and it can use another column in the same table or view as a conditional column. A conditional column is a column that determines whether the masking policy is applied or not based on its value1.
In this case, the requirement can be met by using a masking policy on the username column with event_timestamp as a conditional column. The masking policy can use a function that masks the username if the event_timestamp is older than six months based on the current date, and returns the original username otherwise. The masking policy can be applied to the user_events table, and it will also be applied when creating views using the table or cloning the table2.
The other options are not correct because:
A). Using a masking policy on the username column using an entitlement table with valid dates would require creating another table that stores the valid dates for each username, and joining it with the user_events table in the masking policy function. This would add

complexity and overhead to the masking policy, and it would not use the event_timestamp column as the condition for masking.
B). Using a row level policy on the user_events table using an entitlement table with valid dates would require creating another table that stores the valid dates for each username, and joining it with the user_events table in the row access policy function. This would filter out the rows that have event_timestamp older than six months based on the valid dates, instead of masking the username column. This would not meet the requirement of masking the username, and it would also reduce the visibility of the event data.
D). Using a secure view on the user_events table using a case statement on the username column would require creating a view that uses a case expression to mask the username column based on the event_timestamp column. This would meet the requirement of masking the username, but it would not be applied when cloning the table. A secure view is a view that prevents the underlying data from being exposed by queries on the view. However, a secure view does not prevent the underlying data from being exposed by cloning the table3.
References:
1: Masking Policies | Snowflake Documentation
2: Using Conditional Columns in Masking Policies | Snowflake Documentation
3: Secure Views | Snowflake Documentation

NEW QUESTION: 27
A company needs to have the following features available in its Snowflake account:
1. Support for Multi-Factor Authentication (MFA)
2. A minimum of 2 months of Time Travel availability
3. Database replication in between different regions
4. Native support for JDBC and ODBC
5. Customer-managed encryption keys using Tri-Secret Secure
6. Support for Payment Card Industry Data Security Standards (PCI DSS)
In order to provide all the listed services, what is the MINIMUM Snowflake edition that should be selected during account creation?
A. Standard
B. Enterprise
C. Business Critical
D. Virtual Private Snowflake (VPS)
Answer: C (LEAVE A REPLY)
According to the Snowflake documentation1, the Business Critical edition offers the following features that are relevant to the question:
* Support for Multi-Factor Authentication (MFA): This is a standard feature available in all Snowflake editions1.
* A minimum of 2 months of Time Travel availability: This is an enterprise feature that allows users to

* access historical data for up to 90 days1.
* Database replication in between different regions: This is an enterprise feature that enables users to replicate databases across different regions or cloud platforms1.
* Native support for JDBC and ODBC: This is a standard feature available in all Snowflake editions1.
* Customer-managed encryption keys using Tri-Secret Secure: This is a business critical feature that provides enhanced security and data protection by allowing customers to manage their own encryption keys1.
* Support for Payment Card Industry Data Security Standards (PCI DSS): This is a business critical feature that ensures compliance with PCI DSS regulations for handling sensitive cardholder data1.
Therefore, the minimum Snowflake edition that should be selected during account creation to provide all the listed services is the Business Critical edition.
References:
* Snowflake Editions | Snowflake Documentation

NEW QUESTION: 28
What is a characteristic of Role-Based Access Control (RBAC) as used in Snowflake?
A. Privileges can be granted at the database level and can be inherited by all underlying objects.
B. A user can use a "super-user" access along with securityadmin to bypass authorization checks and access all databases, schemas, and underlying objects.
C. A user can create managed access schemas to support future grants and ensure only schema owners can grant privileges to other roles.
D. A user can create managed access schemas to support current and future grants and ensure only object owners can grant privileges to other roles.
Answer: (SHOW ANSWER)
Role-Based Access Control (RBAC) is the Snowflake Access Control Framework that allows privileges to be granted by object owners to roles, and roles, in turn, can be assigned to users to restrict or allow actions to be performed on objects. A characteristic of RBAC as used in Snowflake is:
* Privileges can be granted at the database level and can be inherited by all underlying objects. This means that a role that has a certain privilege on a database, such as CREATE SCHEMA or USAGE, can also perform the same action on any schema, table, view, or other object within that database, unless explicitly revoked. This simplifies the access control management and reduces the number of grants required.
* A user can create managed access schemas to support future grants and ensure only schema owners can grant privileges to other roles. This means that a user can create a schema with the MANAGED ACCESS option, which changes the default behavior of object ownership and privilege granting within the schema. In a managed access schema, object owners lose the ability to grant privileges on their objects to other roles, and only the

schema owner or a role with the MANAGE GRANTS privilege can do so. This enhances the security and governance of the schema and its objects.
The other options are not characteristics of RBAC as used in Snowflake:
* A user can use a "super-user" access along with securityadmin to bypass authorization checks and access all databases, schemas, and underlying objects. This is not true, as there is no such thing as a
"super-user" access in Snowflake. The securityadmin role is a predefined role that can manage users and
* roles, but it does not have any privileges on any database objects by default. To access any object, the securityadmin role must be explicitly granted the appropriate privilege by the object owner or another role with the grant option.
* A user can create managed access schemas to support current and future grants and ensure only object owners can grant privileges to other roles. This is not true, as this contradicts the definition of a managed access schema. In a managed access schema, object owners cannot grant privileges on their objects to other roles, and only the schema owner or a role with the MANAGE GRANTS privilege can do so.
References:
* Overview of Access Control
* A Functional Approach For Snowflake's Role-Based Access Controls
* Snowflake Role-Based Access Control simplified
* Snowflake RBAC security prefers role inheritance to role composition
* Overview of Snowflake Role Based Access Control

NEW QUESTION: 29
A table for IOT devices that measures water usage is created. The table quickly becomes large and contains more than 2 billion rows.
The general query patterns for the table are:
1. DeviceId, IOT_timestamp and CustomerId are frequently used in the filter predicate for the select statement
2. The columns City and DeviceManuf acturer are often retrieved
3. There is often a count on UniqueId
Which field(s) should be used for the clustering key?
A. IOT_timestamp
B. City and DeviceManuf acturer
C. DeviceId and CustomerId
D. UniqueId
Answer: C (LEAVE A REPLY)
A clustering key is a subset of columns or expressions that are used to co-locate the data in the same micro-partitions, which are the units of storage in Snowflake. Clustering can improve the performance of queries that filter on the clustering key columns, as it reduces the amount of data that needs to be scanned. The best choice for a clustering key depends

on the query patterns and the data distribution in the table. In this case, the columns DeviceId, IOT_timestamp, and CustomerId are frequently used in the filter predicate for the select statement, which means they are good candidates for the clustering key. The columns City and DeviceManufacturer are often retrieved, but not filtered on, so they are not as important for the clustering key.
The column UniqueId is used for counting, but it is not a good choice for the clustering key, as it is likely to have a high cardinality and a uniform distribution, which means it will not help to co-locate the data.
Therefore, the best option is to use DeviceId and CustomerId as the clustering key, as they can help to prune the micro-partitions and speed up thequeries. References: Clustering Keys & Clustered Tables, Micro-partitions & Data Clustering, A Complete Guide to Snowflake Clustering

NEW QUESTION: 30
A company has several sites in different regions from which the company wants to ingest data.
Which of the following will enable this type of data ingestion?
A. The company must have a Snowflake account in each cloud region to be able to ingest data to that account.
B. The company must replicate data between Snowflake accounts.
C. The company should provision a reader account to each site and ingest the data through the reader accounts.
D. The company should use a storage integration for the external stage.
Answer: D (LEAVE A REPLY)
This is the correct answer because it allows the company to ingest data from different regions using a storage integration for the external stage. A storage integration is a feature that enables secure and easy access to files in external cloud storage from Snowflake. A storage integration can be used to create an external stage, which is a named location that references the files in the external storage. An external stage can be used to load data into Snowflake tables using the COPY INTO command, or to unload data from Snowflake tables using the COPY INTO LOCATION command. A storage integration can support multiple regions and cloud platforms, as long as the external storage service is compatible with Snowflake12.
References:
Snowflake Documentation: Storage Integrations
Snowflake Documentation: External Stages

NEW QUESTION: 31
What built-in Snowflake features make use of the change tracking metadata for a table? (Choose two.)
A. The MERGE command

B. The UPSERT command
C. The CHANGES clause
D. A STREAM object
E. The CHANGE_DATA_CAPTURE command
Answer: A,D (LEAVE A REPLY)
In Snowflake, the change tracking metadata for a table is utilized by the MERGE command and the STREAM object. The MERGE command uses change tracking to determine how to apply updates and inserts efficiently based on differences between source and target tables. STREAM objects, on the other hand, specifically capture and store change data, enabling incremental processing based on changes made to a table since the last stream offset was committed.References: Snowflake Documentation on MERGE and STREAM Objects.

NEW QUESTION: 32
Which feature provides the capability to define an alternate cluster key for a table with an existing cluster key?
A. External table
B. Materialized view
C. Search optimization
D. Result cache
Answer: B (LEAVE A REPLY)
A materialized view is a feature that provides the capability to define an alternate cluster key for a table with an existing cluster key. A materialized view is a pre-computed result set that is stored in Snowflake and can be queried like a regular table. A materialized view can have a different cluster key than the base table, which can improve the performance and efficiency of queries on the materialized view. A materialized view can also support aggregations, joins, and filters on the base table data. A materialized view is automatically refreshed when the underlying data in the base table changes, as long as the AUTO_REFRESH parameter is set to true1.
References:
Materialized Views | Snowflake Documentation

NEW QUESTION: 33
A healthcare company wants to share data with a medical institute. The institute is running a Standard edition of Snowflake; the healthcare company is running a Business Critical edition.
How can this data be shared?
A. The healthcare company will need to change the institute's Snowflake edition in the accounts panel.
B. By default, sharing is supported from a Business Critical Snowflake edition to a Standard edition.
C. Contact Snowflake and they will execute the share request for the healthcare company.
D. Set the share_restriction parameter on the shared object to false.
Answer: D (LEAVE A REPLY)
By default, Snowflake does not allow sharing data from a Business Critical edition to a non-Business Critical edition. This is because Business Critical edition provides enhanced security and data protection features that are not available in lower editions. However, this restriction can be overridden by setting the share_restriction parameter on the shared object (database, schema, or table) to false. This parameter allows the data provider to explicitly allow sharing data with lower edition accounts. Note that this parameter can only be set by the data provider, not the data consumer. Also, setting this parameter to false may reduce the level of security and data protection for the shared data.
References:
Enable Data Share:Business Critical Account to Lower Edition
Sharing Is Not Allowed From An Account on BUSINESS CRITICAL Edition to an Account On A Lower Edition SQL Execution Error: Sharing is Not Allowed from an Account on BUSINESS CRITICAL Edition to an Account on a Lower Edition Snowflake Editions | Snowflake Documentation

NEW QUESTION: 34
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 root level (HSM)
B. At the account level (AMK)
C. At the table level (TMK)
D. At the micro-partition level
Answer: B (LEAVE A REPLY)
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: 35
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 current_role targeting the role in use for the current session.
B. Set masking policy conditions using is_role_in_session targeting the role in use for the current account.
C. Set masking policy conditions using invoker_role targeting the executing role in a SQL statement.
D. Determine if there are ownership privileges on the masking policy that would allow the use of any function.
E. Assign the accountadmin role to the user who is executing the object.
Answer: A,C (LEAVE A REPLY)
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: 36
What step will im the performance of queries executed against an external table?
A. Partition the external table.
B. Shorten the names of the source files.
C. Convert the source files' character encoding to UTF-8.
D. Use an internal stage instead of an external stage to store the source files.
Answer: (SHOW ANSWER)
Partitioning an external table is a technique that improves the performance of queries executed against the table by reducing the amount of data scanned. Partitioning an external table involves creating one or more partition columns that define how the table is logically divided into subsets of data based on the values in those columns. The partition columns can be derived from the file metadata (such as file name, path, size, or modification time) or from the file content (such as a column value or a JSON attribute). Partitioning an external table allows the query optimizer to prune the files that do not match the query predicates, thus avoiding unnecessary data scanning and processing2 The other

options are not effective steps for improving the performance of queries executed against an external table:
* Shorten the names of the source files. This option does not have any impact on the query performance, as the file names are not used for query processing. The file names are only used for creating the external table and displaying the query results3
* Convert the source files' character encoding to UTF-8. This option does not affect the query performance, as Snowflake supports various character encodings for external table files, such as UTF-8, UTF-16, UTF-32, ISO-8859-1, and Windows-1252. Snowflake automatically detects the character encoding of the files and converts them to UTF-8 internally for query processing4
* Use an internal stage instead of an external stage to store the source files. This option is not applicable, as external tables can only reference files stored in external stages, such as Amazon S3, Google Cloud Storage, or Azure Blob Storage. Internal stages are used for loading data into internal tables, not external tables5 References:
* 1: SnowPro Advanced: Architect | Study Guide
* 2: Snowflake Documentation | Partitioning External Tables
* 3: Snowflake Documentation | Creating External Tables
* 4: Snowflake Documentation | Supported File Formats and Compression for Staged Data Files
* 5: Snowflake Documentation | Overview of Stages
* : SnowPro Advanced: Architect | Study Guide
* : Partitioning External Tables
* : Creating External Tables
* : Supported File Formats and Compression for Staged Data Files
* : Overview of Stages

NEW QUESTION: 37
A user is executing the following command sequentially within a timeframe of 10 minutes from start to finish:
What would be the output of this query?
A. Table T_SALES_CLONE successfully created.
B. Time Travel data is not available for table T_SALES.
C. The offset -> is not a valid clause in the clone operation.
D. Syntax error line 1 at position 58 unexpected 'at'.
Answer: A (LEAVE A REPLY)
The query is executing a clone operation on an existing table t_sales with an offset to account for the retention time. The syntax used is correct for cloning a table in Snowflake, and the use of the at(offset => -60*30) clause is valid. This specifies that the clone should be based on the state of the table 30 minutes prior (60 seconds * 30). Assuming the table t_sales exists and has been modified within the last 30 minutes, and considering the data_retention_time_in_days is set to 1 day (which enables time travel queries for the past

24 hours), the table t_sales_clone would be successfully created based on the state of t_sales 30 minutes before the clone command was issued.

NEW QUESTION: 38
An Architect has a design where files arrive every 10 minutes and are loaded into a primary database table using Snowpipe. A secondary database is refreshed every hour with the latest data from the primary database.
Based on this scenario, what Time Travel query options are available on the secondary database?
A. A query using Time Travel in the secondary database is available for every hourly table version within the retention window.
B. A query using Time Travel in the secondary database is available for every hourly table version within and outside the retention window.
C. Using Time Travel, secondary database users can query every iterative version within each hour (the individual Snowpipe loads) in the retention window.
D. Using Time Travel, secondary database users can query every iterative version within each hour (the individual Snowpipe loads) and outside the retention window.
Answer: A (LEAVE A REPLY)
Snowflake's Time Travel feature allows users to query historical data within a defined retention period. In the given scenario, since the secondary database is refreshed every hour, Time Travel can be used to query each hourly version of the table as long as it falls within the retention window. This does not include individual Snowpipe loads within each hour unless they coincide with the hourly refresh.
References: The answer is verified using Snowflake's official documentation, which provides detailed information on Time Travel and its usage within the retention period123.

NEW QUESTION: 39
What are characteristics of the use of transactions in Snowflake? (Select TWO).
A. Explicit transactions can contain DDL, DML, and query statements.
B. The autocommit setting can be changed inside a stored procedure.
C. A transaction can be started explicitly by executing a begin work statement and end explicitly by executing a commit work statement.
D. A transaction can be started explicitly by executing a begin transaction statement and end explicitly by executing an end transaction statement.
E. Explicit transactions should contain only DML statements and query statements. All DDL statements implicitly commit active transactions.
Answer: A,D (LEAVE A REPLY)
In Snowflake, a transaction is a sequence of SQL statements that are processed as an atomic unit. All statements in the transaction are either applied (i.e. committed) or undone (i.e. rolled back) together.

Snowflake transactions guarantee ACID properties. A transaction can include both reads and writes1.
Explicit transactions are transactions that are started and ended explicitly by using the BEGIN TRANSACTION, COMMIT, and ROLLBACK statements. Snowflake supports the synonyms BEGIN WORK and BEGIN TRANSACTION, and COMMIT WORK and ROLLBACK WORK. Explicit transactions can contain DDL, DML, and query statements. However, explicit transactions should contain only DML statements and query statements, because DDL statements implicitly commit active transactions. This means that any changes made by the previous statements in the transaction are applied, and any changes made by the subsequent statements in the transaction are not part of the same transaction1.
The other options are not correct because:
B). The autocommit setting can be changed inside a stored procedure, but this does not affect the use of transactions in Snowflake. The autocommit setting determines whether each statement is executed in its own implicit transaction or not. If autocommit is enabled, each statement is committed automatically. If autocommit is disabled, each statement is executed in an implicit transaction until an explicit COMMIT or ROLLBACK is issued. Changing the autocommit setting inside a stored procedure only affects the statements within the stored procedure, and does not affect the statements outside the stored procedure2.
C). A transaction can be started explicitly by executing a BEGIN WORK statement and end explicitly by executing a COMMIT WORK statement, but this is not a characteristic of the use of transactions in Snowflake. This is just one way of writing the statements that start and end an explicit transaction. Snowflake also supports the synonyms BEGIN TRANSACTION and COMMIT, which are recommended over BEGIN WORK and COMMIT WORK1.
D). A transaction can be started explicitly by executing a BEGIN TRANSACTION statement and end explicitly by executing an END TRANSACTION statement, but this is not a valid syntax in Snowflake.
Snowflake does not support the END TRANSACTION statement. The correct way to end an explicit transaction is to use the COMMIT or ROLLBACK statement1.
References:
1: Transactions | Snowflake Documentation
2: AUTOCOMMIT | Snowflake Documentation

NEW QUESTION: 40
When loading data into a table that captures the load time in a column with a default value of either CURRENT_TIME () or CURRENT_TIMESTAMP() what will occur?
A. All rows loaded using a specific COPY statement will have varying timestamps based on when the rows were inserted.

B. Any rows loaded using a specific COPY statement will have varying timestamps based on when the rows were read from the source.
C. Any rows loaded using a specific COPY statement will have varying timestamps based on when the rows were created in the source.
D. All rows loaded using a specific COPY statement will have the same timestamp value.
Answer: D (LEAVE A REPLY)
According to the Snowflake documentation, when loading data into a table that captures the load time in a column with a default value of either CURRENT_TIME () or CURRENT_TIMESTAMP(), the default value is evaluated once per COPY statement, not once per row. Therefore, all rows loaded using a specific COPY statement will have the same timestamp value. This behavior ensures that the timestamp value reflects the time when the data was loaded into the table, not when the data was read from the source or created in the source.
References:
Snowflake Documentation: Loading Data into Tables with Default Values
Snowflake Documentation: COPY INTO table

NEW QUESTION: 41
The diagram shows the process flow for Snowpipe auto-ingest with Amazon Simple Notification Service (SNS) with the following steps:
Step 1: Data files are loaded in a stage.
Step 2: An Amazon S3 event notification, published by SNS, informs Snowpipe - by way of Amazon Simple Queue Service (SQS) - that files are ready to load. Snowpipe copies the files into a queue.
Step 3: A Snowflake-provided virtual warehouse loads data from the queued files into the target table based on parameters defined in the specified pipe.
If an AWS Administrator accidentally deletes the SQS subscription to the SNS topic in Step 2, what will happen to the pipe that references the topic to receive event messages from Amazon S3?
A. The pipe will continue to receive the messages as Snowflake will automatically restore the subscription to the same SNS topic and will recreate the pipe by specifying the same SNS topic name in the pipe definition.
B. The pipe will no longer be able to receive the messages and the user must wait for 24 hours from the time when the SNS topic subscription was deleted. Pipe recreation is not required as the pipe will reuse the same subscription to the existing SNS topic after 24 hours.
C. The pipe will continue to receive the messages as Snowflake will automatically restore the subscription by creating a new SNS topic. Snowflake will then recreate the pipe by specifying the new SNS topic name in the pipe definition.

D. The pipe will no longer be able to receive the messages. To restore the system immediately, the user needs to manually create a new SNS topic with a different name and then recreate the pipe by specifying the new SNS topic name in the pipe definition.

Answer: D (LEAVE A REPLY)

If an AWS Administrator accidentally deletes the SQS subscription to the SNS topic in Step 2, the pipe that references the topic to receive event messages from Amazon S3 will no longer be able to receive the messages.

This is because the SQS subscription is the link between the SNS topic and the Snowpipe notification channel.

Without the subscription, the SNS topic will not be able to send notifications to the Snowpipe queue, and the pipe will not be triggered to load the new files. To restore the system immediately, the user needs to manually create a new SNS topic with a different name and then recreate the pipe by specifying the new SNS topic name in the pipe definition. This will create a new notification channel and a new SQS subscription for the pipe. Alternatively, the user can also recreate the SQS subscription to the existing SNS topic and then alter the pipe to use the same SNS topic name in the pipe definition. This will also restore the notification channel and the pipe functionality. References:

* Automating Snowpipe for Amazon S3
* Enabling Snowpipe Error Notifications for Amazon SNS
* HowTo: Configuration steps for Snowpipe Auto-Ingest with AWS S3 Stages

NEW QUESTION: 42

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. Developers create their own datasets to work against transformed versions of the live data.

B. Production and development run in different databases in the same account, and Developers need to see production-like data but with specific columns masked.

C. 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.

D. 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.

E. 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.

Answer: A,C (LEAVE A REPLY)

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 References:
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
: SnowPro Advanced: Architect | Study Guide
: Cloning Overview
: Loading Data Using COPY into a Table
: Transforming Data During a Load
: Data Sharing Overview
: Secure Views
: Cloning Databases, Schemas, and Tables
: Cloning for Testing and Development

NEW QUESTION: 43
What considerations need to be taken when using database cloning as a tool for data lifecycle management in a development environment? (Select TWO).
A. Any pipes in the source are not cloned.
B. Any pipes in the source referring to internal stages are not cloned.
C. Any pipes in the source referring to external stages are not cloned.
D. The clone inherits all granted privileges of all child objects in the source object, including the database.
E. The clone inherits all granted privileges of all child objects in the source object, excluding the database.
Answer: A,D (LEAVE A REPLY)
Database cloning is a feature of Snowflake that allows creating a copy of a database, schema, table, or view without consuming any additional storage space. Database cloning can be used as a tool for data lifecycle management in a development environment, where

developers and testers can work on isolated copies of production data without affecting the original data or each other1.
However, there are some considerations that need to be taken when using database cloning in a development environment, such as:
Any pipes in the source are not cloned. Pipes are objects that load data from a stage into a table continuously. Pipes are not cloned because they are associated with a specific stage and table, and cloning them would create duplicate data loading and potential conflicts2.
The clone inherits all granted privileges of all child objects in the source object, including the database.
Privileges are the permissions that control the access and actions that can be performed on an object.
When a database is cloned, the clone inherits all the privileges that were granted on the source database and its child objects, such as schemas, tables, and views. This means that the same roles that can access and modify the source database can also access and modify the clone, unless the privileges are explicitly revoked or modified3.
The other options are not correct because:
B). Any pipes in the source referring to internal stages are not cloned. This is a subset of option A, which states that any pipes in the source are not cloned, regardless of the type of stage they refer to.
C). Any pipes in the source referring to external stages are not cloned. This is also a subset of option A, which states that any pipes in the source are not cloned, regardless of the type of stage they refer to.
E). The clone inherits all granted privileges of all child objects in the source object, excluding the database. This is incorrect, as the clone inherits all granted privileges of the source object, including the database.
References:
1: Database Cloning | Snowflake Documentation
2: Pipes | Snowflake Documentation
3: Access Control Privileges | Snowflake Documentation

NEW QUESTION: 44
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 CLONE TABLE Recover_Data FROM Data AT(OFFSET => -60*5);
B. CREATE CLONE Recover_Data FROM Data AT(OFFSET => -60*5);
C. CREATE TABLE Recover_Data CLONE Data AT(OFFSET => -60*5);
D. CREATE TABLE Recover Data CLONE Data AT(TIME => -60*5);
Answer: C (LEAVE A REPLY)
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: 45
An Architect is troubleshooting a query with poor performance using the QUERY_HIST0RY function. The Architect observes that the COMPILATIONJHME is greater than the EXECUTIONJTIME.
What is the reason for this?
A. The query is processing a very large dataset.
B. The query has overly complex logic.
C. The query is queued for execution.
D. The query is reading from remote storage.
Answer: B (LEAVE A REPLY)
Compilation time is the time it takes for the optimizer to create an optimal query plan for the efficient execution of the query. It also involves some pruning of partition files, making the query execution efficient2 If the compilation time is greater than the execution time, it means that the optimizer spent more time analyzing the query than actually running it. This could indicate that the query has overly complex logic, such as multiple joins, subqueries, aggregations, or expressions. The complexity of the query could also affect the size and quality of the query plan, which could impact the performance of the query3 To reduce the compilation time, the Architect can try to simplify the query logic, use views or common table expressions (CTEs) to break down the query into smaller parts, or use hints to guide the optimizer. The Architect can also use the EXPLAIN command to examine the query plan and identify potential bottlenecks or inefficiencies4 References:
1: SnowPro Advanced: Architect | Study Guide 5
2: Snowflake Documentation | Query Profile Overview 6
3: Understanding Why Compilation Time in Snowflake Can Be Higher than Execution Time 7
4: Snowflake Documentation | Optimizing Query Performance 8
: SnowPro Advanced: Architect | Study Guide

: Query Profile Overview
: Understanding Why Compilation Time in Snowflake Can Be Higher than Execution Time
: Optimizing Query Performance

NEW QUESTION: 46
An Architect is designing a pipeline to stream event data into Snowflake using the Snowflake Kafka connector. The Architect's highest priority is to configure the connector to stream data in the MOST cost-effective manner.
Which of the following is recommended for optimizing the cost associated with the Snowflake Kafka connector?
A. Utilize a higher Buffer.flush.time in the connector configuration.
B. Utilize a higher Buffer.size.bytes in the connector configuration.
C. Utilize a lower Buffer.size.bytes in the connector configuration.
D. Utilize a lower Buffer.count.records in the connector configuration.
Answer: A (LEAVE A REPLY)
The minimum value supported for the buffer.flush.time property is 1 (in seconds). For higher average data flow rates, we suggest that you decrease the default value for improved latency. If cost is a greater concern than latency, you could increase the buffer flush time. Be careful to flush the Kafka memory buffer before it becomes full to avoid out of memory exceptions.
https://docs.snowflake.com/en/user-guide/data-load-snowpipe-streaming-kafka

NEW QUESTION: 47
A company is designing high availability and disaster recovery plans and needs to maximize redundancy and minimize recovery time objectives for their critical application processes. Cost is not a concern as long as the solution is the best available. The plan so far consists of the following steps:
1. Deployment of Snowflake accounts on two different cloud providers.
2. Selection of cloud provider regions that are geographically far apart.
3. The Snowflake deployment will replicate the databases and account data between both cloud provider accounts.
4. Implementation of Snowflake client redirect.

What is the MOST cost-effective way to provide the HIGHEST uptime and LEAST application disruption if there is a service event?

A. Connect the applications using the <organization_name>-<connection_name> URL. Use the Business Critical Snowflake edition.

B. Connect the applications using the <organization_name>-<connection_name> URL. Use the Virtual Private Snowflake (VPS) edition.

C. Connect the applications using the <organization_name>-<accountLocator> URL. Use the Enterprise Snowflake edition.

D. Connect the applications using the <organization_name>-<accountLocator> URL. Use the Business Critical Snowflake edition.

Answer: D (LEAVE A REPLY)

To provide the highest uptime and least application disruption in case of a service event, the best option is to use the Business Critical Snowflake edition and connect the applications using the <organization_name>-<accountLocator> URL. The Business CriticalSnowflake edition offers the highest level of security, performance, and availability for Snowflake accounts. It includes features such as customer-managed encryption keys, HIPAA compliance, and 4-hour RPO and RTO SLAs. It also supports account replication and failover across regions and cloud platforms, which enables business continuity and disaster recovery. By using the <organization_name>-<accountLocator> URL, the applications can leverage the Snowflake Client Redirect feature, which automatically redirects the client connections to the secondary account in case of a failover. This way, the applications can seamlessly switch to the backup account without any manual intervention or configuration changes. The other options are less cost-effective or less reliable because they either use a lower edition of Snowflake, which does not support account replication and failover, or they use the <organization_name>-<connection_name> URL, which does not support client redirect and requires manual updates to the connection string in case of a failover. References:
[Snowflake Editions] 1
[Replication and Failover/Failback] 2
[Client Redirect] 3
[Snowflake Account Identifiers] 4

NEW QUESTION: 48

A company's daily Snowflake workload consists of a huge number of concurrent queries triggered between 9pm and 11pm. At the individual level, these queries are smaller statements that get completed within a short time period.

What configuration can the company's Architect implement to enhance the performance of this workload?
(Choose two.)

A. Enable a multi-clustered virtual warehouse in maximized mode during the workload duration.
B. Set the MAX_CONCURRENCY_LEVEL to a higher value than its default value of 8 at the virtual warehouse level.
C. Increase the size of the virtual warehouse to size X-Large.
D. Reduce the amount of data that is being processed through this workload.
E. Set the connection timeout to a higher value than its default.
Answer: A,B (LEAVE A REPLY)
These two configuration options can enhance the performance of the workload that consists of a huge number of concurrent queries that are smaller and faster.
Enabling a multi-clustered virtual warehouse in maximized mode allows the warehouse to scale out automatically by adding more clusters as soon as the current cluster is fully loaded, regardless of the number of queries in the queue. This can improve the concurrency and throughput of the workload by minimizing or preventing queuing. The maximized mode is suitable for workloads that require high performance and low latency, and are less sensitive to credit consumption1.
Setting the MAX_CONCURRENCY_LEVEL to a higher value than its default value of 8 at the virtual warehouse level allows the warehouse to run more queries concurrently on each cluster. This can improve the utilization and efficiency of the warehouse resources, especially for smaller and faster queries that do not require a lot of processing power. The MAX_CONCURRENCY_LEVEL parameter can be set when creating or modifying a warehouse, and it can be changed at any time2.
References:
Snowflake Documentation: Scaling Policy for Multi-cluster Warehouses
Snowflake Documentation: MAX_CONCURRENCY_LEVEL

NEW QUESTION: 49
Company A would like to share data in Snowflake with Company B. Company B is not on the same cloud platform as Company A.
What is required to allow data sharing between these two companies?
A. Create a pipeline to write shared data to a cloud storage location in the target cloud provider.
B. Ensure that all views are persisted, as views cannot be shared across cloud platforms.
C. Setup data replication to the region and cloud platform where the consumer resides.
D. Company A and Company B must agree to use a single cloud platform: Data sharing is only possible if the companies share the same cloud provider.
Answer: (SHOW ANSWER)
According to the SnowPro Advanced: Architect documents and learning resources, the requirement to allow data sharing between two companies that are not on the same cloud platform is to set up data replication to the region and cloud platform where the consumer resides. Data replication is a feature of Snowflake that enables copying databases across

accounts in different regions and cloud platforms. Data replication allows data providers to securely share data with data consumers across different regions and cloud platforms by creating a replica database in the consumer's account. The replica database is read-only and automatically synchronized with the primary database in the provider's account. Data replication is useful for scenarios where data sharing is not possible or desirable due to latency, compliance, or security reasons1. The other options are incorrect because they are not required or feasible to allow data sharing between two companies that are not on the same cloud platform. Option A is incorrect because creating a pipeline to write shared data to a cloud storage location in the target cloud provider is not a secure or efficient way of sharing data. It would require additional steps to load the data from the cloud storage to the consumer's account, and it would not leverage the benefits of Snowflake's data sharing features. Option B is incorrect because ensuring that all views are persisted is not relevant for data sharing across cloud platforms. Views can be shared across cloud platforms as long as they reference objects in the same database. Persisting views is an option to improve the performance of querying views, but it is not required for data sharing2. Option D is incorrect because Company A and Company B do not need to agree to use a single cloud platform. Data sharing is possible across different cloud platforms using data replication or other methods, such as listings or auto-fulfillment3. References: Replicating Databases Across Multiple Accounts | Snowflake Documentation, Persisting Views | Snowflake Documentation, Sharing Data Across Regions and Cloud Platforms | Snowflake Documentation

NEW QUESTION: 50
A global company needs to securely share its sales and Inventory data with a vendor using a Snowflake account.
The company has its Snowflake account In the AWS eu-west 2 Europe (London) region. The vendor's Snowflake account Is on the Azure platform in the West Europe region. How should the company's Architect configure the data share?
A. 1. Create a share.
2. Add objects to the share.
3. Add a consumer account to the share for the vendor to access.
B. 1. Create a share.
2. Create a reader account for the vendor to use.
3. Add the reader account to the share.
C. 1. Create a new role called db_share.
2. Grant the db_share role privileges to read data from the company database and schema.
3. Create a user for the vendor.
4. Grant the ds_share role to the vendor's users.
D. 1. Promote an existing database in the company's local account to primary.
2. Replicate the database to Snowflake on Azure in the West-Europe region.

3. Create a share and add objects to the share.
4. Add a consumer account to the share for the vendor to access.

Answer: A (LEAVE A REPLY)

The correct way to securely share data with a vendor using a Snowflake account on a different cloud platform and region is to create a share, add objects to the share, and add a consumer account to the share for the vendor to access. This way, the company can control what data is shared, who can access it, and how long the share is valid. The vendor can then query the shared data without copying or moving it to their own account. The other options are either incorrect or inefficient, as they involve creating unnecessary reader accounts, users, roles, or database replication.
https://learn.snowflake.com/en/certifications/snowpro-advanced-architect/

NEW QUESTION: 51

Consider the following scenario where a masking policy is applied on the CREDICARDND column of the CREDITCARDINFO table. The masking policy definition Is as follows:
Sample data for the CREDITCARDINFO table is as follows:
NAME EXPIRYDATE CREDITCARDNO
JOHN DOE 2022-07-23 4321 5678 9012 1234
if the Snowflake system rotes have not been granted any additional roles, what will be the result?

A. The sysadmin can see the CREDICARDND column data in clear text.
B. The owner of the table will see the CREDICARDND column data in clear text.
C. Anyone with the PI_ANALYTICS role will see the last 4 characters of the CREDICARDND column data in dear text.
D. Anyone with the PI_ANALYTICS role will see the CREDICARDND column as*** 'MASKED* **'.

Answer: D (LEAVE A REPLY)

The masking policy defined in the image indicates that if a user has the PI_ANALYTICS role, they will be able to see the last 4 characters of the CREDITCARDNO column data in clear text. Otherwise, they will see 'MASKED'. Since Snowflake system roles have not been granted any additional roles, they won't have the PI_ANALYTICS role and therefore cannot view the last 4 characters of credit card numbers.
To apply a masking policy on a column in Snowflake, you need to use the ALTER TABLE ... ALTER COLUMN command or the ALTER VIEW command and specify the policy name. For example, to apply the creditcardno_mask policy on the CREDITCARDNO column of the CREDITCARDINFO table, you can use the following command:
ALTER TABLE CREDITCARDINFO ALTER COLUMN CREDITCARDNO SET MASKING POLICY creditcardno_mask; For more information on how to create and use masking policies in Snowflake, you can refer to the following resources:

CREATE MASKING POLICY: This document explains the syntax and usage of the CREATE MASKING POLICY command, which allows you to create a new masking policy or replace an existing one.
Using Dynamic Data Masking: This guide provides instructions on how to configure and use dynamic data masking in Snowflake, which is a feature that allows you to mask sensitive data based on the execution context of the user.
ALTER MASKING POLICY: This document explains the syntax and usage of the ALTER MASKING POLICY command, which allows you to modify the properties of an existing masking policy.
References: 1: https://docs.snowflake.com/en/sql-reference/sql/create-masking-policy 2: https://docs.snowflake.com/en/user-guide/security-column-ddm-use 3: https://docs.snowflake.com/en/sql-reference/sql/alter-masking-policy

NEW QUESTION: 52
How can the Snowpipe REST API be used to keep a log of data load history?
A. Call insertReport every 20 minutes, fetching the last 10,000 entries.
B. Call loadHistoryScan every minute for the maximum time range.
C. Call insertReport every 8 minutes for a 10-minute time range.
D. Call loadHistoryScan every 10 minutes for a 15-minutes range.
Answer: D (LEAVE A REPLY)
The Snowpipe REST API provides two endpoints for retrieving the data load history: insertReport and loadHistoryScan. The insertReport endpoint returns the status of the files that were submitted to the insertFiles endpoint, while the loadHistoryScan endpoint returns the history of the files that were actually loaded into the table by Snowpipe. To keep a log of data load history, it is recommended to use the loadHistoryScan endpoint, which provides more accurate and complete information about the data ingestion process. The loadHistoryScan endpoint accepts a start time and an end time as parameters, and returns the files that were loaded within that time range. The maximum time range that can be specified is 15 minutes, and the maximum number of files that can be returned is 10,000. Therefore, to keep a log of data load history, the best option is to call the loadHistoryScan endpoint every 10 minutes for a 15-minute time range, and store the results in a log file or a table. This way, the log will capture all the files that were loaded by Snowpipe, and avoid any gaps or overlaps in the time range. The other options are incorrect because:
Calling insertReport every 20 minutes, fetching the last 10,000 entries, will not provide a complete log of data load history, as some files may be missed or duplicated due to the asynchronous nature of Snowpipe. Moreover, insertReport only returns the status of the files that were submitted, not the files that were loaded.
Calling loadHistoryScan every minute for the maximum time range will result in too many API calls and unnecessary overhead, as the same files will be returned multiple times. Moreover, the maximum time range is 15 minutes, not 1 minute.

Calling insertReport every 8 minutes for a 10-minute time range will suffer from the same problems as option A, and also create gaps or overlaps in the time range.
References:
Snowpipe REST API
Option 1: Loading Data Using the Snowpipe REST API
PIPE_USAGE_HISTORY

NEW QUESTION: 53
An Architect is implementing a CI/CD process. When attempting to clone a table from a production to a development environment, the cloning operation fails.
What could be causing this to happen?
A. The table is transient.
B. The table has a masking policy.
C. The retention time for the table is set to zero.
D. Tables cannot be cloned from a higher environment to a lower environment.
Answer: B (LEAVE A REPLY)
Cloning a table with a masking policy can cause the cloning operation to fail because the masking policy is not automatically cloned with the table. This is due to the fact that the masking policy is considered a separate object with its own set of privileges1.
References
Snowflake Documentation on Cloning Considerations1.

NEW QUESTION: 54
A Snowflake Architect is setting up database replication to support a disaster recovery plan. The primary database has external tables.
How should the database be replicated?
A. Create a clone of the primary database then replicate the database.
B. Move the external tables to a database that is not replicated, then replicate the primary database.
C. Replicate the database ensuring the replicated database is in the same region as the external tables.
D. Share the primary database with an account in the same region that the database will be replicated to.
Answer: B (LEAVE A REPLY)
Database replication is a feature that allows you to create a copy of a database in another account, region, or cloud platform for disaster recovery or business continuity purposes. However, not all database objects can be replicated. External tables are one of the exceptions, as they reference data files stored in an external stage that is not part of Snowflake. Therefore, to replicate a database that contains external tables, you need to move the external tables to a separate database that is not replicated, and then replicate the primary database that contains the other objects. This way, you can avoid replication

errors and ensure consistency between the primary and secondary databases. The other options are incorrect because they either do not address the issue of external tables, or they use an alternative method that is not supported by Snowflake. You cannot create a clone of the primary database and then replicate it, as replication only works on the original database, not on its clones. You also cannot share the primary database with another account, as sharing is a different feature that does not create a copy of the database, but rather grants access to the shared objects. Finally, you do not need to ensure that the replicated database is in the same region as the external tables, as external tables can access data files stored in any region or cloud platform, as long as the stage URL is valid and accessible. References:
[Replication and Failover/Failback] 1
[Introduction to External Tables] 2
[Working with External Tables] 3
[Replication : How to migrate an account from One Cloud Platform or Region to another in Snowflake] 4

NEW QUESTION: 55
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. Use, at minimum, the Business Critical edition of Snowflake.
B. Create Dynamic Data Masking policies and apply them to columns that contain PHI.
C. Use the Internal Tokenization feature to obfuscate sensitive data.
D. Use the External Tokenization feature to obfuscate sensitive data.
E. Rewrite SQL queries to eliminate projections of PHI data based on current_role().
F. Avoid sharing data with partner organizations.
Answer: A,B,D (LEAVE A REPLY)
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 orredacting 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: 56
A Snowflake Architect is designing an application and tenancy strategy for an organization where strong legal isolation rules as well as multi-tenancy are requirements.
Which approach will meet these requirements if Role-Based Access Policies (RBAC) is a viable option for isolating tenants?
A. Create accounts for each tenant in the Snowflake organization.
B. Create an object for each tenant strategy if row level security is viable for isolating tenants.
C. Create an object for each tenant strategy if row level security is not viable for isolating tenants.
D. Create a multi-tenant table strategy if row level security is not viable for isolating tenants.
Answer: A (LEAVE A REPLY)
This approach meets the requirements of strong legal isolation and multi-tenancy. By creating separate accounts for each tenant, the application can ensure that each tenant has its own dedicated storage, compute, and metadata resources, as well as its own

encryption keys and security policies. This provides the highest level of isolation and data protection among the tenancy models. Furthermore, by creating the accounts within the same Snowflake organization, the application can leverage the features of Snowflake Organizations, such as centralized billing, account management, and cross-account data sharing.
References:
Snowflake Organizations Overview | Snowflake Documentation
Design Patterns for Building Multi-Tenant Applications on Snowflake

NEW QUESTION: 57
What are some of the characteristics of result set caches? (Choose three.)
A. Time Travel queries can be executed against the result set cache.
B. Snowflake persists the data results for 24 hours.
C. Each time persisted results for a query are used, a 24-hour retention period is reset.
D. The data stored in the result cache will contribute to storage costs.
E. The retention period can be reset for a maximum of 31 days.
F. The result set cache is not shared between warehouses.
Answer: B,C,E (LEAVE A REPLY)
Comprehensive and Detailed Explanation: According to the SnowPro Advanced: Architect documents and learning resources, some of the characteristics of result set caches are:
Snowflake persists the data results for 24 hours. This means that the result set cache holds the results of every query executed in the past 24 hours, and can be reused if the same query is submitted again and the underlying data has not changed1.
Each time persisted results for a query are used, a 24-hour retention period is reset. This means that the result set cache extends the lifetime of the results every time they are reused, up to a maximum of 31 days from the date and time that the query was first executed1.
The retention period can be reset for a maximum of 31 days. This means that the result set cache will purge the results after 31 days, regardless of whether they are reused or not. After 31 days, the next time the query is submitted, a new result is generated and persisted1.
The other options are incorrect because they are not characteristics of result set caches. Option A is incorrect because Time Travel queries cannot be executed against the result set cache. Time Travel queries use the AS OF clause to access historical data that is stored in the storage layer, not the result set cache2. Option D is incorrect because the data stored in the result set cache does not contribute to storage costs. The result set cache is maintained by the service layer, and does not incurany additional charges1.
Option F is incorrect because the result set cache is shared between warehouses. The result set cache is available across virtual warehouses, so query results returned to one user are available to any other user on the system who executes the same query, provided

the underlying data has not changed1. References: Using Persisted Query Results | Snowflake Documentation, Time Travel | Snowflake Documentation

NEW QUESTION: 58
Why might a Snowflake Architect use a star schema model rather than a 3NF model when designing a data architecture to run in Snowflake? (Select TWO).
A. Snowflake cannot handle the joins implied in a 3NF data model.
B. The Architect wants to remove data duplication from the data stored in Snowflake.
C. The Architect is designing a landing zone to receive raw data into Snowflake.
D. The BI tool needs a data model that allows users to summarize facts across different dimensions, or to drill down from the summaries.
E. The Architect wants to present a simple flattened single view of the data to a particular group of end users.
Answer: (SHOW ANSWER)
A star schema model is a type of dimensional data model that consists of a single fact table and multiple dimension tables. A 3NF model is a type of relational data model that follows the third normal form, which eliminates data redundancy and ensures referential integrity. A Snowflake Architect might use a star schema model rather than a 3NF model when designing a data architecture to run in Snowflake for the following reasons:
A star schema model is more suitable for analytical queries that require aggregating and slicing data across different dimensions, such as those performed by a BI tool. A 3NF model is more suitable for transactional queries that require inserting, updating, and deleting individual records.
A star schema model is simpler and faster to query than a 3NF model, as it involves fewer joins and less complex SQL statements. A 3NF model is more complex and slower to query, as it involves more joins and more complex SQL statements.
A star schema model can provide a simple flattened single view of the data to a particular group of end users, such as business analysts or data scientists, who need to explore and visualize the data. A 3NF model can provide a more detailed and normalized view of the data to a different group of end users, such as application developers or data engineers, who need to maintain and update the data.
The other options are not valid reasons for choosing a star schema model over a 3NF model in Snowflake:
Snowflake can handle the joins implied in a 3NF data model, as it supports ANSI SQL and has a powerful query engine that can optimize and execute complex queries efficiently.
The Architect can use both star schema and 3NF models to remove data duplication from the data stored in Snowflake, as both models can enforce data integrity and avoid data anomalies. However, the trade-off is that a star schema model may have more data redundancy than a 3NF model, as it denormalizes the data for faster query performance, while a 3NF model may have less data redundancy than a star schema model, as it normalizes the data for easier data maintenance.

The Architect can use both star schema and 3NF models to design a landing zone to receive raw data into Snowflake, as both models can accommodate different types of data sources and formats. However, the choice of the model may depend on the purpose and scope of the landing zone, such as whether it is a temporary or permanent storage, whether it is a staging area or a data lake, and whether it is a single source or a multi-source integration.
References:
Snowflake Architect Training
Data Modeling: Understanding the Star and Snowflake Schemas
Data Vault vs Star Schema vs Third Normal Form: Which Data Model to Use?
Star Schema vs Snowflake Schema: 5 Key Differences
Dimensional Data Modeling - Snowflake schema
Star schema vs Snowflake Schema

NEW QUESTION: 59
Which of the below commands will use warehouse credits?
A. SHOW TABLES LIKE 'SNOWFL%';
B. SELECT MAX(FLAKE_ID) FROM SNOWFLAKE;
C. SELECT COUNT(*) FROM SNOWFLAKE;
D. SELECT COUNT(FLAKE_ID) FROM SNOWFLAKE GROUP BY FLAKE_ID;
Answer: (SHOW ANSWER)
Warehouse credits are used to pay for the processing time used by each virtual warehouse in Snowflake.
A virtual warehouse is a cluster of compute resources that enables executing queries, loading data, and performing other DML operations. Warehouse credits are charged based on the number of virtual warehouses you use, how long they run, and their size1.
Among the commands listed in the question, the following ones will use warehouse credits:
SELECT MAX(FLAKE_ID) FROM SNOWFLAKE: This command will use warehouse credits because it is a query that requires a virtual warehouse to execute. The query will scan the SNOWFLAKE table and return the maximum value of the FLAKE_ID column2. Therefore, option B is correct.
SELECT COUNT(*) FROM SNOWFLAKE: This command will also use warehouse credits because it is a query that requires a virtual warehouse to execute. The query will scanthe SNOWFLAKE table and return the number of rows in the table3. Therefore, option C is correct.
SELECT COUNT(FLAKE_ID) FROM SNOWFLAKE GROUP BY FLAKE_ID: This command
will also use warehouse credits because it is a query that requires a virtual warehouse to execute. The query will scan the SNOWFLAKE table and return the number of rows for each distinct value of the FLAKE_ID column4. Therefore, option D is correct.
The command that will not use warehouse credits is:

SHOW TABLES LIKE 'SNOWFL%': This command will not use warehouse credits because it is a metadata operation that does not require a virtual warehouse to execute. The command will return the names of the tables that match the pattern 'SNOWFL%' in the current database and schema5. Therefore, option A is incorrect.
References: : Understanding Compute Cost : MAX Function : COUNT Function : GROUP BY Clause : SHOW TABLES

NEW QUESTION: 60
An Architect needs to design a data unloading strategy for Snowflake, that will be used with the COPY INTO
<location> command.
Which configuration is valid?
A. Location of files: Snowflake internal location
. File formats: CSV, XML
. File encoding: UTF-8
. Encryption: 128-bit
B. Location of files: Amazon S3
. File formats: CSV, JSON
. File encoding: Latin-1 (ISO-8859)
. Encryption: 128-bit
C. Location of files: Google Cloud Storage
. File formats: Parquet
. File encoding: UTF-8
Compression: gzip
D. Location of files: Azure ADLS
. File formats: JSON, XML, Avro, Parquet, ORC
. Compression: bzip2
. Encryption: User-supplied key
Answer: C (LEAVE A REPLY)
For the configuration of data unloading in Snowflake, the valid option among the provided choices is "C." This is because Snowflake supports unloading data into Google Cloud Storage using the COPY INTO
<location> command with specific configurations. The configurations listed in option C, such as Parquet file format with UTF-8 encoding and gzip compression, are all supported by Snowflake. Notably, Parquet is a columnar storage file format, which is optimal for high-performance data processing tasks in Snowflake. The UTF-8 file encoding and gzip compression are both standard and widely used settings that are compatible with Snowflake's capabilities for data unloading to cloud storage platforms.References:
* Snowflake Documentation on COPY INTO command
* Snowflake Documentation on Supported File Formats
* Snowflake Documentation on Compression and Encoding Options

NEW QUESTION: 61
A user is executing the following command sequentially within a timeframe of 10 minutes from start to finish:
What would be the output of this query?
A. Table T_SALES_CLONE successfully created.
B. Time Travel data is not available for table T_SALES.
C. The offset -> is not a valid clause in the clone operation.
D. Syntax error line 1 at position 58 unexpected 'at'.
Answer: A (LEAVE A REPLY)
The query is executing a clone operation on an existing tablet_saleswith an offset to account for the retention time. The syntax used is correct for cloning a table in Snowflake, and the use of theat(offset => -60*30)clause is valid. This specifies that the clone should be based on the state of the table 30 minutes prior (60 seconds *
30). Assuming the tablet_salesexists and has been modified within the last 30 minutes, and considering the data_retention_time_in_daysisset to 1 day (which enables time travel queries for the past 24 hours), the table t_sales_clonewould be successfully created based on the state oft_sales30 minutes before the clone command was issued.

NEW QUESTION: 62
An Architect needs to allow a user to create a database from an inbound share.
To meet this requirement, the user's role must have which privileges? (Choose two.)
A. IMPORT SHARE;
B. IMPORT PRIVILEGES;
C. CREATE DATABASE;
D. CREATE SHARE;
E. IMPORT DATABASE;
Answer: (SHOW ANSWER)
According to the Snowflake documentation, to create a database from an inbound share, the user's role must have the following privileges:
The CREATE DATABASE privilege on the current account. This privilege allows the user to create a new database in the account1.

The IMPORT DATABASE privilege on the share. This privilege allows the user to import a database from the share into the account2. The other privileges listed are not relevant for this requirement. The IMPORT SHARE privilege is used to import a share into the account, not a database3. The IMPORT PRIVILEGES privilege is used to import the privileges granted on the shared objects, not the objects themselves2. The CREATE SHARE privilege is used to create a share to provide data to other accounts, not to consume data from other accounts4.
References:
CREATE DATABASE | Snowflake Documentation
Importing Data from a Share | Snowflake Documentation
Importing a Share | Snowflake Documentation
CREATE SHARE | Snowflake Documentation

NEW QUESTION: 63
Which security, governance, and data protection features require, at a MINIMUM, the Business Critical edition of Snowflake? (Choose two.)
A. Extended Time Travel (up to 90 days)
B. Customer-managed encryption keys through Tri-Secret Secure
C. Periodic rekeying of encrypted data
D. AWS, Azure, or Google Cloud private connectivity to Snowflake
E. Federated authentication and SSO
Answer: B,D (LEAVE A REPLY)
According to the SnowPro Advanced: Architect documents and learning resources, the security, governance, and data protection features that require, at a minimum, the Business Critical edition of Snowflake are:
Customer-managed encryption keys through Tri-Secret Secure. This feature allows customers to manage their own encryption keys for data at rest in Snowflake, using a combination of three secrets: a master key, a service key, and a security password. This provides an additional layer of security and control over the data encryption and decryption process1.
Periodic rekeying of encrypted data. This feature allows customers to periodically rotate the encryption keys for data at rest in Snowflake, using either Snowflake-managed keys or customer-managed keys. This enhances the security and protection of the data by reducing the risk of key compromise or exposure2.
The other options are incorrect because they do not require the Business Critical edition of Snowflake. Option A is incorrect because extended Time Travel (up to 90 days) is available with the Enterprise edition of Snowflake3. Option D is incorrect because AWS, Azure, or Google Cloud private connectivity to Snowflake is available with the Standard edition of Snowflake4. Option E is incorrect because federated authentication and SSO are available with the Standard edition of Snowflake5. References: Tri-Secret Secure | Snowflake Documentation, Periodic Rekeying of Encrypted Data | Snowflake

Documentation, Snowflake Editions | Snowflake Documentation, Snowflake Network Policies | Snowflake Documentation, Configuring Federated Authentication and SSO | Snowflake Documentation

NEW QUESTION: 64
An Architect has a VPN_ACCESS_LOGS table in the SECURITY_LOGS schema containing timestamps of the connection and disconnection, username of the user, and summary statistics.
What should the Architect do to enable the Snowflake search optimization service on this table?
A. Assume role with OWNERSHIP on future tables and ADD SEARCH OPTIMIZATION on the SECURITY_LOGS schema.
B. Assume role with ALL PRIVILEGES including ADD SEARCH OPTIMIZATION in the SECURITY LOGS schema.
C. Assume role with OWNERSHIP on VPN_ACCESS_LOGS and ADD SEARCH OPTIMIZATION in the SECURITY_LOGS schema.
D. Assume role with ALL PRIVILEGES on VPN_ACCESS_LOGS and ADD SEARCH OPTIMIZATION in the SECURITY_LOGS schema.
Answer: C (LEAVE A REPLY)
According to the SnowPro Advanced: Architect Exam Study Guide, to enable the search optimization service on a table, the user must have the ADD SEARCH OPTIMIZATION privilege on the table and the schema.
The privilege can be granted explicitly or inherited from a higher-level object, such as a database or a role. The OWNERSHIP privilege on a table implies the ADD SEARCH OPTIMIZATION privilege, so the user who owns the table can enable the search optimization service on it. Therefore, the correct answer is to assume a role with OWNERSHIP on VPN_ACCESS_LOGS and ADD SEARCH OPTIMIZATION in the SECURITY_LOGS schema. This will allow the user to enable the search optimization service on the VPN_ACCESS_LOGS table and any future tables created in the SECURITY_LOGS schema. The other options are incorrect because they either grant excessive privileges or do not grant the required privileges on the table or the schema.
References:
SnowPro Advanced: Architect Exam Study Guide, page 11, section 2.3.1
Snowflake Documentation: Enabling the Search Optimization Service

NEW QUESTION: 65
Two queries are run on the customer_address table:
create or replace TABLE CUSTOMER_ADDRESS (CA_ADDRESS_SK NUMBER(38,0), CA_ADDRESS_ID VARCHAR(16), CA_STREET_NUMBER VARCHAR(IO) CA_STREET_NAME

VARCHAR(60), CA_STREET_TYPE VARCHAR(15), CA_SUITE_NUMBER VARCHAR(10), CA_CITY VARCHAR(60), CA_COUNTY VARCHAR(30), CA_STATE VARCHAR(2), CA_ZIP VARCHAR(10), CA_COUNTRY VARCHAR(20), CA_GMT_OFFSET NUMBER(5,2), CA_LOCATION_TYPE VARCHAR(20)); ALTER TABLE DEMO_DB.DEMO_SCH.CUSTOMER_ADDRESS ADD SEARCH OPTIMIZATION ON SUBSTRING(CA_ADDRESS_ID); Which queries will benefit from the use of the search optimization service? (Select TWO).

A. select * from DEMO_DB.DEMO_SCH.CUSTOMER_ADDRESS Where substring(CA_ADDRESS_ID,1,8)= substring('AAAAAAAAPHPPLBAAASKDJHASLKDJHASKJD',1,8);

B. select * from DEMO_DB.DEMO_SCH.CUSTOMER_ADDRESS Where CA_ADDRESS_ID= substring('AAAAAAAAPHPPLBAAASKDJHASLKDJHASKJD',1,16);

C. select*fromDEMO_DB.DEMO_SCH.CUSTOMER_ADDRESSWhereCA_ADDRESS_IDLIKE '%BAAASKD%';

D. select*fromDEMO_DB.DEMO_SCH.CUSTOMER_ADDRESSWhereCA_ADDRESS_IDLIKE '%PHPP%';

E. select*fromDEMO_DB.DEMO_SCH.CUSTOMER_ADDRESSWhereCA_ADDRESS_IDNOT LIKE '%AAAAAAAAPHPPL%';

Answer: A,B (LEAVE A REPLY)

The use of the search optimization service in Snowflake is particularly effective when queries involve operations that match exact substrings or start from the beginning of a string. The ALTER TABLE command adding search optimization specifically for substrings on the CA_ADDRESS_ID field allows the service to create an optimized search path for queries using substring matches.

* Option A benefits because it directly matches a substring from the start of the CA_ADDRESS_ID, aligning with the optimization's capability to quickly locate records based on the beginning segments of strings.

* Option B also benefits, despite performing a full equality check, because it essentially compares the full length of CA_ADDRESS_ID to a substring, which can leverage the substring index for efficient retrieval.Options C, D, and E involve patterns that do not start from the beginning of the string or use negations, which are not optimized by the search optimization service configured for starting substring matches.References: Snowflake's documentation on the use of search optimization for substring matching in SQL queries.

NEW QUESTION: 66

Which query will identify the specific days and virtual warehouses that would benefit from a multi-cluster warehouse to improve the performance of a particular workload?

A. A screen shot of a computer Description automatically generated
B. A white background with black text Description automatically generated
C. A white background with black text Description automatically generated
D. A close up of a message Description automatically generated
Answer: C (LEAVE A REPLY)
A multi-cluster warehouse is a virtual warehouse that can scale compute resources by adding or removing clusters based on the workload demand. A multi-cluster warehouse can improve the performance of a particular workload by reducing the query queue time and the data spillage to local storage. To identify the specific days and virtual warehouses that would benefit from a multi-cluster warehouse, you need to analyze the query history and look for the following indicators:
High average queued load: This metric shows the average number of queries waiting in the queue for each warehouse cluster. A high value indicates that the warehouse is overloaded and cannot handle the concurrency demand.
High bytes spilled to local storage: This metric shows the amount of data that was spilled from memory to local disk during query processing. A high value indicates that the warehouse size is too small and cannot fit the data in memory.
High variation in workload: This metric shows the fluctuation in the number of queries submitted to the warehouse over time. A high variation indicates that the workload is unpredictable and dynamic, and requires a flexible scaling policy.
The query in option C is the best one to identify these indicators, as it selects the date, warehouse name, bytes spilled to local storage, and sum of average queued load from the query history table, and filters the results where bytes spilled to local storage is greater than zero. This query will show the days and warehouses that experienced data spillage and high queue time, and could benefit from a multi-cluster warehouse with auto-scale mode.
The query in option A is not correct, as it only selects the date and warehouse name, and does not include any metrics to measure the performance of the workload. The query in option B is not correct, as it selects the date, warehouse name, and average execution time, which is not a good indicator of the need for a multi-cluster warehouse. The query in option D is not correct, as it selects the date, warehouse name, and average credits used, which is not a good indicator of the need for a multi-cluster warehouse either.
References: Multi-cluster Warehouses, Query History View, Reducing Queues

NEW QUESTION: 67
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 must be refreshed in order for new data to be visible.
C. Shared databases cannot be cloned.
D. Shared databases are not supported by Time Travel.

E. Shared databases will have the PUBLIC or INFORMATION_SCHEMA schemas without explicitly granting these schemas to the share.
F. Shared databases can also be created as transient databases.
Answer: A,C,D (LEAVE A REPLY)
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: 68
A company's Architect needs to find an efficient way to get data from an external partner, who is also a Snowflake user. The current solution is based on daily JSON extracts that are placed on an FTP server and uploaded to Snowflake manually. The files are changed

several times each month, and the ingestion process needs to be adapted to accommodate these changes.
What would be the MOST efficient solution?
A. Ask the partner to create a share and add the company's account.
B. Ask the partner to use the data lake export feature and place the data into cloud storage where Snowflake can natively ingest it (schema-on-read).
C. Keep the current structure but request that the partner stop changing files, instead only appending new files.
D. Ask the partner to set up a Snowflake reader account and use that account to get the data for ingestion.
Answer: A (LEAVE A REPLY)
The most efficient solution is to ask the partner to create a share and add the company's account (Option A).
This way, the company can access the live data from the partner without any data movement or manual intervention. Snowflake's secure data sharing feature allows data providers to share selected objects in a database with other Snowflake accounts. The shared data is read-only and does not incur any storage or compute costs for the data consumers. The data consumers can query the shared data directly or create local copies of the shared objects in their own databases. Option B is not efficient because it involves using the data lake export feature, which is intended for exporting data from Snowflake to an external data lake, not for importing data from another Snowflake account. The data lake export feature also requires the data provider to create an external stage on cloud storage and use the COPY INTO <location> command to export the data into parquet files. The data consumer then needs to create an external table or a file format to load the data from the cloud storage into Snowflake. This process can be complex and costly, especially if the data changes frequently. Option C is not efficient because it does not solve the problem of manual data ingestion and adaptation. Keeping the current structure of daily JSON extracts on an FTP server and requesting the partner to stop changing files, instead only appending new files, does not improve the efficiency or reliability of the data ingestion process. The company still needs to upload the data to Snowflake manually and deal with any schema changes or data quality issues. Option D is not efficient because it requires the partner to set up a Snowflake reader account and use that account to get the data for ingestion. A reader account is a special type of account that can only consume data from the provider account that created it. It is intended for data consumers who are not Snowflake customers and do not have a licensing agreement with Snowflake. A reader account is not suitable for data ingestion from another Snowflake account, as it does not allow uploading, modifying, or unloading data. The company would need to use external tools or interfaces to access the data from the reader account and load it into their own account, which can be slow and expensive. References: The answer can be verified from Snowflake's official documentation on secure data sharing, data lake export, and reader accounts available on their website. Here are some relevant links:

* Introduction to Secure Data Sharing | Snowflake Documentation
* Data Lake Export Public Preview Is Now Available on Snowflake | Snowflake Blog
* Managing Reader Accounts | Snowflake Documentation

NEW QUESTION: 69
A company needs to have the following features available in its Snowflake account:
1. Support for Multi-Factor Authentication (MFA)
2. A minimum of 2 months of Time Travel availability
3. Database replication in between different regions
4. Native support for JDBC and ODBC
5. Customer-managed encryption keys using Tri-Secret Secure
6. Support for Payment Card Industry Data Security Standards (PCI DSS)
In order to provide all the listed services, what is the MINIMUM Snowflake edition that should be selected during account creation?
A. Standard
B. Enterprise
C. Business Critical
D. Virtual Private Snowflake (VPS)
Answer: C (LEAVE A REPLY)
According to the Snowflake documentation1, the Business Critical edition offers the following features that are relevant to the question:
Support for Multi-Factor Authentication (MFA): This is a standard feature available in all Snowflake editions1.
A minimum of 2 months of Time Travel availability: This is an enterprise feature that allows users to access historical data for up to 90 days1.
Database replication in between different regions: This is an enterprise feature that enables users to replicate databases across different regions or cloud platforms1.
Native support for JDBC and ODBC: This is a standard feature available in all Snowflake editions1.
Customer-managed encryption keys using Tri-Secret Secure: This is a business critical feature that provides enhanced security and data protection by allowing customers to manage their own encryption keys1.
Support for Payment Card Industry Data Security Standards (PCI DSS): This is a business critical feature that ensures compliance with PCI DSS regulations for handling sensitive cardholder data1.
Therefore, the minimum Snowflake edition that should be selected during account creation to provide all the listed services is the Business Critical edition.
References:
Snowflake Editions | Snowflake Documentation

NEW QUESTION: 70

When loading data into a table that captures the load time in a column with a default value of either CURRENT_TIME () or CURRENT_TIMESTAMP () what will occur?
A. All rows loaded using a specific COPY statement will have varying timestamps based on when the rows were inserted.
B. Any rows loaded using a specific COPY statement will have varying timestamps based on when the rows were read from the source.
C. Any rows loaded using a specific COPY statement will have varying timestamps based on when the rows were created in the source.
D. All rows loaded using a specific COPY statement will have the same timestamp value.
Answer: D (LEAVE A REPLY)
When using the COPY command to load data into Snowflake, if a column has a default value set to CURRENT_TIME() or CURRENT_TIMESTAMP(), all rows loaded by that specific COPY command will have the same timestamp. This is because the default value for the timestamp is evaluated at the start of the COPY operation, and that same value is applied to all rows loaded by that operation.
References: This behavior is consistent with Snowflake's documentation on the CURRENT_TIMESTAMP function, which specifies that the timestamp is captured at the time the statement is executed1.

NEW QUESTION: 71
Which command will create a schema without Fail-safe and will restrict object owners from passing on access to other users?
A. create schema EDW.ACCOUNTING WITH MANAGED ACCESS;
B. create schema EDW.ACCOUNTING WITH MANAGED ACCESS DATA_RETENTION_TIME_IN_DAYS - 7;
C. create TRANSIENT schema EDW.ACCOUNTING WITH MANAGED ACCESS DATA_RETENTION_TIME_IN_DAYS = 1;
D. create TRANSIENT schema EDW.ACCOUNTING WITH MANAGED ACCESS DATA_RETENTION_TIME_IN_DAYS = 7;
Answer: (SHOW ANSWER)
A transient schema in Snowflake is designed without a Fail-safe period, meaning it does not incur additional storage costs once it leaves Time Travel, and it is not protected by Fail-safe in the event of a data loss. The WITH MANAGED ACCESS option ensures that all privilege grants, including future grants on objects within the schema, are managed by the schema owner, thus restricting object owners from passing on access to other users1.
References =
*Snowflake Documentation on creating schemas1
*Snowflake Documentation on configuring access control2
*Snowflake Documentation on understanding and viewing Fail-safe3

NEW QUESTION: 72

An Architect needs to allow a user to create a database from an inbound share.
To meet this requirement, the user's role must have which privileges? (Choose two.)
A. IMPORT SHARE;
B. IMPORT PRIVILEGES;
C. CREATE DATABASE;
D. CREATE SHARE;
E. IMPORT DATABASE;
Answer: C,E (LEAVE A REPLY)
According to the Snowflake documentation, to create a database from an inbound share, the user's role must have the following privileges:
* The CREATE DATABASE privilege on the current account. This privilege allows the user to create a new database in the account1.
* The IMPORT DATABASE privilege on the share. This privilege allows the user to import a database from the share into the account2. The other privileges listed are not relevant for this requirement. The IMPORT SHARE privilege is used to import a share into the account, not a database3. The IMPORT PRIVILEGES privilege is used to import the privileges granted on the shared objects, not the objects
* themselves2. The CREATE SHARE privilege is used to create a share to provide data to other accounts, not to consume data from other accounts4.
References:
* CREATE DATABASE | Snowflake Documentation
* Importing Data from a Share | Snowflake Documentation
* Importing a Share | Snowflake Documentation
* CREATE SHARE | Snowflake Documentation

NEW QUESTION: 73
A company is using a Snowflake account in Azure. The account has SAML SSO set up using ADFS as a SCIM identity provider. To validate Private Link connectivity, an Architect performed the following steps:
* Confirmed Private Link URLs are working by logging in with a username/password account
* Verified DNS resolution by running nslookups against Private Link URLs
* Validated connectivity using SnowCD
* Disabled public access using a network policy set to use the company's IP address range
However, the following error message is received when using SSO to log into the company account:
IP XX.XXX.XX.XX is not allowed to access snowflake. Contact your local security administrator.
What steps should the Architect take to resolve this error and ensure that the account is accessed using only Private Link? (Choose two.)
A. Alter the Azure security integration to use the Private Link URLs.

B. Add the IP address in the error message to the allowed list in the network policy.
C. Generate a new SCIM access token using system$generate_scim_access_token and save it to Azure AD.
D. Update the configuration of the Azure AD SSO to use the Private Link URLs.
E. Open a case with Snowflake Support to authorize the Private Link URLs' access to the account.
Answer: (SHOW ANSWER)
The error message indicates that the IP address in the error message is not allowed to access Snowflake because it is not in the allowed list of the network policy. The network policy is a feature that allows restricting access to Snowflake based on IP addresses or ranges. To resolve this error, the Architect should take the following steps:
Add the IP address in the error message to the allowed list in the network policy. This will allow the IP address to access Snowflake using the Private Link URLs. Alternatively, the Architect can disable the network policy if it is not required for security reasons.
Update the configuration of the Azure AD SSO to use the Private Link URLs. This will ensure that the SSO authentication process uses the Private Link URLs instead of the public URLs. The configuration can be updated by following the steps in the Azure documentation1.
These two steps should resolve the error and ensure that the account is accessed using only Private Link. The other options are not necessary or relevant for this scenario. Altering the Azure security integration to use the Private Link URLs is not required because the security integration is used for SCIM provisioning, not for SSO authentication. Generating a new SCIM access token using system$generate_scim_access_token and saving it to Azure AD is not required because the SCIM access token is used for SCIM provisioning, not for SSO authentication. Opening a case with Snowflake Support to authorize the Private Link URLs' access to the account is not required because the authorization can be done by the account administrator using the SYSTEM$AUTHORIZE_PRIVATELINK function2.

NEW QUESTION: 74
What integration object should be used to place restrictions on where data may be exported?
A. Stage integration
B. Security integration
C. Storage integration
D. API integration
Answer: C (LEAVE A REPLY)
In Snowflake, a storage integration is used to define and configure external cloud storage that Snowflake will interact with. This includes specifying security policies for access control. One of the main features of storage integrations is the ability to set restrictions on where data may be exported. This is done by binding the storage integration to specific

cloud storage locations, thereby ensuring that Snowflake can only access those locations. It helps to maintain control over the data and complies with data governance and security policies by preventing unauthorized data exports to unspecified locations.

NEW QUESTION: 75
What are purposes for creating a storage integration? (Choose three.)
A. Control access to Snowflake data using a master encryption key that is maintained in the cloud provider's key management service.
B. Store a generated identity and access management (IAM) entity for an external cloud provider regardless of the cloud provider that hosts the Snowflake account.
C. Support multiple external stages using one single Snowflake object.
D. Avoid supplying credentials when creating a stage or when loading or unloading data.
E. Create private VPC endpoints that allow direct, secure connectivity between VPCs without traversing the public internet.
F. Manage credentials from multiple cloud providers in one single Snowflake object.
Answer: B,C,D (LEAVE A REPLY)
The purpose of creating a storage integration in Snowflake includes:B. Store a generated identity and access management (IAM) entity for an external cloud provider - This helps in managing authentication and authorization with external cloud storage without embedding credentials in Snowflake. It supports various cloud providers like AWS, Azure, or GCP, ensuring that the identity management is streamlined across platforms.C. Support multiple external stages using one single Snowflake object - Storage integrations allow you to set up access configurations that can be reused across multiple external stages, simplifying the management of external data integrations.D. Avoid supplying credentials when creating a stage or when loading or unloading data - By using a storage integration, Snowflake can interact with external storage without the need to continuously manage or expose sensitive credentials, enhancing security and ease of operations.References: Snowflake documentation on storage integrations, found within the SnowPro Advanced: Architect course materials.

NEW QUESTION: 76
An Architect has designed a data pipeline that Is receiving small CSV files from multiple sources. All of the files are landing in one location. Specific files are filtered for loading into Snowflake tables using the copy command. The loading performance is poor.
What changes can be made to Improve the data loading performance?
A. Increase the size of the virtual warehouse.
B. Create a multi-cluster warehouse and merge smaller files to create bigger files.
C. Create a specific storage landing bucket to avoid file scanning.
D. Change the file format from CSV to JSON.
Answer: B (LEAVE A REPLY)

According to the Snowflake documentation, the data loading performance can be improved by following some best practices and guidelines for preparing and staging the data files. One of the recommendations is to aim for data files that are roughly 100-250 MB (or larger) in size compressed, as this will optimize the number of parallel operations for a load. Smaller files should be aggregated and larger files should be split to achieve this size range. Another recommendation is to use a multi-cluster warehouse for loading, as this will allow for scaling up or out the compute resources depending on the load demand. A single-cluster warehouse may not be able to handle the load concurrency and throughput efficiently. Therefore, by creating a multi-cluster warehouse and merging smaller files to create bigger files, the data loading performance can be improved. References:
Data Loading Considerations
Preparing Your Data Files
Planning a Data Load

NEW QUESTION: 77
A Snowflake Architect is designing an application and tenancy strategy for an organization where strong legal isolation rules as well as multi-tenancy are requirements.
Which approach will meet these requirements if Role-Based Access Policies (RBAC) is a viable option for isolating tenants?
A. Create a multi-tenant table strategy if row level security is not viable for isolating tenants.
B. Create accounts for each tenant in the Snowflake organization.
C. Create an object for each tenant strategy if row level security is viable for isolating tenants.
D. Create an object for each tenant strategy if row level security is not viable for isolating tenants.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 78
A company needs to share its product catalog data with one of its partners. The product catalog data is stored in two database tables: product_category, and product_details. Both

tables can be joined by the product_id column. Data access should be governed, and only the partner should have access to the records.
The partner is not a Snowflake customer. The partner uses Amazon S3 for cloud storage.
Which design will be the MOST cost-effective and secure, while using the required Snowflake features?
A. Use Secure Data Sharing with an S3 bucket as a destination.
B. Publish product_category and product_details data sets on the Snowflake Marketplace.
C. Create a database user for the partner and give them access to the required data sets.
D. Create a reader account for the partner and share the data sets as secure views.
Answer: D (LEAVE A REPLY)
A reader account is a type of Snowflake account that allows external users to access data shared by a provider account without being a Snowflake customer. A reader account can be created and managed by the provider account, and can use the Snowflake web interface or JDBC/ODBC drivers to query the shared data. A reader account is billedto the provider account based on the credits consumed by the queries1. A secure view is a type of view that applies row-level security filters to the underlying tables, and masks the data that is not accessible to the user. A secure view can be shared with a reader account to provide granular and governed access to the data2. In this scenario, creating a reader account for the partner and sharing the data sets as secure views would be the most cost-effective and secure design, while using the required Snowflake features, because:
It would avoid the data transfer and storage costs of using an S3 bucket as a destination, and the potential security risks of exposing the data to unauthorized access or modification.
It would avoid the complexity and overhead of publishing the data sets on the Snowflake Marketplace, and the potential loss of control over the data ownership and pricing.
It would avoid the need to create a database user for the partner and grant them access to the required data sets, which would require the partner to have a Snowflake account and consume the provider's resources.
References:
Reader Accounts
Secure Views

NEW QUESTION: 79
What are characteristics of Dynamic Data Masking? (Select TWO).
A. A masking policy that Is currently set on a table can be dropped.
B. A single masking policy can be applied to columns in different tables.
C. A masking policy can be applied to the value column of an external table.
D. The role that creates the masking policy will always see unmasked data In query results
E. A masking policy can be applied to a column with the GEOGRAPHY data type.
Answer: (SHOW ANSWER)

Dynamic Data Masking is a feature that allows masking sensitive data in query results based on the role of the user who executes the query. A masking policy is a user-defined function that specifies the masking logic and can be applied to one or more columns in one or more tables. A masking policy that is currently set on a table can be dropped using the ALTER TABLE command. A single masking policy can be applied to columns in different tables usingthe ALTER TABLE command with the SET MASKING POLICY clause. The other options are either incorrect or not supported by Snowflake. A masking policy cannot be applied to the value column of an external table, as external tables do not support column-level security. The role that creates the masking policy will not always see unmasked data in query results, as the masking policy can be applied to the owner role as well. A masking policy cannot be applied to a column with the GEOGRAPHY data type, as Snowflake only supports masking policies for scalar data types. References: Snowflake Documentation:
Dynamic Data Masking, Snowflake Documentation: ALTER TABLE

NEW QUESTION: 80
The diagram shows the process flow for Snowpipe auto-ingest with Amazon Simple Notification Service (SNS) with the following steps:
Step 1: Data files are loaded in a stage.
Step 2: An Amazon S3 event notification, published by SNS, informs Snowpipe - by way of Amazon Simple Queue Service (SQS) - that files are ready to load. Snowpipe copies the files into a queue.
Step 3: A Snowflake-provided virtual warehouse loads data from the queued files into the target table based on parameters defined in the specified pipe.
If an AWS Administrator accidentally deletes the SQS subscription to the SNS topic in Step 2, what will happen to the pipe that references the topic to receive event messages from Amazon S3?
A. The pipe will continue to receive the messages as Snowflake will automatically restore the subscription to the same SNS topic and will recreate the pipe by specifying the same SNS topic name in the pipe definition.
B. The pipe will no longer be able to receive the messages and the user must wait for 24 hours from the time when the SNS topic subscription was deleted. Pipe recreation is not required as the pipe will reuse the same subscription to the existing SNS topic after 24 hours.
C. The pipe will continue to receive the messages as Snowflake will automatically restore the subscription by creating a new SNS topic. Snowflake will then recreate the pipe by specifying the new SNS topic name in the pipe definition.
D. The pipe will no longer be able to receive the messages. To restore the system immediately, the user needs to manually create a new SNS topic with a different name and then recreate the pipe by specifying the new SNS topic name in the pipe definition.
Answer: D (LEAVE A REPLY)

If an AWS Administrator accidentally deletes the SQS subscription to the SNS topic in Step 2, the pipe that references the topic to receive event messages from Amazon S3 will no longer be able to receive the messages.
This is because the SQS subscription is the link between the SNS topic and the Snowpipe notification channel.
Without the subscription, the SNS topic will not be able to send notifications to the Snowpipe queue, and the pipe will not be triggered to load the new files. To restore the system immediately, the user needs to manually create a new SNS topic with a different name and then recreate the pipe by specifying the new SNS topic name in the pipe definition. This will create a new notification channel and a new SQS subscription for the pipe. Alternatively, the user can also recreate the SQS subscription to the existing SNS topic and then alter the pipe to use the same SNS topic name in the pipe definition. This will also restore the notification channel and the pipe functionality. References:
Automating Snowpipe for Amazon S3
Enabling Snowpipe Error Notifications for Amazon SNS
HowTo: Configuration steps for Snowpipe Auto-Ingest with AWS S3 Stages

NEW QUESTION: 81
A new table and streams are created with the following commands:
CREATE OR REPLACE TABLE LETTERS (ID INT, LETTER STRING) ;
CREATE OR REPLACE STREAM STREAM_1 ON TABLE LETTERS;
CREATE OR REPLACE STREAM STREAM_2 ON TABLE LETTERS APPEND_ONLY = TRUE;
The following operations are processed on the newly created table:
INSERT INTO LETTERS VALUES (1, 'A');
INSERT INTO LETTERS VALUES (2, 'B');
INSERT INTO LETTERS VALUES (3, 'C');
TRUNCATE TABLE LETTERS;
INSERT INTO LETTERS VALUES (4, 'D');
INSERT INTO LETTERS VALUES (5, 'E');
INSERT INTO LETTERS VALUES (6, 'F');
DELETE FROM LETTERS WHERE ID = 6;
What would be the output of the following SQL commands, in order?
SELECT COUNT (*) FROM STREAM_1;
SELECT COUNT (*) FROM STREAM_2;
A. 2 & 6
B. 2 & 3
C. 4 & 3
D. 4 & 6
Answer: C (LEAVE A REPLY)

In Snowflake, a stream records data manipulation language (DML) changes to its base table since the stream was created or last consumed. STREAM_1 will show all changes including the TRUNCATE operation, while STREAM_2, being APPEND_ONLY, will not show deletions like TRUNCATE. Therefore, STREAM_1 will count the three inserts, the TRUNCATE (counted as a single operation), and the subsequent two inserts before the delete, totaling 4. STREAM_2 will only count the three initial inserts and the two after the TRUNCATE, totaling 3, as it does not count the TRUNCATE or the delete operation. References: The explanation is based on the Snowflake documentation on streams, which details how streams track changes and the difference between standard and APPEND_ONLY streams12.

NEW QUESTION: 82
What are some of the characteristics of result set caches? (Choose three.)
A. Time Travel queries can be executed against the result set cache.
B. Snowflake persists the data results for 24 hours.
C. Each time persisted results for a query are used, a 24-hour retention period is reset.
D. The data stored in the result cache will contribute to storage costs.
E. The retention period can be reset for a maximum of 31 days.
F. The result set cache is not shared between warehouses.
Answer: B,C,F (LEAVE A REPLY)
In Snowflake, the characteristics of result set caches include persistence of data results for 24 hours (B), each use of persisted results resets the 24-hour retention period (C), and result set caches are not shared between different warehouses (F). The result set cache is specifically designed to avoid repeated execution of the same query within this timeframe, reducing computational overhead and speeding up query responses. These caches do not contribute to storage costs, and their retention period cannot be extended beyond the default duration nor up to 31 days, as might be misconstrued.References: Snowflake Documentation on Result Set Caching.