ARA-C01 Q&As

SnowPro Advanced: Architect Certification Exam

Pass Snowflake ARA-C01 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass2lead.com/ara-c01.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Snowflake Official Exam Center

QUESTION 1

Which organization-related tasks can be performed by the ORGADMIN role? (Choose three.)

A. Changing the name of the organization

B. Creating an account

C. Viewing a list of organization accounts

D. Changing the name of an account

E. Deleting an account

F. Enabling the replication of a database

Correct Answer: BCF

Explanation: According to the SnowPro Advanced: Architect documents and learning resources, the organization-related tasks that can be performed by the ORGADMIN role are: Creating an account in the organization. A user with the ORGADMIN role can use the CREATE ACCOUNT command to create a new account that belongs to the same organization as the current account1. Viewing a list of organization accounts. A user with the ORGADMIN role can use the SHOW ORGANIZATION ACCOUNTS command to view the names and properties of all accounts in the organization2. Alternatively, the user can use the Admin ?Accounts page in the web interface to view the organization name and account names3. Enabling the replication of a database. A user with the ORGADMIN role can use the SYSTEM$GLOBAL_ACCOUNT_SET_PARAMETER function to enable database replication for an account in the organization. This allows the user to replicate databases across accounts in different regions and cloud platforms for data availability and durability4. The other options are incorrect because they are not organization-related tasks that can be performed by the ORGADMIN role. Option A is incorrect because changing the name of the organization is not a task that can be performed by the ORGADMIN role. To change the name of an organization, the user must contact Snowflake Support3. Option D is incorrect because changing the name of an account is not a task that can be performed by the ORGADMIN role. To change the name of an account, the user must contact Snowflake Support5. Option E is incorrect because deleting an account is not a task that can be performed by the ORGADMIN role. To delete an account, the user must contact Snowflake Support. References: CREATE ACCOUNT | Snowflake Documentation, SHOW ORGANIZATION ACCOUNTS | Snowflake Documentation, Getting Started with Organizations | Snowflake Documentation, SYSTEM$GLOBAL_ACCOUNT_SET_PARAMETER | Snowflake Documentation, ALTER ACCOUNT | Snowflake Documentation, [DROP ACCOUNT | Snowflake Documentation]

QUESTION 2

How can an Architect enable optimal clustering to enhance performance for different access paths on a given table?

A. Create multiple clustering keys for a table.

B. Create multiple materialized views with different cluster keys.

C. Create super projections that will automatically create clustering.

D. Create a clustering key that contains all columns used in the access paths.

Correct Answer: B

Explanation: According to the SnowPro Advanced: Architect documents and learning resources, the best way to enable

optimal clustering to enhance performance for different access paths on a given table is to create multiple materialized views with different cluster keys. A materialized view is a pre-computed result set that is derived from a query on one or more base tables. A materialized view can be clustered by specifying a clustering key, which is a subset of columns or expressions that determines how the data in the materialized view is co-located in micro-partitions. By creating multiple materialized views with different cluster keys, an Architect can optimize the performance of queries that use different access paths on the same base table. For example, if a base table has columns A, B, C, and D, and there are queries that filter on A and B, or on C and D, or on A and C, the Architect can create three materialized views, each with a different cluster key: (A, B), (C, D), and (A, C). This way, each query can leverage the optimal clustering of the corresponding materialized view and achieve faster scan efficiency and better compression. References: Snowflake Documentation: Materialized Views Snowflake Learning: Materialized Views https://www.snowflake.com/blog/using-materialized-views-to-solve-multi-clustering- performance-problems/

QUESTION 3

A company wants to deploy its Snowflake accounts inside its corporate network with no visibility on the internet. The company is using a VPN infrastructure and Virtual Desktop Infrastructure (VDI) for its Snowflake users. The company also wants to re-use the login credentials set up for the VDI to eliminate redundancy when managing logins.

What Snowflake functionality should be used to meet these requirements? (Choose two.)

A. Set up replication to allow users to connect from outside the company VPN.

B. Provision a unique company Tri-Secret Secure key.

C. Use private connectivity from a cloud provider.

D. Set up SSO for federated authentication.

E. Use a proxy Snowflake account outside the VPN, enabling client redirect for user logins.

Correct Answer: CD

Explanation: According to the SnowPro Advanced: Architect documents and learning resources, the Snowflake functionality that should be used to meet these requirements are: Use private connectivity from a cloud provider. This feature allows customers to connect to Snowflake from their own private network without exposing their data to the public Internet. Snowflake integrates with AWS PrivateLink, Azure Private Link, and Google Cloud Private Service Connect to offer private connectivity from customers\\' VPCs or VNets to Snowflake endpoints. Customers can control how traffic reaches the Snowflake endpoint and avoid the need for proxies or public IP addresses123. Set up SSO for federated authentication. This feature allows customers to use their existing identity provider (IdP) to authenticate users for SSO access to Snowflake. Snowflake supports most SAML 2.0-compliant vendors as an IdP, including Okta, Microsoft AD FS, Google G Suite, Microsoft Azure Active Directory, OneLogin, Ping Identity, and PingOne. By setting up SSO for federated authentication, customers can leverage their existing user credentials and profile information, and provide stronger security than username/password authentication4. The other options are incorrect because they do not meet the requirements or are not feasible. Option A is incorrect because setting up replication does not allow users to connect from outside the company VPN. Replication is a feature of Snowflake that enables copying databases across accounts in different regions and cloud platforms. Replication does not affect the connectivity or visibility of the accounts5. Option B is incorrect because provisioning a unique company Tri-Secret Secure key does not affect the network or authentication requirements. Tri-Secret Secure is a feature of Snowflake that 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. Tri- Secret Secure provides an additional layer of security and control over the data encryption and decryption process, but it does not enable private connectivity or SSO6. Option E is incorrect because using a proxy Snowflake account outside the VPN, enabling client redirect for user logins, is not a supported or recommended way of meeting the requirements. Client redirect is a feature of Snowflake that allows customers to connect to a different Snowflake account than the one specified in the connection string. This feature is useful for scenarios such as cross-region failover, data sharing, and account migration, but it does not provide private connectivity

or SSO7. References: AWS PrivateLink and Snowflake | Snowflake Documentation, Azure Private Link and Snowflake | Snowflake Documentation, Google Cloud Private Service Connect and Snowflake | Snowflake Documentation, Overview of Federated Authentication and SSO | Snowflake Documentation, Replicating Databases Across Multiple Accounts | Snowflake Documentation, Tri-Secret Secure | Snowflake Documentation, Redirecting Client Connections | Snowflake Documentation

QUESTION 4

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

Correct Answer: A

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]

QUESTION 5

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.

Correct Answer: AC

Explanation: These two options are the best ways to meet the requirement of loading data from an external stage and making it accessible by dashboards with the least amount of coding. Snowpipe with auto-ingest is a feature that enables continuous and automated data loading from an external stage into a Snowflake table. Snowpipe uses event notifications from the cloud storage service to detect new or modified files in the stage and triggers a COPY INTO command to load the data into the table. Snowpipe is efficient, scalable, and serverless, meaning it does not require any infrastructure or maintenance from the user. Snowpipe also supports loading data from files of any size, as long as they are in a supported format1. A materialized view on an external table is a feature that enables creating a pre- computed result set from an external table and storing it in Snowflake. A materialized view can improve the performance and efficiency of querying data from an external table, especially for complex queries or dashboards. A materialized view can also support aggregations, joins, and filters on the external table data. A materialized view on an external table is automatically refreshed when the underlying data in the external stage changes, as long as the AUTO_REFRESH parameter is set to true2. References: Snowpipe Overview | Snowflake Documentation Materialized Views on External Tables | Snowflake Documentation

QUESTION 6

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.

Correct Answer: BCE

Explanation: 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 incur any 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

QUESTION 7

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);

Correct Answer: C

Explanation: This is the correct command to create a clone of the table Data as it was 5 minutes ago using cloning and Time Travel. Cloning is a feature that allows creating a copy of a database, schema, table, or view without duplicating the

data or metadata. Time Travel is a feature that enables accessing historical data (i.e. data that has been changed or deleted) at any point within a defined period. To create a clone of a table at a point in time in the past, the syntax is:

CREATE TABLE CLONE AT (OFFSET =>);

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

QUESTION 8

What is a characteristic of loading data into Snowflake using the Snowflake Connector for Kafka?

A. The Connector only works in Snowflake regions that use AWS infrastructure.

B. The Connector works with all file formats, including text, JSON, Avro, Ore, Parquet, and XML.

C. The Connector creates and manages its own stage, file format, and pipe objects.

D. Loads using the Connector will have lower latency than Snowpipe and will ingest data in real time.

Correct Answer: C

Explanation: According to the SnowPro Advanced: Architect documents and learning resources, a characteristic of loading data into Snowflake using the Snowflake Connector for Kafka is that the Connector creates and manages its own stage, file format, and pipe objects. The stage is an internal stage that is used to store the data files from the Kafka topics. The file format is a JSON or Avro file format that is used to parse the data files. The pipe is a Snowpipe object that is used to load the data files into the Snowflake table. The Connector automatically creates and configures these objects based on the Kafka configuration properties, and handles the cleanup and maintenance of these objects1. The other options are incorrect because they are not characteristics of loading data into Snowflake using the Snowflake Connector for Kafka. Option A is incorrect because the Connector works in Snowflake regions that use any cloud infrastructure, not just AWS. The Connector supports AWS, Azure, and Google Cloud platforms, and can load data across different regions and cloud platforms using data replication2. Option B is incorrect because the Connector does not work with all file formats, only JSON and Avro. The Connector expects the data in the Kafka topics to be in JSON or Avro format, and parses the data accordingly. Other file formats, such as text, ORC, Parquet, or XML, are not supported by the Connector3. Option D is incorrect because loads using the Connector do not have lower latency than Snowpipe, and do not ingest data in real time. The Connector uses Snowpipe to load data into Snowflake, and inherits the same latency and performance characteristics of Snowpipe. The Connector does not provide real-time ingestion, but near real-time ingestion, depending on the frequency and size of the data files4. References: Installing and Configuring the Kafka Connector | Snowflake Documentation, Sharing Data Across Regions and Cloud Platforms | Snowflake Documentation, Overview of the Kafka Connector | Snowflake Documentation, Using Snowflake Connector for Kafka With Snowpipe Streaming | Snowflake Documentation

QUESTION 9

Which of the following are characteristics of how row access policies can be applied to external tables? (Choose three.)

A. An external table can be created with a row access policy, and the policy can be applied to the VALUE column.

B. A row access policy can be applied to the VALUE column of an existing external table.

C. A row access policy cannot be directly added to a virtual column of an external table.

D. External tables are supported as mapping tables in a row access policy.

E. While cloning a database, both the row access policy and the external table will be cloned.

F. A row access policy cannot be applied to a view created on top of an external table.

Correct Answer: ABC

Explanation: These three statements are true according to the Snowflake documentation and the web search results. A row access policy is a feature that allows filtering rows based on user-defined conditions. A row access policy can be applied to an external table, which is a table that reads data from external files in a stage. However, there are some limitations and considerations for using row access policies with external tables. An external table can be created with a row access policy by using the WITH ROW ACCESS POLICY clause in the CREATE EXTERNAL TABLE statement. The policy can be applied to the VALUE column, which is the column that contains the raw data from the external files in a VARIANT data type1. A row access policy can also be applied to the VALUE column of an existing external table by using the ALTER TABLE statement with the SET ROW ACCESS POLICY clause2. A row access policy cannot be directly added to a virtual column of an external table. A virtual column is a column that is derived from the VALUE column using an expression. To apply a row access policy to a virtual column, the policy must be applied to the VALUE column and the expression must be repeated in the policy definition3. External tables are not supported as mapping tables in a row access policy. A mapping table is a table that is used to determine the access rights of users or roles based on some criteria. Snowflake does not support using an external table as a mapping table because it may cause performance issues or errors4. While cloning a database, Snowflake clones the row access policy, but not the external table. Therefore, the policy in the cloned database refers to a table that is not present in the cloned database. To avoid this issue, the external table must be manually cloned or recreated in the cloned database4. A row access policy can be applied to a view created on top of an external table. The policy can be applied to the view itself or to the underlying external table. However, if the policy is applied to the view, the view must be a secure view, which is a view that hides the underlying data and the view definition from unauthorized users5. References: CREATE EXTERNAL TABLE | Snowflake Documentation ALTER EXTERNAL TABLE | Snowflake Documentation Understanding Row Access Policies | Snowflake Documentation Snowflake Data Governance: Row Access Policy Overview Secure Views | Snowflake Documentation

QUESTION 10

Which of the following are characteristics of Snowflake\\'s parameter hierarchy?

A. Session parameters override virtual warehouse parameters.

B. Virtual warehouse parameters override user parameters.

C. Table parameters override virtual warehouse parameters.

D. Schema parameters override account parameters.

Correct Answer: D

Explanation: This is the correct answer because it reflects the characteristics of Snowflake\\'s parameter hierarchy. Snowflake provides three types of parameters that can be set for an account: account parameters, session parameters, and object parameters. All parameters have default values, which can be set and then overridden at different levels depending on the parameter type. The following diagram illustrates the hierarchical relationship between the different parameter types and how individual parameters can be overridden at each level1: As shown in the diagram, schema parameters are a type of object parameters that can be set for schemas. Schema parameters can override the account parameters that are set at the account level. For example, the LOG_LEVEL parameter can be set at the account level to control the logging level for all objects in the account, but it can also be overridden at the schema level to control the

logging level for specific stored procedures and UDFs in that schema2. The other options listed are not correct because they do not reflect the characteristics of Snowflake\\'s parameter hierarchy. Session parameters do not override virtual warehouse parameters, because virtual warehouse parameters are a type of session parameters that can be set for virtual warehouses. Virtual warehouse parameters do not override user parameters, because user parameters are a type of session parameters that can be set for users. Table parameters do not override virtual warehouse parameters, because table parameters are a type of object parameters that can be set for tables, and object parameters do not affect session parameters1. References: Snowflake Documentation: Parameters Snowflake Documentation: Setting Log Level

QUESTION 11

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.

Correct Answer: C

Explanation: 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

QUESTION 12

Which statements describe characteristics of the use of materialized views in Snowflake? (Choose two.)

A. They can include ORDER BY clauses.

B. They cannot include nested subqueries.

C. They can include context functions, such as CURRENT_TIME().

D. They can support MIN and MAX aggregates.

E. They can support inner joins, but not outer joins.

Correct Answer: BD

Explanation: According to the Snowflake documentation, materialized views have some limitations on the query specification that defines them. One of these limitations is that they cannot include nested subqueries, such as subqueries in the FROM clause or scalar subqueries in the SELECT list. Another limitation is that they cannot include ORDER BY clauses, context functions (such as CURRENT_TIME()), or outer joins. However, materialized views can support MIN and MAX aggregates, as well as other aggregate functions, such as SUM, COUNT, and AVG. References: Limitations on Creating Materialized Views | Snowflake Documentation Working with Materialized Views | Snowflake Documentation

QUESTION 13

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

Correct Answer: B

Explanation: According to the SnowPro Advanced: Architect documents and learning resources, the integration object that should be used to place restrictions on where data may be exported is the security integration. A security integration is a Snowflake object that provides an interface between Snowflake and third-party security services, such as Okta, Duo, or Google Authenticator. A security integration can be used to enforce policies on data export, such as requiring multi-factor authentication (MFA) or restricting the export destination to a specific network or domain. A security integration can also be used to enable single sign-on (SSO) or federated authentication for Snowflake users1. The other options are incorrect because they are not integration objects that can be used to place restrictions on where data may be exported. Option A is incorrect because a stage integration is not a valid type of integration object in Snowflake. A stage is a Snowflake object that references a location where data files are stored, such as an internal stage, an external stage, or a named stage. A stage is not an integration object that provides an interface between Snowflake and third-party services2. Option C is incorrect because a storage integration is a Snowflake object that provides an interface between Snowflake and external cloud storage, such as Amazon S3, Azure Blob Storage, or Google Cloud Storage. A storage integration can be used to securely access data files from external cloud storage without exposing the credentials, but it cannot be used to place restrictions on where data may be exported3. Option D is incorrect because an API integration is a Snowflake object thatprovides an interface between Snowflake and third-party services that use REST APIs, such as Salesforce, Slack, or Twilio. An API integration can be used to securely call external REST APIs from Snowflake using the CALL_EXTERNAL_API function, but it cannot be used to place restrictions on where data may be exported4. References: CREATE SECURITY INTEGRATION | Snowflake Documentation, CREATE STAGE | Snowflake Documentation, CREATE STORAGE INTEGRATION | Snowflake Documentation, CREATE API INTEGRATION | Snowflake Documentation

QUESTION 14

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.

Correct Answer: BCD

A storage integration is a Snowflake object that stores a generated identity and access management (IAM) entity for an external cloud provider, such as Amazon S3, Google Cloud Storage, or Microsoft Azure Blob Storage. This integration allows Snowflake to read data from and write data to an external storage location referenced in an external stage1. One purpose of creating a storage integration is to support multiple external stages using one single Snowflake object. An integration can list buckets (and optional paths) that limitthe locations users can specify when creating external stages that use the integration. Note that many external stage objects can reference different buckets and paths and use the same storage integration for authentication1. Therefore, option C is correct. Another purpose of creating a storage integration is to avoid supplying credentials when creating a stage or when loading or unloading data. Integrations are named, first-class Snowflake objects that avoid the need for passing explicit cloud provider credentials such as secret keys or access tokens. Integration objects store an IAM user ID, and an administrator in your organization grants the IAM user permissions in the cloud provider account1. Therefore, option D is correct. A third purpose of creating a storage integration is to store a generated IAM entity for an external cloud provider regardless of the cloud provider that hosts the Snowflake account. For example, you can create a storage integration for Amazon S3 even if your Snowflake account is hosted on Azure or Google Cloud Platform. This allows you to access data across different cloud platforms using Snowflake1. Therefore, option B is correct. Option A is incorrect, because creating a storage integration does not control access to Snowflake data using a master encryption key. Snowflake encrypts all data using a hierarchical key model, and the master encryption key is managed by Snowflake or by the customer using a cloud provider\\'s key management service. This is independent of the storage integration feature2. Option E is incorrect, because creating a storage integration does not create private VPC endpoints. Private VPC endpoints are a network configuration option that allow direct, secure connectivity between VPCs without traversing the public internet. This is also independent of the storage integration feature3. Option F is incorrect, because creating a storage integration does not manage credentials from multiple cloud providers in one single Snowflake object. A storage integration is specific to one cloud provider, and you need to create separate integrations for each cloud provider you want to access4. References: : Encryption and Decryption : Private Link for Snowflake : CREATE STORAGE INTEGRATION : Option 1: Configuring a Snowflake Storage Integration to Access Amazon S3

QUESTION 15

Which Snowflake data modeling approach is designed for BI queries?

A. 3 NF

B. Star schema

C. Data Vault

D. Snowflake schema

Correct Answer: B

Explanation: 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