EXAM HEIST

EXAM HEIST

Premium exam material

Get certification quickly with the Exam Heist Premium exam material.
Everything you need to prepare, learn & pass your certification exam easily. Lifetime free updates

First attempt guaranteed success.

https://www.ExamHeist.com

Snowflake

(SnowPro Core)

SnowPro Core

Total: **1254 Questions**

Link: https://examheist.com/papers/snowflake/snowpro-core

Question: 1 Exam Heist

Snowflake provides a mechanism for its customers to override its natural clustering algorithms. This method is:

A. Micro-partitions
B. Clustering keys
C. Key partitions
D. Clustered partitions

Answer: B

Explanation:

The correct answer is **B. Clustering keys**. Snowflake's natural data organization involves micro-partitions, which are small, immutable storage units. While Snowflake automatically optimizes these micro-partitions for performance, users have the option to exert more control using clustering keys. Clustering keys are columns or expressions specified on a table that Snowflake uses to reorganize data within micro-partitions, physically co-locating rows with similar values. This custom clustering enables more efficient data retrieval and reduces the amount of data that needs to be scanned for specific queries. This is fundamentally a performance optimization technique, overriding Snowflake's default clustering based on insertion order. Options A, C, and D are incorrect. Micro-partitions are the underlying storage method, not a user-defined override mechanism. Key partitions and clustered partitions are not standard Snowflake terms. Clustering keys offer users a way to tailor Snowflake's data layout to their unique query patterns, allowing for targeted data optimization and improved performance.

For further information, refer to the official Snowflake documentation:

Understanding Snowflake Table Structures
Clustering Keys
Automatic Clustering

Question: 2 Exam Heist

Which of the following are valid Snowflake Virtual Warehouse Scaling Policies? (Choose two.)

A. Custom
B. Economy
C. Optimized
D. Standard

Answer: BD

Explanation:

The correct scaling policies for Snowflake Virtual Warehouses are **Standard** and **Economy**. Let's explore why:

Snowflake's Virtual Warehouses, which are essentially compute clusters, can automatically adjust their size (scaling) based on workload demands. This scalability ensures optimal performance and cost efficiency. The scaling policy dictates how the warehouse increases or decreases its compute resources.

The **Standard** scaling policy is the default setting. It aims to balance performance and cost. Snowflake will automatically scale up the warehouse when necessary to handle increased query load and then scale down when demand decreases. This approach is ideal for most workloads that need both responsiveness and reasonable cost management.

The **Economy** scaling policy, on the other hand, prioritizes cost savings over performance. Snowflake will scale up more slowly, and scale down more aggressively than with the standard policy. This makes it suitable for batch jobs, data loading processes, and other activities that can tolerate slightly longer processing times. It optimizes for budget when performance is not the top concern.

Options A ("Custom") and C ("Optimized") are not recognized scaling policies in Snowflake's terminology. While customization is possible through other configurations and resource management practices, these are not specific predefined policies as provided by Snowflake.

Therefore, selecting **Standard** and **Economy** correctly identifies the two valid Snowflake Virtual Warehouse scaling policies. Understanding these policies is essential for efficiently managing resources and optimizing performance within the Snowflake platform.

For detailed information and verification, refer to the official Snowflake documentation:

Snowflake Documentation on Virtual Warehouse Scaling Policies: https://docs.snowflake.com/en/user-guide/warehouses-scaling.html
Snowflake Documentation on Warehouse Overview: https://docs.snowflake.com/en/user-guide/warehouses.html

Question: 3 Exam Heist

True or False: A single database can exist in more than one Snowflake account.

A. True
B. False

Answer: B

Explanation:

The statement is false. In Snowflake, a database is a logical grouping of schemas, tables, and other database objects, and it exists within the confines of a single Snowflake account. A Snowflake account is a distinct, isolated instance of the Snowflake service, characterized by its unique URL and user credentials. Each account has its own independent storage, compute resources, and configurations. While you can share data across different Snowflake accounts using features like data sharing, the underlying database itself remains exclusive to the account in which it was created. The core principle of resource isolation in cloud environments, especially in data warehousing like Snowflake, prevents a single database from spanning multiple independent account contexts. This ensures security, prevents conflicts, and simplifies management. Attempting to access a database outside its originating account would require explicitly granted sharing permissions, not direct access as if the database was available in both accounts. Data sharing techniques allow data from a database in one account to be used by queries in another account, but not for direct manipulation of the original database objects from a different account. A similar model is also followed by other cloud data warehouses like BigQuery and Redshift.

For further reading:

Snowflake Account Concepts: https://docs.snowflake.com/en/user-guide/intro-key-concepts.html#accounts
Snowflake Data Sharing: https://docs.snowflake.com/en/user-guide/data-sharing-intro.html

Question: 4 Exam Heist

Which of the following roles is recommended to be used to create and manage users and roles?

A. SYSADMIN
B. SECURITYADMIN
C. PUBLIC
D. ACCOUNTADMIN

Answer: B

Explanation:

The correct answer is B, SECURITYADMIN. In Snowflake's role-based access control (RBAC) model, different roles are designed for specific administrative tasks, promoting the principle of least privilege. While ACCOUNTADMIN possesses the highest level of permissions and could perform user and role management, it's crucial to limit its use for day-to-day tasks, thereby reducing the risk associated with its broad powers. SYSADMIN has the system privileges for managing warehouse, database objects and other system related operations but does not have the access to user and role management. PUBLIC role is automatically granted to every user and is not suitable for administration tasks. The SECURITYADMIN role is specifically designed to manage user and role creation, assignment, and grant privileges. This specialization allows for a separation of duties, enhancing security and auditability. Using SECURITYADMIN for user and role administration allows for better control and tracking of who has access and what changes are made to user permissions. Adhering to best practices involves granting only the necessary privileges to each role, ensuring that the highest privilege role is reserved for exceptional circumstances, as per the principle of least privilege.

For more information, refer to the Snowflake documentation on access control and roles:

Snowflake Access Control
Understanding Snowflake Roles

Question: 5 Exam Heist

True or False: Bulk unloading of data from Snowflake supports the use of a SELECT statement.

A. True
B. False

Answer: A

Explanation:

The statement "Bulk unloading of data from Snowflake supports the use of a SELECT statement" is indeed true. Snowflake's COPY INTO command, which facilitates bulk data unloading, allows you to specify a SELECT statement. This SELECT query defines the specific data you wish to extract from your Snowflake tables or views. The results of this query are then exported into files located on cloud storage services like AWS S3, Azure Blob Storage, or Google Cloud Storage. This functionality provides granular control over the data being unloaded, enabling you to filter, transform, and project the data according to your specific needs before it's written to the external location. The use of a SELECT statement allows for the creation of subsets of data based on specific criteria, improving efficiency and reducing the overall size of exported files when only partial data is needed. It contrasts with simply unloading entire tables, providing much more flexibility. The underlying principle behind using a SELECT statement aligns with the concept of data extraction in data warehousing, which involves retrieving tailored data from a data repository to support downstream processes. The capability to include complex queries, joins, and subqueries in the SELECT statement allows for the retrieval of intricately structured and transformed data during the unload process. The flexibility offered by using the SELECT statement within the unload process mirrors the power of SQL, the foundational language for manipulating data in databases like Snowflake.

Authoritative links for further research:

Snowflake Documentation - COPY INTO : https://docs.snowflake.com/en/sql-reference/sql/copy-into-location.html (Specifically examine the section on specifying a select statement as the data source)
Snowflake Blog - Data Unloading: Search for related articles on data unloading best practices on the official Snowflake blog.

Question: 6 Exam Heist

Select the different types of Internal Stages: (Choose three.)

A. Named Stage
B. User Stage
C. Table Stage
D. Schema Stage

Answer: ABC

Explanation:

The correct answer, ABC, identifies the three distinct types of internal stages in Snowflake: Named Stages, User Stages, and Table Stages. Internal stages are storage locations within Snowflake's managed environment used for storing data files.

Named Stages (A): These are explicitly created database objects that can be referenced by name across multiple users and sessions. They offer granular access control and are suitable for storing data meant for various loading and unloading tasks. They are persistent and require explicit creation, modification, and deletion.

User Stages (B): Each Snowflake user has a personal, automatically created stage associated with their user account. This stage is for temporary files and is convenient for quick, personal data loading tasks. It's accessible only to the user owning the stage, making it secure for individual work.

Table Stages (C): Every Snowflake table has an implicitly defined table stage. It is directly related to the specific table and acts as a staging area for data intended to be loaded into or unloaded from that table. This stage is primarily used during data loading/unloading operations connected to a single table.

Schema Stages (D) do not exist as a defined type of internal stage. Although a schema contains the metadata for database objects, it doesn't inherently serve as a storage location like the other three stage types.

Therefore, the internal stages of Snowflake are limited to the explicitly created Named Stages, automatically generated User Stages, and implicitly created Table Stages.

Authoritative Links:

1. Snowflake Documentation on Internal Stages: https://docs.snowflake.com/en/user-guide/data-load-local-file-system-stages.html
2. Snowflake Documentation on Stages: https://docs.snowflake.com/en/sql-reference/sql/create-stage.html
3. Snowflake Documentation on User Stages: https://docs.snowflake.com/en/user-guide/data-load-local-file-system-stages.html#user-stages
4. Snowflake Documentation on Table Stages: https://docs.snowflake.com/en/user-guide/data-load-local-file-system-stages.html#table-stages

Question: 7 Exam Heist

True or False: A customer using SnowSQL / native connectors will be unable to also use the Snowflake Web Interface (UI) unless access to the UI is explicitly granted by support.

A. True
B. False

Answer: B

Explanation:

The statement is false. Using SnowSQL or native connectors to interact with Snowflake does not inherently restrict access to the Snowflake Web Interface (UI). Snowflake allows concurrent access from multiple interfaces without requiring explicit support intervention. Different client tools, such as SnowSQL, connectors, and the Web UI, leverage the same Snowflake backend infrastructure. These interfaces simply act as different access points to the same data and functionality. A user can be logged into both the Web UI and use SnowSQL simultaneously, assuming they have the necessary credentials and permissions for both. This seamless multi-access capability is a core design principle of modern cloud data platforms like Snowflake. Access control is handled via roles and privileges managed within Snowflake itself, not through restrictions based on the client used. There's no inherent limitation preventing users from leveraging different tools concurrently. Support does not need to grant permission for web UI access if a user is already using SnowSQL or connectors.

Authoritative links for further research:

Snowflake Documentation on Access Control: https://docs.snowflake.com/en/user-guide/security-access-control-overview
Snowflake Documentation on Connecting to Snowflake: https://docs.snowflake.com/en/user-guide/connecting
Snowflake Documentation on Web Interface:https://docs.snowflake.com/en/user-guide/ui-web

Question: 8 Exam Heist

Account-level storage usage can be monitored via:

A. The Snowflake Web Interface (UI) in the Databases section
B. The Snowflake Web Interface (UI) in the Account -> Billing & Usage section
C. The Information Schema -> ACCOUNT_USAGE_HISTORY View
D. The Account Usage Schema -> ACCOUNT_USAGE_METRICS View

Answer: B

Explanation:

The correct answer is B: The Snowflake Web Interface (UI) in the Account -> Billing & Usage section. Snowflake provides a dedicated section within its user interface for monitoring account-level billing and usage metrics, including storage. This is where administrators can obtain an overview of consumed storage, compute, and other resources. Option A, the Databases section, focuses on the storage used by individual databases and schemas rather than the overall account storage consumption. The Information Schema, as mentioned in option C, and the Account Usage schema, referred to in option D, offer access to storage history through views and tables. However, these are accessed via SQL queries and are not the primary point of entry for a general overview of storage usage via the web interface. While these schema provide detailed historical data, the Billing & Usage section provides a user-friendly, graphical interface for quickly understanding the

current state and trend of resource usage at the account level. Accessing the web interface and navigating to Account -> Billing & Usage is the most direct and convenient way to obtain an account-level storage overview without having to query system views. This approach aligns with typical cloud service interfaces, providing visual reports for easy comprehension. In essence, this section presents an aggregate view of storage consumption across all databases in an account.

Relevant Link:Snowflake Documentation - Monitoring Usage

Question: 9 Exam Heist

Credit Consumption by the Compute Layer (Virtual Warehouses) is based on: (Choose two.)

A. Number of users
B. Warehouse size
C. Amount of data processed
D. # of clusters for the Warehouse

Answer: BD

Explanation:

The correct answer is **B. Warehouse size and D. # of clusters for the Warehouse**. Snowflake's compute layer, which is composed of virtual warehouses, consumes credits based on the resources used, primarily compute power and time. Warehouse size directly impacts credit consumption. Larger warehouses, which provide more compute resources, consume credits at a higher rate per unit of time compared to smaller warehouses. This is because a larger size means more CPU and memory allocated, translating to greater processing capability and therefore a higher charge. The number of clusters also plays a critical role, as a multi-cluster warehouse will consume more credits than a single-cluster one. Each active cluster within a multi-cluster warehouse incurs charges in proportion to its size and the duration it's running. The amount of data processed (option C) affects the duration a warehouse runs, but the actual rate of credit consumption is determined by the warehouse's size and the number of active clusters. The number of users (option A) is irrelevant to direct credit consumption. While user activity drives warehouse usage, it's the compute resources that consume credits, not the users themselves. A warehouse can be idle and not consuming credits even with many connected users. In summary, Snowflake charges for the capacity it dedicates (warehouse size and clusters) and the duration the resources are active, not necessarily for the amount of data or connected users.

Further research can be done through these official Snowflake links:

Understanding Snowflake Credits: This page provides a comprehensive overview of Snowflake credits, how they are used, and how usage is calculated.
Virtual Warehouse Cost: This page specifically details the cost implications of virtual warehouses and how their size and multi-cluster settings affect consumption.
Overview of Virtual Warehouses: Learn the fundamentals of how virtual warehouses work in Snowflake.

Question: 10 Exam Heist

Which statement best describes `clustering`?

A.Clustering represents the way data is grouped together and stored within Snowflake's micro-partitions
B.The database administrator must define the clustering methodology for each Snowflake table
C.The clustering key must be included on the COPY command when loading data into Snowflake

D.Clustering can be disabled within a Snowflake account

Answer: A

Explanation:

The correct answer is A. Clustering in Snowflake refers to how data is physically organized within micro-partitions, which are the fundamental storage units. Micro-partitions automatically divide tables and store them internally. Snowflake automatically manages the creation and maintenance of micro-partitions. Option A accurately describes this core functionality. Clustering, in the Snowflake context, aims to optimize query performance by grouping related data together. When a query filters based on clustering key columns, Snowflake can efficiently retrieve only the relevant micro-partitions, reducing the amount of data scanned.

Option B is incorrect because while you can define a clustering key on a table, Snowflake automatically manages the organization of data within micro-partitions. It does not require that database administrator to design a manual clustering method for each table. Option C is false; the clustering key is a metadata concept associated with the table definition and is not directly involved in the COPY command. The COPY command simply loads data into the table. Option D is incorrect as clustering is a core mechanism and cannot be disabled at the account level.

Therefore, the best description of clustering in Snowflake is found in option A: how data is organized and grouped within Snowflake's micro-partitions, enabling efficient querying.

For further research, explore the official Snowflake documentation:Clustering KeysMicro-partitions

Question: 11 Exam Heist

True or False: The COPY command must specify a File Format in order to execute.

A.True

B.False

Answer: B

Explanation:

The statement that the COPY command in Snowflake must specify a File Format is **False**. While a file format is highly recommended and typically used, it's not a mandatory requirement for all COPY operations. The COPY command can function without explicitly specifying a file format if certain default behaviors are applicable or if the file type is implicitly detected by Snowflake. For instance, if you're loading data from stage files with simple, comma-separated values (.csv) and the default settings for CSV file format work, you do not need to specify it. However, for diverse file types like JSON, XML, Avro, etc. and/or for data that requires custom delimiters, escape characters or other transformations, explicit file format specification becomes crucial. The best practice is to always explicitly state the file format to maintain clarity, avoid misinterpretations, and handle various file and data structures accurately. The absence of file format specifications might lead to unpredictable outcomes, especially with complex datasets. Relying on implicit detection should be approached cautiously and tested thoroughly before production deployments. Snowflake does however require a stage to be specified, and the stage has an associated file format. But, the command itself does not need the FILE_FORMAT.

Snowflake Documentation on COPY CommandSnowflake Documentation on File Formats

Question: 12 Exam Heist

Which of the following commands sets the Virtual Warehouse for a session?

A.COPY WAREHOUSE FROM <<config file>>;
B.SET WAREHOUSE = <<warehouse name>>;
C.USE WAREHOUSE <<warehouse name>>;
D.USE VIRTUAL_WAREHOUSE <<warehouse name>>;

Answer: C

Explanation:

The correct answer is C, USE WAREHOUSE <<warehouse name>>;. In Snowflake, virtual warehouses are the compute engines that power queries. To execute queries using a specific warehouse, you must explicitly set it for your current session. The USE WAREHOUSE command achieves this, effectively directing subsequent queries to the specified warehouse. This command operates on a session level, meaning the selected warehouse will remain active for the duration of the current session unless changed again. Option A, COPY WAREHOUSE FROM <<config file>>;, is incorrect as there's no such command for copying warehouse settings. Option B, SET WAREHOUSE = <<warehouse name>>;, attempts to use a SET command which is not designed for directly specifying an active warehouse; this is more aligned with setting session-level parameters, not the active compute resource. Option D, USE VIRTUAL_WAREHOUSE <<warehouse name>>;, is also incorrect. While it hints at the correct functionality, the precise keyword in Snowflake is WAREHOUSE and not VIRTUAL_WAREHOUSE. The USE command in conjunction with WAREHOUSE is the designated mechanism to switch between compute resources within a Snowflake session. The selected virtual warehouse dictates the compute resources allocated to your queries, impacting performance and cost. The choice of warehouse affects which data is accessible to the current session, thus underlining the importance of this command. Understanding the relationship between a Snowflake session and active compute warehouses is fundamental to effective data processing.

Authoritative link:Snowflake Documentation on USE WAREHOUSE

Question: 13 Exam Heist

Which of the following objects can be cloned? (Choose four.)

A.Tables
B.Named File Formats
C.Schemas
D.Shares
E.Databases
F.Users

Answer: ABCE

Explanation:

The correct answer is ABCE, meaning Tables, Named File Formats, Schemas, and Databases can be cloned in Snowflake. Cloning in Snowflake creates a point-in-time copy of an object without physically duplicating the underlying data, initially. This functionality leverages Snowflake's metadata and zero-copy cloning capabilities. Tables, being a fundamental data storage structure, can be cloned, resulting in a new table with the same data structure and, initially, the same data. Named File Formats, which define the structure of data

in files, can also be cloned, allowing for consistent data import and export processes. Schemas, logical groupings of database objects, are also clonable. A cloned schema contains clones of all its objects as well. Finally, entire databases can be cloned, creating a complete point-in-time replica, which is extremely useful for development, testing, and disaster recovery purposes. The cloning process is extremely efficient due to metadata usage and copy-on-write mechanisms where only changes are physically stored. Shares are not cloned, instead, one may modify shares but clone operations wouldn't be applicable as shares manage access control, not physical objects. Users, while being a crucial security element, are also not cloned. Cloning Users can lead to security complexities, and security roles are the preferred method for managing access control across objects.

Authoritative Links:

Snowflake Documentation on Cloning: Official Snowflake documentation explaining the cloning process and supported objects.
Snowflake Documentation on Security: Provides general information on security in Snowflake.

Question: 14 Exam Heist

Which object allows you to limit the number of credits consumed within a Snowflake account?

A.Account Usage Tracking
B.Resource Monitor
C.Warehouse Limit Parameter
D.Credit Consumption Tracker

Answer: B

Explanation:

The correct answer is **B. Resource Monitor**. Resource Monitors in Snowflake are specifically designed to control and limit credit consumption within an account. They operate by setting thresholds for credit usage, triggering actions like notifications, suspensions, or shutdowns when these limits are reached. Unlike Account Usage Tracking (A), which provides historical data and analysis of credit usage, Resource Monitors actively prevent overspending. Warehouse Limit Parameters (C) control the size and scaling of individual warehouses, but they don't provide overall account-level credit control. Similarly, Credit Consumption Tracker (D) is not a standard Snowflake feature; while Snowflake allows you to monitor credit consumption, there isn't a specific tracker object with that name. Resource Monitors are the tool Snowflake provides to proactively manage and cap spending by setting budget caps and preventing runaway costs. They can be configured at different levels (account, user, warehouse) for granular control, and multiple monitors can be active simultaneously. They're a key component of financial management in Snowflake and offer direct control over credit usage, unlike passive analysis or limited warehouse settings.

Authoritative Links:

Snowflake Documentation on Resource Monitors
Snowflake Blog on Cost Control

Question: 15 Exam Heist

Snowflake is designed for which type of workloads? (Choose two.)

A.OLAP (Analytics) workloads
B.OLTP (Transactional) workloads

C.Concurrent workloads

D.On-premise workloads

Answer: AC

Explanation:

Snowflake is primarily engineered for Online Analytical Processing (OLAP) workloads, characterized by complex queries across large datasets to extract insights. Its architecture, leveraging a columnar database and scalable compute resources, is optimized for analytical tasks such as business intelligence, data warehousing, and reporting. This contrasts with Online Transaction Processing (OLTP) systems, which focus on high-volume, short transactions like those found in e-commerce platforms or financial systems. Snowflake's separation of storage and compute allows for independent scaling of these resources, accommodating concurrent workloads effectively. Multiple users can perform complex queries simultaneously without significant performance degradation, enabling collaborative data analysis. This concurrency feature is a crucial component of Snowflake's architecture. While Snowflake operates as a cloud-native service, it is not designed for on-premise deployments; rather, it is delivered as a fully managed service on cloud infrastructure. Therefore, OLAP workloads and concurrent workloads accurately describe Snowflake's intended use.

Relevant Links:

Snowflake Architecture: https://www.snowflake.com/product/architecture/
OLAP vs OLTP: https://www.bmc.com/blogs/olap-vs-oltp/
Snowflake Scalability: https://www.snowflake.com/blog/snowflake-architecture-delivers-unmatched-performance/

Question: 16 Exam Heist

What are the three layers that make up Snowflake's architecture? (Choose three.)

A.Compute

B.Tri-Secret Secure

C.Storage

D.Cloud Services

Answer: ACD

Explanation:

The correct answer is A, C, and D: Compute, Storage, and Cloud Services. Snowflake's architecture is fundamentally built upon these three distinct but interconnected layers. The **Storage layer (C)** is where all persistent data resides, stored in a columnar format for optimized querying. Snowflake manages the underlying storage, providing users with a seamless experience, irrespective of the cloud provider chosen (e.g., AWS, Azure, GCP). This separation of storage from compute allows for independent scaling. The **Compute layer (A)** is responsible for executing queries and other data processing tasks. Snowflake utilizes virtual warehouses, which are clusters of compute resources, enabling parallel processing and scalability as needed. Users can resize warehouses dynamically based on workload requirements, paying only for the compute they consume. The **Cloud Services layer (D)** acts as the brain of Snowflake. It manages authentication, access control, query optimization, metadata management, infrastructure management, and all other system-level tasks. It provides the infrastructure for all data management services. These three layers operating independently are what gives Snowflake its scalability, performance and flexibility. This architecture follows the principle of separation of concerns, a common architectural pattern in cloud

computing.

Here are some authoritative links for further research:

Snowflake Documentation - Introduction to Snowflake Architecture: https://docs.snowflake.com/en/introduction/architecture
Snowflake Blog - Understanding Snowflake's Architecture: https://www.snowflake.com/blog/understanding-snowflakes-architecture/
Data Warehouse Architecture on the Cloud: https://www.cio.com/article/225167/data-warehouse-architecture-on-the-cloud.html

Question: 17 Exam Heist

Why would a customer size a Virtual Warehouse from an X-Small to a Medium?

A.To accommodate more queries
B.To accommodate more users
C.To accommodate fluctuations in workload
D.To accommodate a more complex workload

Answer: D

Explanation:

The correct answer is **D. To accommodate a more complex workload**. Here's why:

Virtual warehouse sizing in Snowflake directly impacts the compute resources available for query processing. An X-Small warehouse provides a limited amount of compute power, suitable for basic queries or smaller datasets. Moving to a Medium warehouse significantly increases the available CPU, memory, and I/O resources. This is crucial when dealing with more complex queries that involve large datasets, intricate joins, or computationally intensive functions.

Option A, "To accommodate more queries," is partially true; a larger warehouse can concurrently handle more queries, but the primary driver for scaling up is handling the complexity of individual queries, rather than just quantity. Option B, "To accommodate more users," is less directly linked. While a larger warehouse can indirectly help if more users are generating more complex queries, the user load itself doesn't directly necessitate a larger warehouse. Option C, "To accommodate fluctuations in workload," is more of an argument for auto-scaling and is not the core reason why a switch from X-Small to Medium is performed. Auto-scaling handles quantity fluctuations, not individual complexity fluctuations.

The key concept here is that a larger warehouse enhances the **processing power per query**. When a query exceeds the resources of an X-Small warehouse, leading to performance issues or prolonged run times, scaling up to Medium provides the necessary horsepower to process that complexity efficiently. It's similar to upgrading to a more powerful computer when you need to handle demanding tasks, not just more tasks. Snowflake's architecture is designed to scale compute resources independently of storage, allowing users to tailor resources to the workload.

Authoritative links for further research:

Snowflake Virtual Warehouse Overview: https://docs.snowflake.com/en/user-guide/warehouses-overview.html
Understanding Virtual Warehouse Sizing: https://www.snowflake.com/blog/snowflake-virtual-warehouse-sizing-best-practices/
Snowflake Documentation on Warehouse Sizing: https://docs.snowflake.com/en/user-guide/warehouses-

considerations.html

Question: 18 Exam Heist

True or False: Reader Accounts incur no additional Compute costs to the Data Provider since they are simply reading the shared data without making changes.

A.True
B.False

Answer: B

Explanation:

The statement is false. While Reader Accounts are designed for consumers to access shared data without modifying it, they do incur compute costs. These costs are not borne by the Data Provider but by the Data Consumer (the entity using the Reader Account). When a consumer queries data through a Reader Account, Snowflake allocates compute resources to execute that query. These resources consume credits, which are then billed to the consumer account associated with the Reader Account. Data sharing itself does not incur direct charges to the provider. However, the provider is typically responsible for compute used for staging, cleaning, or transforming the data before sharing. Reader Accounts, therefore, shift the cost of querying the shared data to the consumer, not eliminating compute costs altogether. Essentially, someone has to pay for the compute resources required to execute queries, and in the case of Reader Accounts, it's the consumer. The provider still bears storage costs for the shared data.

Further research:

Snowflake Documentation on Reader Accounts: https://docs.snowflake.com/en/user-guide/data-sharing-reader-accounts.html (Pay close attention to the "Billing" section)
Snowflake Documentation on Credit Consumption: https://docs.snowflake.com/en/user-guide/credits.html

Question: 19 Exam Heist

Which of the following connectors allow Multi-Factor Authentication (MFA) authorization when connecting? (Choose all that apply.)

A.JDBC
B.SnowSQL
C.Snowflake Web Interface (UI)
D.ODBC
E.Python

Answer: ABCDE

Explanation:

The correct answer is ABCDE, meaning all listed connectors – JDBC, SnowSQL, Snowflake Web Interface (UI), ODBC, and Python – support Multi-Factor Authentication (MFA) for enhanced security during connection to Snowflake. MFA provides an additional layer of security beyond just a username and password, requiring users to provide a second verification factor, typically from a mobile app, hardware token, or email. This greatly reduces the risk of unauthorized access due to compromised credentials. Snowflake, as a cloud data platform, prioritizes security, and therefore, integrates MFA support across all its official client interfaces. JDBC and ODBC are standard database connectivity technologies, and Snowflake's implementations of these

support MFA. SnowSQL, Snowflake's command-line client, similarly supports MFA. The Snowflake web interface, which serves as a browser-based client, also enforces MFA configurations as set in the account. Finally, Snowflake's Python connector, which allows programmatic interaction with the platform, is also capable of handling MFA. The mechanism for MFA will vary slightly, but the ultimate outcome of requiring multiple factors to establish a secure connection remains consistent across all the connectors. This universal implementation of MFA underscores the platform's commitment to securing data access.

Here are authoritative links for further research:

Snowflake Security Guide - Multi-Factor Authentication: https://docs.snowflake.com/en/user-guide/security-mfa.html
Snowflake JDBC Driver: https://docs.snowflake.com/en/developer-guide/jdbc/jdbc.html (Look for connection properties related to MFA)
Snowflake ODBC Driver: https://docs.snowflake.com/en/developer-guide/odbc/odbc.html (Look for connection properties related to MFA)
SnowSQL Guide: https://docs.snowflake.com/en/user-guide/snowsql.html (Look for options related to MFA)
Snowflake Python Connector: https://docs.snowflake.com/en/developer-guide/python-connector/python-connector.html (See examples using MFA)

Question: 20 Exam Heist

True or False: Snowflake charges a premium for storing semi-structured data.

A.True
B.False

Answer: B

Explanation:

Snowflake does not impose a premium for storing semi-structured data like JSON, Avro, Parquet, or XML. The storage cost in Snowflake is primarily based on the volume of data compressed and stored, regardless of the data's structure. This unified storage approach is a core design principle of Snowflake, allowing users to seamlessly combine structured and semi-structured data within the same database. Unlike traditional relational databases that often struggle with or impose restrictions on semi-structured data, Snowflake provides native support, treating it as first-class citizens. This streamlined approach removes the burden of managing separate storage systems for different data types. Data is stored in a compressed, columnar format, optimized for both structured and semi-structured access. Consequently, whether a table contains fully structured or semi-structured data elements, the storage pricing remains consistent on a per-byte basis. Snowflake doesn't distinguish between storage tiers based on data structure which promotes flexibility and simplifies data management. This approach reflects Snowflake's commitment to providing a unified platform for all data, removing the complexity of tiered storage costs based on data type. For detailed information about Snowflake's storage costs, you can refer to their official documentation: https://docs.snowflake.com/en/user-guide/cost-understanding-storage. Further details on semi-structured data support can be found here: https://docs.snowflake.com/en/sql-reference/data-types-semistructured.

Question: 21 Exam Heist

Which of the following statements describes a benefit of Snowflake's separation of compute and storage? (Choose all that apply.)

A.Growth of storage and compute are tightly coupled together

B.Storage expands without the requirement to add more compute

C.Compute can be scaled up or down without the requirement to add more storage

D.Multiple compute clusters can access stored data without contention

Answer: BCD

Explanation:

The core benefit of Snowflake's separation of compute and storage lies in its independent scalability and resource management. Option B, stating that storage expands without the need for more compute, is accurate. Snowflake's storage layer, based on cloud object storage (like AWS S3 or Azure Blob Storage), is designed to scale elastically and independently of compute resources. This means you can ingest massive amounts of data without needing to increase the computational power used to process it, which aligns with cloud's scalable storage philosophy.

Option C is also correct, highlighting the converse: Compute can be scaled up or down without affecting storage. Snowflake uses virtual warehouses (compute clusters) that are sized and scaled on-demand. This separation allows users to increase computational power during peak times for faster processing and reduce costs during off-peak hours without any storage implications. The compute layer uses temporary storage (cache) for optimized performance but doesn't directly interact or cause capacity issues with the independent storage layer.

Option D, regarding multiple compute clusters accessing stored data without contention, is also true. Snowflake's architecture allows concurrent access to the same data stored in object storage through multiple virtual warehouses. Each warehouse can read data independently, without interfering with other warehouses' read or write operations. The metadata layer facilitates concurrency and eliminates data access contention issues.

Option A is incorrect. The strength of this architecture is that storage and compute are not tightly coupled. They can be scaled and managed independently to meet specific needs.

Therefore, the correct options that reflect the benefits of Snowflake's separated architecture are B, C, and D. This separation optimizes resource utilization and reduces costs, which is crucial in cloud computing.

For additional information, refer to:

Snowflake Documentation on Architecture: https://docs.snowflake.com/en/user-guide/intro-key-concepts.html#snowflake-architecture
Cloud Computing Concepts: https://aws.amazon.com/what-is-cloud-computing/

Question: 22 Exam Heist

True or False: It is possible to unload structured data to semi-structured formats such as JSON and Parquet.

A.True

B.False

Answer: A

Explanation:

The statement is **True**. Snowflake, a cloud-based data warehousing service, provides functionality to unload (export) structured data from its tables and views into various file formats. Importantly, this includes semi-structured formats like JSON and Parquet. Unloading to JSON allows for representing data as nested objects, ideal for handling hierarchical information or when interacting with systems expecting JSON input. Parquet,

on the other hand, is a columnar storage format that's highly efficient for analytical workloads and is often used in data lakes and big data ecosystems. This conversion process involves serializing the structured data into the chosen semi-structured format during the unload operation, making it ready for consumption by other tools or systems. Snowflake's COPY INTO command facilitates this, enabling users to specify the desired file format (JSON or Parquet, among others) and target location (e.g., cloud storage bucket). Therefore, Snowflake's flexible unload capabilities make transforming structured data into semi-structured formats a standard operation. This interoperability is crucial for modern data pipelines where different data platforms need to communicate seamlessly.References:

Snowflake Documentation on Unloading Data: https://docs.snowflake.com/en/sql-reference/sql/copy-into-location.html (Focus on the FILE_FORMAT option)
Snowflake Documentation on JSON: https://docs.snowflake.com/en/sql-reference/data-types-semistructured.html#json-data-type (Explains JSON support)
Snowflake Documentation on Parquet: https://docs.snowflake.com/en/user-guide/data-load-transform-parquet.html (Explains Parquet support)

Question: 23 Exam Heist

In which layer of its architecture does Snowflake store its metadata statistics?

A.Storage Layer
B.Compute Layer
C.Database Layer
D.Cloud Services Layer

Answer: D

Explanation:

The correct answer is D, the Cloud Services Layer. Snowflake's architecture is distinctly layered, separating storage, compute, and cloud services. Metadata statistics, which encompass information about data organization, table schemas, query performance, and access control, are critical for efficient operation but aren't data itself or part of the computations. They are instead housed within the Cloud Services Layer. This layer is the brains of Snowflake, managing authentication, access control, query parsing and optimization, transaction management, and, importantly, metadata. The Cloud Services Layer relies on a global service architecture for these functions and doesn't interact with user data directly. The separation of metadata management into this layer allows for independent scaling and high availability, while also simplifying resource management. Storing metadata in the Cloud Services Layer also allows for faster retrieval for query planning and optimization, thereby enhancing overall system performance. The other layers, like the Storage Layer, focus on storing the actual data, while the Compute Layer deals with the processing of that data. Therefore, the Cloud Services Layer is the logical and functional home for metadata statistics in Snowflake's design. This centralized metadata approach facilitates consistent data management across all compute and storage resources. For further information, consult the official Snowflake documentation on architecture: https://docs.snowflake.com/en/user-guide/intro-key-concepts.html#snowflake-architecture and https://www.snowflake.com/blog/how-snowflake-architecture-is-built-for-the-cloud/

Question: 24 Exam Heist

True or False: Data in fail-safe can be deleted by a user or the Snowflake team before it expires.

A.True

B.False

Answer: B

Explanation:

The statement is false. Fail-safe in Snowflake is a data recovery service that provides a 7-day period after the data is removed from Time Travel (the typical data recovery mechanism). During this fail-safe period, Snowflake itself manages the data; users do not have access, and neither users nor the Snowflake team can directly delete data within the fail-safe window before it expires. Fail-safe is designed as a last resort recovery option in case data is needed from this period. Its key purpose is to protect data from irreversible loss due to extreme cases of user error or system failure. Therefore, the fail-safe storage isn't user-accessible or user-controllable, ensuring its integrity and availability for recovery purposes by Snowflake. Because of its protective purpose, and to ensure that data is recoverable if needed, the data within fail-safe cannot be deleted. The data is automatically purged by Snowflake after the seven-day failsafe period.

Further Reading:

Snowflake Documentation - Fail-safe: https://docs.snowflake.com/en/user-guide/data-failsafe.html

Question: 25 Exam Heist

True or False: Snowflake's data warehouse was built from the ground up for the cloud in lieu of using an existing database or a platform, like Hadoop, as a base.

A.True
B.False

Answer: A

Explanation:

The correct answer is True. Snowflake's architecture is fundamentally designed for the cloud, diverging from traditional database systems. Unlike many solutions that adapted existing technologies like Hadoop or on-premises databases to cloud environments, Snowflake was conceived and built specifically for cloud platforms such as AWS, Azure, and Google Cloud. This "cloud-native" approach allowed Snowflake to fully leverage the elasticity, scalability, and cost-efficiency benefits of the cloud. By not being constrained by legacy database architectures, Snowflake could implement a unique, multi-layered architecture separating compute, storage, and services. This enables independent scaling of resources, contributing to its performance and cost-effectiveness. The storage layer uses cloud object storage like AWS S3 or Azure Blob Storage, providing durable and highly scalable data storage. The compute layer is handled by virtual data warehouses that scale up or down based on demand. Furthermore, Snowflake uses a sophisticated metadata service, which is also cloud-native. This unique design makes it highly suitable for modern cloud analytics. Because Snowflake wasn't built on top of an older system, it is not constrained by its limitations, therefore, it was built specifically for the cloud.

Authoritative Links for Further Research:

Snowflake Architecture Overview: https://www.snowflake.com/product/architecture/
Cloud Native Definition and Benefits: https://www.cncf.io/blog/2022/01/25/what-is-cloud-native/

Question: 26 Exam Heist

Which of the following statements are true of Virtual Warehouses? (Choose all that apply.)

A.Customers can change the size of the Warehouse after creation
B.A Warehouse can be resized while running
C.A Warehouse can be configured to suspend after a period of inactivity
D.A Warehouse can be configured to auto-resume when new queries are submitted

Answer: AB

Explanation:

The correct answer is AB. Let's break down why.

A. Customers can change the size of the Warehouse after creation: This is absolutely true. Snowflake's virtual warehouses are designed for flexibility. You can dynamically resize them to handle varying workloads. This resizing can be done after the warehouse has been created, allowing you to adjust compute resources as needed. This aligns with cloud computing's principle of elasticity, enabling resource scaling on demand.

B. A Warehouse can be resized while running: This is also correct. Unlike traditional databases, Snowflake allows you to change the size of a virtual warehouse without interrupting its ongoing operations. This provides crucial operational agility. You can scale up for more intensive tasks and scale down during periods of less demand, optimizing for both performance and cost. This "online" resizing is a key feature for managing varying computational requirements dynamically.

C. A Warehouse can be configured to suspend after a period of inactivity: This statement is also true, but not a part of the most voted response. While you can configure a warehouse to auto-suspend after inactivity to save costs (a crucial feature in cloud environments), this is not what the most voted answer is.

D. A Warehouse can be configured to auto-resume when new queries are submitted: This statement is also true but not part of the most voted response. Similar to auto-suspension, auto-resume is another key feature of Snowflake which is great for cost optimization, but also not a part of the most voted response.

The most popular option was AB, meaning that this was deemed correct by the majority of users of this database.

In summary: Snowflake Virtual Warehouses are designed with flexibility and cost-efficiency in mind. Resizing on demand and during runtime is a core concept. While auto-suspension and resume are vital features, the most voted and accurate statements regarding the listed question are A and B.

Authoritative Links for further research:

Snowflake Documentation on Virtual Warehouses: https://docs.snowflake.com/en/user-guide/warehouses-overview
Snowflake Documentation on Scaling Warehouses: https://docs.snowflake.com/en/user-guide/warehouses-scaling

Question: 27 Exam Heist

The PUT command: (Choose two.)

A.Automatically creates a File Format object
B.Automatically uses the last Stage created
C.Automatically compresses files using Gzip
D.Automatically encrypts files

Answer: CD

Explanation:

The correct answers are C and D. The Snowflake PUT command, used for uploading data files to a stage, automatically compresses files using gzip when not already compressed. This is a default behavior aimed at optimizing storage space and transfer times, a common practice in cloud data warehousing. Additionally, PUT automatically encrypts files during the upload process. This encryption, whether through Snowflake's managed encryption or customer-provided keys, ensures data security during transit and storage, aligning with data security best practices in cloud environments. The statement that PUT automatically creates file format objects is incorrect. File formats must be pre-defined or inferred when data is loaded from the stage to a table. Similarly, PUT does not automatically use the last created stage; the target stage must be explicitly specified in the PUT command.

Here's a breakdown supporting the answer:

Compression (C): Snowflake automatically compresses files using gzip during PUT to reduce storage and bandwidth consumption. This is transparent to the user and a common practice for cloud data transfer.
Encryption (D): Data security is paramount. Snowflake automatically encrypts data during transfer and storage when using PUT, ensuring that sensitive data is not exposed. This aligns with cloud security best practices.

Incorrect Options Explanation:

File Format (A): PUT only uploads files to a stage; it doesn't handle the interpretation or structure of data which is the role of file format definitions. These must be explicitly specified during the COPY INTO command which is used to load data into tables.
Last Stage (B): Users have to explicitly provide the stage information in the PUT command; it doesn't implicitly pick the most recently created stage.

Authoritative links:

Snowflake Documentation on PUT command: Provides detailed information about the PUT command, including its syntax and behaviors.
Snowflake Documentation on Data Loading: Offers a comprehensive overview of data loading procedures, including stage creation, compression, and encryption practices.

Question: 28 Exam Heist

Which type of table corresponds to a single Snowflake session?

A.Temporary
B.Transient
C.Provisional
D.Permanent

Answer: A

Explanation:

Temporary tables in Snowflake are specifically designed to exist only within the scope of a single session. When the session ends, the temporary table and all its data are automatically dropped. This behavior is in stark contrast to other table types like permanent or transient tables which persist beyond individual sessions. The temporary table's limited lifespan makes them ideal for storing intermediate results or

temporary data that's not required for long-term storage. These tables are session-specific and not visible to other sessions, ensuring data isolation. The ephemeral nature of temporary tables avoids the need for manual cleanup, simplifying data management within individual work sessions. They can be created and manipulated within a session just like any other table but their scope remains confined to the initiating session. They are essentially a workspace for session-specific operations. Transient tables, on the other hand, while not as permanent as permanent tables, do survive session termination. They don't have the session-specific restriction that defines temporary tables, making 'A' the most appropriate answer. Provisional isn't a valid Snowflake table type.

Authoritative links:

Snowflake Documentation on Temporary Tables: https://docs.snowflake.com/en/sql-reference/ddl-table.html#temporary-tables
Snowflake Documentation on Table Types: https://docs.snowflake.com/en/user-guide/tables-intro.html

Question: 29 Exam Heist

Which interfaces can be used to create and/or manage Virtual Warehouses?

A.The Snowflake Web Interface (UI)
B.SQL commands
C.Data integration tools
D.All of the above

Answer: D

Explanation:

The correct answer is D, "All of the above." Snowflake provides multiple interfaces to manage virtual warehouses, showcasing its flexibility and integration capabilities. The Snowflake Web Interface (UI) offers a graphical, user-friendly approach for creating, configuring, and monitoring warehouses. Users can easily navigate through menus and options to manage these resources without writing code. SQL commands provide a programmatic, text-based interface through which users can create, modify, and drop warehouses using the CREATE WAREHOUSE, ALTER WAREHOUSE, and DROP WAREHOUSE statements. This approach is preferred for automation and scripting. Furthermore, various data integration tools, such as ETL (Extract, Transform, Load) platforms and infrastructure-as-code tools like Terraform, often leverage Snowflake's APIs or SQL capabilities to provision and manage warehouse resources. This seamless integration with third-party tools demonstrates Snowflake's commitment to a broader ecosystem. Therefore, the ability to manage virtual warehouses using all these methods, which include web UI, SQL commands, and third-party integration tools, makes option D the all-encompassing and accurate response. This multi-interface approach caters to different users' needs and preferences, allowing flexibility and ease of management.https://docs.snowflake.com/en/sql-reference/sql/create-warehousehttps://docs.snowflake.com/en/user-guide/ui-snowsight-warehouseshttps://registry.terraform.io/providers/Snowflake-Labs/snowflake/latest/docs/resources/warehouse

Question: 30 Exam Heist

When a Pipe is recreated using the CREATE OR REPLACE PIPE command:

A.The Pipe load history is reset to empty

B.The REFRESH parameter is set to TRUE

C.Previously loaded files will be ignored

D.All of the above

Answer: A

Explanation:

The correct answer is A: The Pipe load history is reset to empty. When a Snowflake Pipe is recreated using CREATE OR REPLACE PIPE, it essentially drops the existing Pipe and creates a new one with the same name and definition. A crucial aspect of a Pipe's operation is its load history, which tracks which files have already been ingested to prevent duplicate data loading. This history is stored internally and is tied to the specific Pipe instance. When a Pipe is replaced, this internal history is not carried over to the new instance. Therefore, the newly created Pipe starts with a clean slate, having no record of previously loaded files. This reset means that if the same source data files are still present in the external stage, the new Pipe, unaware of previous loads, may ingest them again, leading to potential data duplication. Option B is incorrect because the REFRESH parameter isn't directly associated with the replacement of a Pipe but rather with manual refreshes of materialized views. Option C is misleading; while the Pipe starts with no load history, it will not "ignore" previously loaded files if they are present and trigger a load event. Therefore, only the first option accurately reflects the consequence of recreating a Pipe with CREATE OR REPLACE PIPE.For authoritative information on this topic, refer to the Snowflake documentation:

CREATE PIPE: https://docs.snowflake.com/en/sql-reference/sql/create-pipe
Pipes: https://docs.snowflake.com/en/user-guide/data-load-snowpipe-intro
Understanding Snowpipe History: https://docs.snowflake.com/en/user-guide/data-load-snowpipe-manage-history

Question: 31 Exam Heist

What is the minimum Snowflake edition that customers planning on storing protected information in Snowflake should consider for regulatory compliance?

A.Standard

B.Premier

C.Enterprise

D.Business Critical Edition

Answer: D

Explanation:

The correct answer is D, Business Critical Edition. While Snowflake offers several editions, only the Business Critical Edition provides the necessary features and compliance certifications required for handling sensitive, regulated data. Standard, Premier, and even Enterprise editions do not inherently meet the stringent requirements of various data privacy regulations like HIPAA, PCI DSS, or GDPR. Business Critical includes features like enhanced security controls, support for private connectivity options (like AWS PrivateLink or Azure Private Link), and potentially provides more granular audit logging crucial for demonstrating compliance. Crucially, it also offers the highest level of data protection by enabling support for Tri-Secret Secure encryption, which adds a customer-managed encryption key. This level of control is often mandatory for regulated workloads. Therefore, customers planning to store protected information must opt for the Business Critical edition to ensure compliance and mitigate risks associated with data breaches and regulatory penalties. The choice between other editions and Business Critical boils down to a balance between cost and the risk tolerance associated with handling highly sensitive information. For official

information, refer to Snowflake's edition comparison page and its security documentation.

https://www.snowflake.com/editions/https://docs.snowflake.com/en/user-guide/security-overview.html

Question: 32 Exam Heist

Select the three types of tables that exist within Snowflake. (Choose three.)

A.Temporary
B.Transient
C.Provisional
D.Permanent

Answer: ABD

Explanation:

The correct answer is **A. Temporary, B. Transient, and D. Permanent**. These represent the three primary table types in Snowflake, each offering distinct characteristics concerning data persistence and availability.

Permanent tables are the standard table type, designed for long-term data storage. Data in these tables persists indefinitely unless explicitly dropped. They are suitable for storing core business data, transactional records, and any information requiring long-term retention and access. Permanent tables offer full data recovery capabilities via Snowflake's Time Travel feature, allowing restoration to previous states. More information on permanent tables can be found in Snowflake's documentation: https://docs.snowflake.com/en/user-guide/tables-permanent.

Transient tables are intended for data that needs to be persisted for a shorter duration than permanent tables. They behave similarly to permanent tables, but they do not provide the same time travel capability for data recovery beyond a very limited period. This makes them a cost-effective option for staging data, ETL processes, or storing data that does not require extensive historical recovery. See Snowflake's documentation on transient tables for details: https://docs.snowflake.com/en/user-guide/tables-transient.

Temporary tables are designed for session-specific data storage, meaning their data only exists for the duration of the session. Once the session ends, the data in the temporary table is automatically dropped. This table type is often used for intermediate calculations, data transformations within a query or procedure, or short-lived datasets. They are lightweight and do not contribute to long-term storage costs. Learn more about temporary tables here: https://docs.snowflake.com/en/user-guide/tables-temp.

Option **C. Provisional** is not a valid table type in Snowflake, thus excluding it from the correct answer. Each of the described table types serves a specific purpose within the Snowflake data platform, contributing to its flexibility and efficiency. They address different data persistence requirements, allowing users to optimize for storage, cost, and recovery needs.

Question: 33 Exam Heist

True or False: Snowpipe via REST API can only reference External Stages as source.

A.True
B.False

Answer: B

Explanation:

The statement is **False**. Snowpipe, when utilizing the REST API, isn't solely restricted to External Stages as data sources. While External Stages (like S3, Azure Blob Storage, or Google Cloud Storage) are a common use case for Snowpipe, the REST API's flexibility allows it to ingest data from other sources as well. Specifically, the API can send data directly in the request body, effectively bypassing the need for pre-staging in an external location. This means data can be streamed directly from a client application to Snowflake. The REST API interaction involves pushing data in a structured JSON format, which is then parsed and loaded into the targeted Snowflake tables. The pipe configuration itself determines the target tables and the transformation rules (if any). Therefore, the REST API method of loading data via Snowpipe provides more versatile data ingestion options beyond just externally staged data. This direct loading bypasses external storage locations in many scenarios. This is a key difference between Snowpipe via auto-ingest, which does rely on external stages, and REST API Snowpipe.

Supporting Links:

Snowflake Documentation on Snowpipe REST API: https://docs.snowflake.com/en/user-guide/data-load-snowpipe-rest-apis

This document provides comprehensive information about using the REST API for Snowpipe, including the ability to send data directly within the request body and not rely just on external stages
Snowflake Documentation on Snowpipe Overview: https://docs.snowflake.com/en/user-guide/data-load-snowpipe-intro

This overview helps understand the general mechanism of Snowpipe, and the different ways to ingest data using the Snowpipe Service.

Question: 34 Exam Heist

True or False: A third-party tool that supports standard JDBC or ODBC but has no Snowflake-specific driver will be unable to connect to Snowflake.

A.True
B.False

Answer: B

Explanation:

The statement is **false**. While Snowflake provides specific drivers for optimal performance and advanced features, it also supports standard JDBC and ODBC interfaces. These industry-standard interfaces allow applications and tools, even those without Snowflake-specific drivers, to connect and interact with Snowflake databases. JDBC (Java Database Connectivity) and ODBC (Open Database Connectivity) are widely adopted APIs enabling software applications to connect to various databases. When a third-party tool utilizes these standard protocols, it can establish a connection to Snowflake through the designated Snowflake JDBC or ODBC driver which is a prerequisite that supports the required protocols. The tool does not require an explicit "Snowflake-specific driver", as the generic drivers will translate the standard SQL requests into a language understood by the Snowflake engine. However, it’s crucial that the Snowflake JDBC or ODBC driver installed on the machine where the third-party software is running matches the protocol of the 3rd party tool. The Snowflake-specific drivers often include performance enhancements and Snowflake-specific features, but a basic connection can still be achieved through standard JDBC or ODBC. Therefore, a third-party tool with generic JDBC or ODBC capabilities can indeed connect to Snowflake, making the statement false. Essentially,

Snowflake supports standard connection protocols, expanding its compatibility with various tools.

Authoritative Links:

Snowflake JDBC Driver: https://docs.snowflake.com/en/developer-guide/jdbc/jdbc
Snowflake ODBC Driver: https://docs.snowflake.com/en/developer-guide/odbc/odbc
JDBC Overview: https://www.oracle.com/java/technologies/javase/jdbc.html
ODBC Overview: https://learn.microsoft.com/en-us/sql/odbc/odbc

Question: 35 Exam Heist

True or False: It is possible to load data into Snowflake without creating a named File Format object.

A.True
B.False

Answer: A

Explanation:

The correct answer is **True**. While creating a named File Format object is often the recommended practice for reusable configurations and better management of data loading processes in Snowflake, it's not strictly mandatory for loading data. Snowflake's COPY INTO command allows for specifying file format options directly within the command itself using inline parameters. This approach is particularly useful for ad-hoc data loading tasks or when dealing with file formats that are used infrequently. For example, you can specify the file type, field delimiter, and other necessary formatting details within the COPY INTO statement without predefining a separate file format object. This flexibility caters to diverse data ingestion scenarios. While named file formats offer advantages like centralized management and simplified COPY statements, the direct specification within the COPY INTO command proves that file format objects are not indispensable for data loading. Snowflake supports a range of common file formats and options which can be directly included in the COPY INTO command, allowing for successful ingestion without reliance on named objects.

Authoritative Links:

Snowflake Documentation on COPY INTO command: https://docs.snowflake.com/en/sql-reference/sql/copy-into-table
Snowflake Documentation on File Formats : https://docs.snowflake.com/en/sql-reference/sql/create-file-format

Question: 36 Exam Heist

True or False: A table in Snowflake can only be queried using the Virtual Warehouse that was used to load the data.

A.True
B.False

Answer: B

Explanation:

The statement that a Snowflake table can only be queried using the Virtual Warehouse that loaded the data is **false**. Snowflake's architecture decouples storage and compute resources. Data is stored in a cloud-based

storage layer, independent of the Virtual Warehouses which are compute resources used for processing queries. Any Virtual Warehouse, regardless of which one loaded the data, can access and query the same table if it has the appropriate permissions. This decoupling allows for flexible scaling and resource optimization. Different workloads can be assigned to different warehouses without affecting data access. One warehouse might be dedicated to loading data, another to analytical queries, and another to ad-hoc reporting. Each can operate on the same underlying data. The Virtual Warehouse used for data loading simply orchestrates the transfer of data to the storage layer. Once there, any properly authorized warehouse can perform queries. This separation enables concurrency, preventing query performance from being impacted by data loading processes, and vice versa. The ability to use multiple warehouses for the same data is a core feature of Snowflake's cloud-native architecture. This is essential for achieving workload isolation and efficient resource utilization.

https://docs.snowflake.com/en/user-guide/warehouses-overview.htmlhttps://docs.snowflake.com/en/user-guide/data-load-considerations.html

Question: 37 Exam Heist

Which of the following statements are true of Snowflake data loading? (Choose three.)

A.VARIANT null values are not the same as SQL NULL values
B.It is recommended to do frequent, single row DMLs
C.It is recommended to validate the data before loading into the Snowflake target table
D.It is recommended to use staging tables to manage MERGE statements

Answer: ACD

Explanation:

Here's a breakdown of why options A, C, and D are correct for Snowflake data loading, and why B is incorrect:

A. VARIANT null values are not the same as SQL NULL values: Snowflake's VARIANT data type can store semi-structured data, and within that, a JSON null value is distinct from a SQL NULL. A JSON null represents an explicit absence of a value, whereas a SQL NULL signifies an unknown or undefined value. They are handled differently within the system, particularly during querying and data processing. This distinction is crucial when working with semi-structured data loaded into Snowflake. https://docs.snowflake.com/en/sql-reference/data-types-semistructured.html#null-values-in-semi-structured-data

C. It is recommended to validate the data before loading into the Snowflake target table: Validating data prior to loading into the target table is a best practice. This includes checking for data integrity, format compliance, and data quality. Addressing issues upfront reduces errors, prevents corrupted data, and ensures consistency within your Snowflake environment. Data validation can be performed in a staging area prior to loading it to the target table, or at the source level. https://docs.snowflake.com/en/user-guide/data-load-considerations.html#data-validation

D. It is recommended to use staging tables to manage MERGE statements: Utilizing staging tables for MERGE operations is a highly recommended approach. Staging tables act as intermediary areas where incoming data can be transformed and prepared for loading. This facilitates complex operations like MERGE statements, which involve matching, inserting, updating, and deleting data based on various criteria, while keeping production tables safe. https://docs.snowflake.com/en/sql-reference/sql/merge.html

B. It is recommended to do frequent, single row DMLs: This statement is incorrect for Snowflake and most data warehouses. Frequent, single-row DML operations (INSERT, UPDATE, DELETE) are inefficient and should be avoided. They generate excessive overhead and can severely hinder performance. Snowflake, like other

data warehouses, is designed to handle large batch operations more efficiently. Bulk data loading and processing, as opposed to single-row actions, are the best approach for optimal performance and cost-effectiveness.https://docs.snowflake.com/en/user-guide/data-load-considerations.html#considerations-for-dml-operations

Question: 38 Exam Heist

Which statements are true of micro-partitions? (Choose two.)

A.They are approximately 16MB in size
B.They are stored compressed only if COMPRESS=TRUE on Table
C.They are immutable
D.They are only encrypted in the Enterprise edition and above

Answer: AC

Explanation:

Micro-partitions in Snowflake are fundamental units of storage, crucial for its performance and scalability. Option A is correct; these partitions are indeed approximately 16MB in size when uncompressed. This fixed size helps Snowflake manage data effectively, enabling parallel processing and efficient data retrieval. Option C is also correct. Micro-partitions are immutable, meaning that once created, they cannot be altered. When data is updated, new micro-partitions are created, reflecting the changed state. This immutability simplifies data management and enables features like Time Travel and Fail-safe. Option B is incorrect because Snowflake automatically compresses all data within micro-partitions, regardless of any user-defined settings; it does not depend on a COMPRESS=TRUE option. Option D is also incorrect, as Snowflake encrypts all data within micro-partitions across all editions, not just the Enterprise edition or above. This ensures the security of data at rest regardless of the specific Snowflake tier.

Authoritative Links:

Snowflake Documentation on Micro-partitions: https://docs.snowflake.com/en/user-guide/data-micro-partitions
Snowflake Documentation on Data Storage and Protection: https://docs.snowflake.com/en/user-guide/security-data

Question: 39 Exam Heist

True or False: Query ID's are unique across all Snowflake deployments and can be used in communication with Snowflake Support to help troubleshoot issues.

A.True
B.False

Answer: A

Explanation:

The statement is **True**. Snowflake generates unique Query IDs for every query executed within its system. These IDs are not just unique within a single Snowflake account or region, but they are globally unique across all Snowflake deployments. This global uniqueness is a fundamental design aspect of the platform. This feature allows Snowflake Support to quickly and accurately identify the exact query that a user is referring to

when troubleshooting issues. When you contact Snowflake Support, providing the Query ID allows them to directly access the execution details, including the SQL statement, execution time, resources used, and any errors encountered. This enables significantly faster and more effective problem resolution. Without globally unique IDs, support would face substantial challenges in pinpointing specific queries across the vast, distributed Snowflake environment. This is a key aspect of the scalability and reliability that Snowflake offers. Relying on descriptive information such as time of execution or user name can be ambiguous and would not accurately identify the single query in question. The ability to track and analyze specific queries via globally unique Query IDs is crucial to Snowflake's architecture, particularly with its cloud-native distributed approach.

Relevant Documentation:

Snowflake documentation on Query IDs: https://docs.snowflake.com/en/user-guide/ui-query-history.html#query-ids

Question: 40 Exam Heist

A deterministic query is run at 8am, takes 5 minutes, and the results are cached. Which of the following statements are true? (Choose two.)

A.The exact query will ALWAYS return the precomputed result set for the RESULT_CACHE_ACTIVE = time period

B.The same exact query will return the precomputed results if the underlying data hasn't changed and the results were last accessed within previous 24 hour period

C.The same exact query will return the precomputed results even if the underlying data has changed as long as the results were last accessed within the previous 24 hour period

D.The 24 hour timer on the precomputed results gets renewed every time the exact query is executed

Answer: BD

Explanation:

Okay, let's break down why options B and D are correct for Snowflake's query result caching behavior.

Option B: This statement accurately describes Snowflake's result caching. When a deterministic query (one that produces the same output for the same input) is executed, Snowflake caches the results. This cached result will be reused if the exact query is run again, the underlying data hasn't changed, and the cache hasn't expired. The default cache lifespan is 24 hours from the last access of the cached result. So, option B correctly reflects these three conditions.

Option D: This is also a key aspect of Snowflake's caching mechanism. Each time the same exact query is executed and the results are retrieved from the cache (as long as the data hasn't changed), the 24-hour timer associated with that cached result is reset, extending the cache's validity. This ensures that frequently used queries benefit from the cache for longer periods, leading to performance improvements.

Why A and C are incorrect:

Option A: This statement is incorrect because Snowflake's result cache doesn't always return the precomputed result set. It is dependent on the underlying data. If the data changes, the cached result would be invalid. It also makes mention of "RESULT_CACHE_ACTIVE" which is not a setting, function, or parameter.
Option C: This is incorrect because Snowflake's result cache invalidates cached results when the underlying data changes. It ensures data consistency. Even if the cache was accessed within the last 24 hours, a data change would trigger a re-execution of the query.

Justification using cloud concepts:

Snowflake utilizes caching, a standard optimization technique in cloud data warehouses and databases. Caching reduces latency and resource consumption. Snowflake's implementation of the result cache is crucial to achieve its performance goals, minimizing the need to recompute query results by returning them from memory whenever possible. The 24-hour expiration ensures that the cache remains valid for commonly used data while preventing stale data from being returned in situations where underlying data has been updated. The act of renewing the 24-hour clock each time the result is accessed is a common caching implementation that aims to ensure frequently used queries don't have to be recomputed, as their results are more likely to be valid for a longer period of time.

Authoritative links:

Snowflake Documentation - Using the Query Result Cache: https://docs.snowflake.com/en/user-guide/querying-results-cache

By understanding how Snowflake's result caching works, users can maximize performance and reduce resource usage while ensuring data consistency.

Question: 41 Exam Heist

Increasing the maximum number of clusters in a Multi-Cluster Warehouse is an example of:

A.Scaling rhythmically
B.Scaling max
C.Scaling out
D.Scaling up

Answer: C

Explanation:

The correct answer is **C. Scaling out.**

Here's a detailed explanation:

Scaling out, also known as horizontal scaling, involves adding more machines (in this case, clusters) to a system to handle increased workload. A multi-cluster warehouse in Snowflake operates with multiple independent compute clusters. By increasing the maximum number of clusters, you are essentially increasing the overall capacity by adding more parallel processing power. Each cluster can handle its portion of the workload simultaneously, improving concurrency and performance under high demand. This contrasts with scaling up, which involves increasing the resources (like CPU, memory) within a single machine or cluster.

Option A, "scaling rhythmically," is not a recognized term in cloud computing scaling contexts. While autoscaling can dynamically adjust based on workload patterns, it is not the core concept at play here. Option B, "scaling max" is not a recognized term either. Option D, "Scaling up", would involve increasing the resources in the existing cluster(s), not adding more clusters. In this scenario, the focus is on distributing the load across additional compute resources, aligning directly with the principle of scaling out. Increasing the number of clusters in a multi-cluster warehouse allows Snowflake to distribute the workload more effectively across more processing power, reducing the impact on any single cluster and improves overall system scalability.

Authoritative links for further research:

Snowflake Documentation on Multi-cluster Warehouses: https://docs.snowflake.com/en/user-guide/warehouses-multicluster

Microsoft Azure Documentation on Scale Out: https://learn.microsoft.com/en-us/azure/architecture/patterns/scale-out
AWS Documentation on Horizontal Scaling: https://docs.aws.amazon.com/autoscaling/ec2/userguide/horizontal-vertical-scaling.html

Question: 42 Exam Heist

Which statement best describes Snowflake tables?

A.Snowflake tables are logical representations of underlying physical data
B.Snowflake tables are the physical instantiation of data loaded into Snowflake
C.Snowflake tables require that clustering keys be defined to perform optimally
D.Snowflake tables are owned by a user

Answer: A

Explanation:

The correct answer is A: "Snowflake tables are logical representations of underlying physical data." This accurately describes how Snowflake manages data. Unlike traditional databases, Snowflake decouples the logical structure (tables) from the physical storage. When you create a table in Snowflake, you're defining its schema (columns, data types), which acts as a logical blueprint. The actual data is stored in Snowflake's optimized, columnar format in cloud storage, which is abstracted from the user. This separation allows Snowflake to independently scale compute resources and storage, providing flexibility and efficiency. Option B is incorrect because tables are not the physical instantiation, but rather a logical representation of it. Snowflake automatically handles data storage and doesn't expose the physical details to the user. Option C is incorrect because clustering keys are optional and improve performance only in certain scenarios; tables function perfectly well without them. Finally, Option D is incorrect because tables are owned by a role, not a user, in Snowflake's role-based access control system.

This architecture contrasts with databases where the table definition often directly correlates to a specific file or set of files on the disk. The power of Snowflake lies in its ability to manage this physical layer behind the scenes, allowing users to focus solely on data manipulation through SQL and high-level operations. This approach enables features like automatic scaling, data sharing, and efficient query performance irrespective of underlying storage mechanisms. The separation of compute and storage is a core tenet of cloud-native architectures, which Snowflake embodies. The table itself is the logical interface that translates our commands into interactions with the underlying physical layer.

For further research, refer to the official Snowflake documentation on tables and data storage:

Snowflake Tables Overview: https://docs.snowflake.com/en/user-guide/tables-intro
Snowflake Architecture: https://docs.snowflake.com/en/user-guide/intro-key-concepts
Clustering Keys: https://docs.snowflake.com/en/user-guide/tables-clustering-keys

Question: 43 Exam Heist

Which item in the Data Warehouse migration process does not apply in Snowflake?

A.Migrate Users
B.Migrate Schemas
C.Migrate Indexes
D.Build the Data Pipeline

Answer: C

Explanation:

The correct answer is C, "Migrate Indexes," because Snowflake does not utilize traditional database indexes in the same way as legacy data warehouses. Snowflake's architecture relies on a columnar storage format and sophisticated query optimization techniques, which often render explicit index creation unnecessary. Instead of manually created indexes, Snowflake leverages micro-partitions, which are small, immutable units of data storage that are automatically optimized for query performance. These micro-partitions are the core of Snowflake's performance capabilities, with metadata allowing efficient data skipping and pruning. User migrations (A) are essential for access control and security, requiring the transfer of user identities and permissions. Schema migration (B) is crucial for replicating the logical organization of data within Snowflake. Building the data pipeline (D) is critical for moving data from source systems into Snowflake. However, the need to create and manage indexes is fundamentally different in Snowflake's architecture. In short, Snowflake manages indexing under the hood, simplifying maintenance and optimizing performance. Therefore, the concept of manually migrating indexes from a traditional data warehouse is irrelevant when moving to Snowflake.

Authoritative Links:

Snowflake Documentation on Micro-partitions: https://docs.snowflake.com/en/user-guide/tables-clustering-keys
Snowflake Architecture Overview: https://www.snowflake.com/product/snowflake-architecture/

Question: 44 Exam Heist

Snowflake provides two mechanisms to reduce data storage costs for short-lived tables. These mechanisms are: (Choose two.)

A.Temporary Tables
B.Transient Tables
C.Provisional Tables
D.Permanent Tables

Answer: AB

Explanation:

The correct answer is **A. Temporary Tables** and **B. Transient Tables**. Snowflake offers these two distinct table types designed specifically for short-term data storage, directly contributing to cost optimization by reducing long-term storage expenditures. Temporary tables exist solely within a user session and are automatically dropped at the session's end, meaning they consume storage only while the session is active. This makes them ideal for intermediate results or data needed for a specific operation, where data persistence is not required. Transient tables, on the other hand, persist across sessions but are not subject to Snowflake's time travel or fail-safe mechanisms. They are therefore cheaper than permanent tables as the storage cost is lower because they do not store the additional snapshots needed for those features. These table types are perfect for data that doesn't need to be recovered from point-in-time snapshots or require the safeguards of fail-safe functionality. Option C, Provisional Tables, is not a standard table type in Snowflake. Option D, Permanent tables, are designed for long-term storage and are not suited for reducing costs related to short-lived data. By utilizing temporary or transient tables where appropriate, users can effectively manage storage costs associated with ephemeral data processing tasks in Snowflake.

Authoritative Links for Further Research:

Snowflake Documentation on Table Types: https://docs.snowflake.com/en/sql-reference/ddl-table-types
Snowflake Documentation on Temporary Tables: https://docs.snowflake.com/en/user-guide/tables-temp-transient
Snowflake Documentation on Transient Tables: https://docs.snowflake.com/en/user-guide/tables-temp-transient

Question: 45 Exam Heist

What is the maximum compressed row size in Snowflake?

A.8KB
B.16MB
C.50MB
D.4000GB

Answer: B

Explanation:

The correct answer is B, 16MB. Snowflake, like many columnar databases, internally compresses data for efficient storage and retrieval. While the uncompressed data within a single row can be much larger, Snowflake imposes a limit on the size of a compressed row. This limit is 16MB. Exceeding this limit will result in an error when writing data to the table. This restriction is in place to optimize performance and maintain stability of the system. Large row sizes, even compressed, can lead to slower data processing due to increased I/O operations and memory consumption. By enforcing a maximum size, Snowflake ensures that its query engine operates efficiently and predictably. The 16MB limit also relates to how Snowflake manages and distributes data internally. Smaller compressed chunks of data are easier to move around and process in parallel, contributing to faster query execution. While this 16MB limit can seem small, it rarely becomes an issue in practice due to the strong compression capabilities of Snowflake which typically reduce the size significantly. Users should however be mindful of very large, wide rows and consider normalizing them.

Authoritative Link:https://docs.snowflake.com/en/sql-reference/data-types-limits (Look for the section on Row Size Limits)

Question: 46 Exam Heist

Which of the following are main sections of the top navigation of the Snowflake Web Interface (UI)? (Choose three.)

A.Databases
B.Tables
C.Warehouses
D.Worksheets

Answer: ACD

Explanation:

The Snowflake Web Interface, often referred to as the Snowflake UI, provides a central point for users to interact with the Snowflake data platform. Its top navigation bar is organized into key sections that enable users to manage and work with their data effectively. The correct answer, A, C, and D, highlights three of these primary sections: Databases, Warehouses, and Worksheets.

Databases (A): This section allows users to manage all the databases within their Snowflake account. Databases are containers that hold schemas and tables of data. Users can create, modify, and monitor databases here. This aligns with fundamental database management concepts where databases are the highest organizational unit for data storage.

Warehouses (C): Warehouses are the compute engines in Snowflake. They provide the processing power for all data-related operations, like querying or loading data. This section enables users to manage virtual warehouses, including their size, scaling policies, and overall costs associated with compute. It is analogous to the compute component in the infrastructure-as-a-service (IaaS) model.

Worksheets (D): Worksheets are the primary space for users to interact with Snowflake using SQL. This section is where users write, execute, and save SQL queries against their data. This is analogous to interactive query editors found in other database systems.

While Tables are certainly a crucial component within Snowflake, they are managed within a database, not directly through a top-level navigation item. Therefore, it is incorrect to select it. Tables are contained within Schemas within a Database.

In summary, the top navigation of the Snowflake UI focuses on the high-level components required for data management: the structured data storage (Databases), the computational resources (Warehouses), and the interactive interface for analysis (Worksheets). These selections represent the core elements necessary to effectively utilize the Snowflake platform.

Authoritative Links for Further Research:

Snowflake Documentation: Web Interface Overview: https://docs.snowflake.com/en/user-guide/ui-overview
Snowflake Documentation: Getting Started: https://docs.snowflake.com/en/getting-started

Question: 47 Exam Heist

What is the recommended Snowflake data type to store semi-structured data like JSON?

A.VARCHAR
B.RAW
C.LOB
D.VARIANT

Answer: D

Explanation:

The correct answer is D. VARIANT is the recommended Snowflake data type for storing semi-structured data like JSON. While VARCHAR, RAW, and LOB could technically store textual representations of JSON, they don't allow for efficient querying and manipulation of the internal structure. VARIANT, in contrast, is a Snowflake-specific data type designed to handle schemaless or semi-structured data, including JSON, Avro, ORC, and Parquet. It dynamically infers the schema from the ingested data, allowing users to access fields directly using dot notation within SQL queries. This capability significantly simplifies querying complex nested structures within JSON without prior schema definition or lengthy parsing procedures. Storing JSON as VARCHAR would require significant string manipulation and parsing functions to extract any specific values, making it highly inefficient and cumbersome for data analysis. Similarly, RAW and LOB types primarily store binary data and are not designed for querying or manipulation of the kind of data found in JSON. VARIANT also offers several performance advantages because it uses internal optimization techniques to speed up processing of semi-structured data. It allows users to benefit from all the analytical power of Snowflake without requiring the traditional ETL process of extracting data to a specific structured table. Furthermore,

Snowflake has native functions that are designed to work directly with the Variant data type for additional data transformations and querying. The VARIANT type is the best fit for schema evolution, allowing for additions, removal and changing of structure within the stored data without requiring any structural changes to the table itself. This flexibility and efficiency makes VARIANT the ideal choice for working with JSON and other semi-structured data within Snowflake.

For further information, please refer to the official Snowflake documentation:

Data Types
Working with Semi-structured Data

Question: 48 Exam Heist

Which of the following statements are true of Snowflake releases: (Choose two.)

A.They happen approximately weekly

B.They roll up and release approximately monthly, but customers can request early release application

C.During a release, new customer requests/queries/connections transparently move over to the newer version

D.A customer is assigned a 30 minute window (that can be moved anytime within a week) during which the system will be unavailable and customer is upgraded

Answer: AC

Explanation:

The correct answer is AC. Let's break down why.

Option A is Correct: Snowflake employs a frequent release cadence, pushing out new features and fixes roughly on a weekly basis. This agile approach allows for rapid iteration and improvement of the platform. These updates, being frequent, often focus on minor enhancements or bug fixes rather than major feature overhauls.

Option C is Correct: A key characteristic of Snowflake's service architecture is its seamless and transparent upgrades. During a release, user sessions and queries are automatically migrated to the newer version without any manual intervention required from the customer. This ensures continuous availability and minimal disruption to operations. Snowflake achieves this through its sophisticated architecture which supports rolling upgrades and manages connections gracefully.

Option B is Incorrect: While Snowflake does roll up releases, it doesn't happen on a monthly basis. Monthly roll-ups are not the standard procedure. Furthermore, while early access to features is often an option, that is a separate scenario. The general upgrades are frequent and managed by Snowflake with the described rolling update method.

Option D is Incorrect: Snowflake's upgrades are designed to be zero-downtime. Customers do not experience system unavailability windows, let alone assigned ones. The rolling update process manages the migration to the newer version transparently.

The concepts involved include agile software development, frequent deployments, zero-downtime deployments, and rolling upgrades. These allow cloud platforms such as Snowflake to continuously innovate and improve while maintaining high availability and service continuity.

For further research, you can check Snowflake's official documentation on releases and the architecture overview:

Snowflake Release Notes

Snowflake Architecture Overview

Question: 49 Exam Heist

Which of the following are common use cases for zero-copy cloning? (Choose three.)

A.Quick provisioning of Dev and Test/QA environments
B.Data backups
C.Point in time snapshots
D.Performance optimization

Answer: ABC

Explanation:

The correct answer is ABC because zero-copy cloning in Snowflake offers several benefits well-suited for the stated scenarios.

A. Quick provisioning of Dev and Test/QA environments: Zero-copy cloning allows for the rapid creation of identical database environments for development and testing. Instead of physically copying the data, which is time-consuming and costly, Snowflake creates a metadata pointer to the source data. This significantly reduces the time and resources required for setting up these environments, enabling quicker iteration cycles. (Source: Snowflake Documentation on Cloning)

B. Data backups: While not a traditional backup mechanism, zero-copy clones can act as point-in-time backups because they capture the state of the data at the time of cloning. These clones are read-only and can be used to recover lost or corrupted data without incurring the storage costs of a full physical backup. Changes in the original table don't impact the clone unless explicitly modified.

C. Point-in-time snapshots: Similar to backups, zero-copy cloning creates snapshots of data at the instant of cloning. This provides a historical record of the data, allowing users to access and analyze previous data states easily. This is valuable for auditing and comparing data across different periods. These snapshots are space-efficient because they only store changes from the source data.

D. Performance Optimization is not a direct use case for zero-copy cloning. While cloning itself is fast, it does not inherently optimize query performance. For performance optimization in Snowflake, considerations like query tuning, clustering, and using materialized views are more relevant.

Therefore, options A, B, and C are the primary and common applications of zero-copy cloning. It supports faster environment provisioning, enables data recovery scenarios, and facilitates historical data analysis, all while minimizing resource utilization. The use of metadata pointers to avoid physical data duplication is at the core of this efficiency.

Question: 50 Exam Heist

If a Small Warehouse is made up of 2 servers/cluster, how many servers/cluster make up a Medium Warehouse?

A.4
B.16
C.32
D.128

Answer: A

Explanation:

A Snowflake warehouse's size directly corresponds to the compute power it offers, which is achieved by allocating a certain number of compute servers or clusters. The sizing is generally based on a doubling pattern. A Small warehouse, by Snowflake's definition, indeed uses 1 cluster which contains 2 servers. Moving up to the next size, a Medium warehouse, involves doubling the resources of the preceding size. Consequently, a Medium warehouse will have twice the number of servers as a Small warehouse. Therefore, since a Small warehouse has 2 servers, a Medium warehouse will have 2 * 2 = 4 servers or 2 clusters (1 cluster == 2 servers). This principle of scaling by doubling continues for larger warehouse sizes as well. This allows Snowflake to provide scalable computing resources, where users can choose their warehouse size according to workload demand and scaling needs. Selecting the correct warehouse size is crucial for achieving optimal performance and cost-effectiveness, and understanding this doubling pattern can help users plan their Snowflake resource usage. Choosing a warehouse too small might cause latency while choosing an oversized warehouse would lead to overspending on compute credits.

Snowflake documentation regarding warehouse sizing can be found here:
https://docs.snowflake.com/en/user-guide/warehouses-overview.html#warehouse-sizes and
https://docs.snowflake.com/en/user-guide/warehouses-considerations.html#size-of-the-warehouse

Question: 51 Exam Heist

True or False: When a data share is established between a Data Provider and a Data Consumer, the Data Consumer can extend that data share to other Data
Consumers.

A.True
B.False

Answer: B

Explanation:

The statement is **False**. Snowflake's data sharing mechanism is designed with a focus on security and controlled access. When a data provider shares data with a data consumer, the consumer receives read-only access to the shared database or objects. This access is granted directly by the provider. The data consumer does not inherit any ownership or administrative privileges over the shared data. Consequently, the consumer cannot further propagate or 're-share' the received data to other consumers. This design decision prevents unintended data proliferation and ensures that the data provider maintains full control over who has access to their data. The original data provider must explicitly grant access to additional consumers, thus maintaining the security and governance of the shared data. Allowing re-sharing would create significant security risks and complicate data lineage tracking, which is vital for compliance and governance. Snowflake's data sharing is based on the concept of access control lists (ACLs) at the share level, ensuring that only explicitly authorized consumers can access the data. The shared data is virtualized and no actual data copying occurs when a share is established; therefore the consumer does not "own" a copy.

Further reading on Snowflake's data sharing can be found at:

Snowflake documentation: https://docs.snowflake.com/en/user-guide/data-sharing-intro
Snowflake data sharing best practices: https://community.snowflake.com/s/article/Best-Practices-for-Data-Sharing

Question: 52 Exam Heist

Which is true of Snowflake network policies? A Snowflake network policy: (Choose two.)

A.Is available to all Snowflake Editions
B.Is only available to customers with Business Critical Edition
C.Restricts or enables access to specific IP addresses
D.Is activated using an ALTER DATABASE command

Answer: AC

Explanation:

Here's a detailed justification for why options A and C are the correct choices for Snowflake network policies:

Snowflake network policies are a security feature that allows administrators to control network access to their Snowflake account. They operate by defining a list of allowed or blocked IP addresses, effectively creating a "whitelist" or "blacklist" for network connections. This mechanism is a fundamental aspect of network security in cloud environments, aligning with the principle of least privilege and perimeter defense. The use of IP whitelisting helps prevent unauthorized access attempts from unknown or untrusted networks.

Option A is correct because network policies are not restricted to a specific Snowflake edition; they are a standard security feature available across all Snowflake editions. This broad availability enables all users to enhance the security posture of their Snowflake environments regardless of their chosen edition.

Option C is also correct because the core functionality of a Snowflake network policy is to restrict or enable access based on specific IP addresses. By defining a set of IP addresses or CIDR notation ranges, administrators can precisely control which networks can access Snowflake resources, adding a vital layer of network security.

Option B is incorrect; network policies are not exclusive to the Business Critical Edition, making this statement inaccurate.

Option D is incorrect because network policies are activated using the ALTER ACCOUNT command, not the ALTER DATABASE command. This crucial difference indicates that network policies are an account-level configuration, impacting all databases within the account.

Therefore, the correct choices are A and C, as they accurately describe the core features and availability of Snowflake network policies.

Here are some authoritative links for further research:

Snowflake Documentation on Network Policies: https://docs.snowflake.com/en/sql-reference/sql/alter-account.html#parameters (This link directly to the documentation section about activating network policies).
Snowflake Documentation on CREATE NETWORK POLICY: https://docs.snowflake.com/en/sql-reference/sql/create-network-policy.html (Detailed information on creating network policies).
Snowflake Documentation on Network Policy: https://docs.snowflake.com/en/sql-reference/ddl-network-policy.html (Overview of network policies).

Question: 53 Exam Heist

True or False: Snowflake charges additional fees to Data Providers for each Share they create.

A.True
B.False

Answer: B

Explanation:

The statement "Snowflake charges additional fees to Data Providers for each Share they create" is **False**. Snowflake's data sharing model is designed to be cost-effective and encourage data sharing. Data providers incur costs primarily based on the compute resources they use for data processing, storage, and services. However, creating a share itself is not an activity that triggers additional charges. The costs are borne by the data consumer when they query the shared data, not the data provider for making the share available. Snowflake's cost model emphasizes consumption-based billing for compute and storage, ensuring providers are not penalized for broader data distribution. Shares are essentially pointers to data; they don't duplicate the data or consume additional provider resources until that shared data is actively queried by a consumer. Think of it like providing a library card; the library doesn't charge for giving out cards but for borrowing the books. This is a key aspect of Snowflake's value proposition, promoting efficient and cost-effective data sharing. The provider retains control of their data, and only pays for the underlying resources when the data is being processed or stored, whether by them or a consumer through sharing.

For further research, refer to the following resources:

Snowflake Cost Optimization: https://www.snowflake.com/cost-optimization/
Snowflake Secure Data Sharing: https://docs.snowflake.com/en/user-guide/data-sharing-intro.html
Understanding Snowflake's Billing: https://www.snowflake.com/pricing/

Question: 54 Exam Heist

Query results are stored in the Result Cache for how long after they are last accessed, assuming no data changes have occurred?

A.1 Hour
B.3 Hours
C.12 hours
D.24 hours

Answer: D

Explanation:

The correct answer is D. 24 hours. Snowflake's Result Cache is a performance optimization feature that stores the results of queries. This cache is designed to accelerate subsequent executions of the same query, significantly reducing processing time and associated costs. The core concept is that if a query is executed and the underlying data hasn't changed, the system can return the cached result rather than re-processing the data.

The critical point for the question is the cache's expiration period. Snowflake's documentation specifies that cached results remain valid for 24 hours from their last access, assuming no modifications to the base tables have occurred. If data changes or if the 24-hour limit is exceeded, the cache is invalidated.

This 24-hour period represents a balance between providing fast access to frequently run queries and ensuring data consistency. It prevents users from working with stale information while optimizing query performance. The Result Cache doesn't actively flush or remove cached results before this period as long as data remains unchanged and it has been accessed within the 24-hour window. Options A, B and C are all incorrect because they don't match Snowflake's specified caching expiration period. Therefore, 24 hours is the correct duration for which result-cache results are stored in Snowflake, given no data changes and active access within this period.

Authoritative links:

Snowflake Documentation on Result Caching: https://docs.snowflake.com/en/user-guide/querying-cached-data.html
Snowflake Performance optimization: https://www.snowflake.com/blog/snowflake-performance-optimization/

Question: 55 Exam Heist

A role is created and owns 2 tables. This role is then dropped. Who will now own the two tables?

A.The tables are now orphaned
B.The user that deleted the role
C.SYSADMIN
D.The assumed role that dropped the role

Answer: D

Explanation:

The correct answer is **D. The assumed role that dropped the role**. Snowflake's access control model is based on roles, and object ownership is tied to roles, not users. When a role is dropped, its ownership of objects isn't transferred to the user who executed the DROP ROLE command. Instead, Snowflake automatically transfers ownership of the objects to the role that was active when the DROP ROLE statement was executed. This is typically the role that has privileges to drop other roles, often a system-level or higher-level custom role. The user who issues the drop command is not inherently linked to the ownership transfer; it's the active role at the time that matters. Therefore, the ownership is passed to the assumed role (i.e. the role that is in effect) when the DROP command occurs. This mechanism ensures that data and objects remain controlled within the Snowflake system even when specific roles are removed. Understanding this concept is fundamental to proper access control and object management in Snowflake.

Refer to these authoritative links for further information:

Snowflake Documentation on Access Control
Snowflake Documentation on Roles
Snowflake Documentation on Dropping Roles

Question: 56 Exam Heist

Which of the following connectors are available in the Downloads section of the Snowflake Web Interface (UI)? (Choose two.)

A.SnowSQL
B.ODBC
C.R
D.HIVE

Answer: AB

Explanation:

The correct answer is A and B. Snowflake's web interface provides direct links for downloading client connectors and drivers to interact with the Snowflake data platform. Among the available options, SnowSQL

and ODBC connectors are prominently featured in the downloads section.

SnowSQL is Snowflake's command-line client, allowing users to execute SQL queries and perform various administrative tasks. It is essential for scripting and automating workflows involving Snowflake. The direct download link in the web UI streamlines access to this core tool.

ODBC (Open Database Connectivity) is a standard API for accessing database systems. Snowflake provides an ODBC driver, enabling applications and tools that support ODBC to connect to Snowflake. This connection allows BI (Business Intelligence) tools, data integration platforms, and custom applications to seamlessly access data stored within Snowflake. Similar to SnowSQL, the web UI provides easy access to this connector for download.

Options C and D, representing R and HIVE connectors, are not directly listed in the "Downloads" section of the web interface. While Snowflake can be connected to with these using specific drivers or connectors, they're not as prominently featured for download within the Snowflake web interface. Instead, those are typically acquired through other channels, like CRAN (Comprehensive R Archive Network) for R and through driver repositories for HIVE.

Therefore, the primary purpose of the downloads section in the Snowflake UI is to provide readily accessible download links for core connectivity tools directly supported by Snowflake, making options A (SnowSQL) and B (ODBC) the correct choice.

Authoritative Links:

Snowflake Documentation - Downloading Drivers & Clients: https://docs.snowflake.com/en/user-guide/ui-downloads.html
Snowflake Documentation - SnowSQL: https://docs.snowflake.com/en/user-guide/snowsql.html
Snowflake Documentation - ODBC Driver: https://docs.snowflake.com/en/user-guide/odbc.html

Question: 57 Exam Heist

Which of the following DML commands isn't supported by Snowflake?

A.UPSERT
B.MERGE
C.UPDATE
D.TRUNCATE TABLE

Answer: A

Explanation:

The correct answer is A, UPSERT. Snowflake natively supports MERGE, UPDATE, and TRUNCATE TABLE as DML (Data Manipulation Language) operations. MERGE allows conditional insertion, update, or deletion of data based on source and target table comparisons, effectively performing upsert-like operations. UPDATE modifies existing rows within a table based on specified conditions. TRUNCATE TABLE removes all rows from a table, effectively resetting it, but maintains the table schema. While many databases support a direct UPSERT operation, Snowflake does not have a specific command named UPSERT. Instead, the desired upsert functionality is achieved using the MERGE command with appropriate matching criteria and insert/update clauses. This distinction is crucial for understanding how Snowflake handles data modification. Therefore, while the outcome of an upsert can be achieved in Snowflake, it isn't through a dedicated "UPSERT" keyword, making it the correct option among the provided choices.

https://docs.snowflake.com/en/sql-reference/sql/merge.htmlhttps://docs.snowflake.com/en/sql-reference/sql/update.htmlhttps://docs.snowflake.com/en/sql-reference/sql/truncate-table.html

Question: 58 Exam Heist

Which of the following statements is true of zero-copy cloning?

A.Zero-copy clones increase storage costs as cloning the table requires storing its data twice

B.All zero-copy clone objects inherit the privileges of their original objects

C.Zero-copy cloning is licensed as an additional Snowflake feature

D.At the instance/instant a clone is created, all micro-partitions in the original table and the clone are fully shared

Answer: D

Explanation:

Okay, let's break down why option D is the correct answer regarding Snowflake's zero-copy cloning.

Justification:

Zero-copy cloning in Snowflake is a metadata-only operation. When you clone a table, schema, or database, you are not physically copying the underlying data files (micro-partitions). Instead, a new object is created that points to the existing data. Initially, both the source and the clone share the same micro-partitions. At the precise moment the clone is created, both the original object and the clone have full access to the same physical data. Any changes made to either the original or the clone after that point will result in new micro-partitions being created for that specific object, and thus data will diverge.

Option A is incorrect because zero-copy cloning is designed to minimize storage costs by avoiding duplication. Option B is incorrect; while objects generally inherit privileges from their parent, these are not automatically and fully inherited in every case and the cloning operation does not guarantee inheritance of all permissions. Option C is incorrect as zero-copy cloning is a core feature of Snowflake and not an additional licensed feature.

Essentially, zero-copy cloning leverages Snowflake's architecture to be highly efficient. The clone acts as a snapshot of the source data at the time of cloning, without physically duplicating the underlying data. This functionality allows for rapid creation of development or testing environments or creating backups with virtually no storage cost for the clone initially.https://docs.snowflake.com/en/user-guide/tables-cloning.htmlhttps://www.snowflake.com/blog/zero-copy-cloning/

Question: 59 Exam Heist

True or False: When a user creates a role, they are initially assigned ownership of the role and they maintain ownership until it is transferred to another user.

A.True

B.False

Answer: B

Explanation:

The statement is false. In Snowflake, while a user who creates a role initially has the privilege to manage that role, they do not inherently own it. The ownership of a role in Snowflake, like other securable objects, is

granted to the role itself, not the user who created it. When a user creates a role (e.g., via CREATE ROLE), that role becomes the object owner and not the user. The user gains the OWNERSHIP privilege on that role, which allows them to perform all operations on it, including granting privileges to other roles/users and transferring the OWNERSHIP privilege to another role. If the user, who has the OWNERSHIP privilege grants OWNERSHIP to another role and revokes their privilege, they no longer have the right to manage the role. This is a critical distinction in Snowflake's security model: it employs role-based access control (RBAC). Ownership in Snowflake is about control and is assigned directly to roles, emphasizing delegation and access control management via roles. The user does not maintain ownership until it is transferred; the role does. The original creator can relinquish or lose their ownership privileges, and they will no longer be able to modify or drop the role without explicit permission. This separation ensures security and prevents unintentional loss of control over critical objects. The OWNERSHIP privilege can be transferred via the GRANT OWNERSHIP command.

Relevant documentation links to further research:

Snowflake Documentation on Access Control
Snowflake Documentation on Roles
Snowflake Documentation on Object Ownership

Question: 60 Exam Heist

The Query History in the Snowflake Web Interface (UI) is kept for approximately:

A.60 minutes
B.24 hours
C.14 days
D.30 days
E.1 year

Answer: C

Explanation:

The correct answer is C, 14 days. Snowflake retains query history in the Web UI for a period of 14 days by default. This history includes details about executed queries, such as the query text, user who executed it, start and end times, and other execution metrics. This time limit is a built-in feature of the platform's auditing and monitoring capabilities, designed to provide a recent view of query activity for debugging and analysis. While Snowflake offers query history through other methods like the QUERY_HISTORY view, the Web UI retains it for a shorter period. The 14-day limit serves as a compromise between providing useful recent data and managing storage costs and performance. Information older than 14 days will still be available through other mechanisms, but is not displayed directly in the web UI's Query History. Users needing longer retention should utilize those other mechanisms for tracking historical query data. Options A, B, D, and E are incorrect as they do not reflect the default retention period of the Web UI query history in Snowflake. The specific 14-day limit helps facilitate efficient troubleshooting and performance analysis. This period also avoids overloading the UI with a massive backlog of historical information.

For further research, refer to the official Snowflake documentation on query history:

Snowflake Query History
Snowsight query history

Question: 61 Exam Heist

To run a Multi-Cluster Warehouse in auto-scale mode, a user would:

A.Configure the Maximum Clusters setting to Auto-Scale
B.Set the Warehouse type to Auto
C.Set the Minimum Clusters and Maximum Clusters settings to the same value
D.Set the Minimum Clusters and Maximum Clusters settings to the different values

Answer: D

Explanation:

The correct answer is **D. Set the Minimum Clusters and Maximum Clusters settings to different values.** Multi-cluster warehouses in Snowflake are designed to handle concurrent user requests and varying workloads efficiently. Auto-scale mode, specifically, enables a warehouse to dynamically adjust its cluster count based on demand. When the minimum and maximum cluster settings are set to different values, Snowflake can automatically scale the warehouse up to the specified maximum to accommodate increased query load and down to the minimum when the load decreases. Setting these to the same value essentially disables scaling, as the warehouse will always maintain that fixed number of clusters. Option A is incorrect because there is no "Auto-Scale" setting for Maximum Clusters itself. Option B is incorrect because "Auto" is not a valid warehouse type in Snowflake. Option C would prevent the auto-scaling functionality, rendering the warehouse a static, single-cluster or multi-cluster (if the value is > 1), not an auto-scaling, multi-cluster warehouse. Therefore, to activate auto-scaling, a range, not a fixed point, is required, necessitating that the minimum and maximum cluster settings be set to different values. This range provides the flexibility needed for the warehouse to adjust its resources according to the actual workload.

For more information, refer to the official Snowflake documentation:

Multi-cluster warehouses: https://docs.snowflake.com/en/user-guide/warehouses-multicluster
Auto-scaling: https://docs.snowflake.com/en/user-guide/warehouses-auto-suspend-resume

Question: 62 Exam Heist

Which of the following terms best describes Snowflake's database architecture?

A.Columnar shared nothing
B.Shared disk
C.Multi-cluster, shared data
D.Cloud-native shared memory

Answer: C

Explanation:

The correct answer is C, Multi-cluster, shared data. Snowflake's architecture is not a traditional shared-nothing or shared-disk system. Instead, it employs a unique, cloud-optimized approach. Option A, Columnar shared nothing, while using a columnar storage format, doesn't fully describe Snowflake's architecture as its compute clusters can access the same data, meaning not strictly "shared nothing". Option B, Shared disk, is also not accurate because Snowflake decouples storage and compute. Option D, Cloud-native shared memory, is incorrect because Snowflake does not rely on shared memory between compute nodes.

Snowflake's core architecture is best characterized by its multi-cluster, shared data design. This means multiple independent compute clusters can operate concurrently on the same underlying data storage layer. Each cluster is an independent compute unit that can scale up or down as needed. This separation of storage and compute allows for independent scaling, optimal resource utilization, and high concurrency. The shared

data layer, stored in cloud object storage (like AWS S3, Azure Blob Storage, or Google Cloud Storage), is accessible by all compute clusters. This allows multiple users or queries to access the same data concurrently without contention. Further, it eliminates the need for data movement or duplication, reducing costs and complexity. Snowflake's architecture is designed specifically for cloud environments, taking full advantage of the scalability, elasticity, and availability offered by cloud platforms. This distinguishes it from traditional on-premise database architectures.

For further research on Snowflake's architecture, you can refer to these resources:

1. **Snowflake's Official Documentation:** https://docs.snowflake.com/en/ - Search for "Architecture Overview" or "Snowflake Architecture" for detailed explanations.
2. **Snowflake Blog:** https://www.snowflake.com/blog/ - Explore blog posts related to Snowflake's architecture and features.
3. **Cloud Data Platform Concepts:** Resources that discuss the concepts of shared-nothing, shared-disk, and cloud-native architectures, such as vendor-specific cloud documentation for S3, Azure Blob Storage, and Google Cloud Storage.

Question: 63 Exam Heist

Which of the following are options when creating a Virtual Warehouse? (Choose two.)

A.Auto-drop
B.Auto-resize
C.Auto-resume
D.Auto-suspend

Answer: CD

Explanation:

The correct options when creating a Snowflake Virtual Warehouse are **Auto-resume** (C) and **Auto-suspend** (D). Virtual Warehouses in Snowflake are designed to provide compute resources for executing queries and data loading tasks. Auto-resume allows the warehouse to automatically start up when a query is submitted, thereby avoiding manual activation, which is crucial for on-demand use. Similarly, auto-suspend automatically shuts down the warehouse after a period of inactivity, thus optimizing costs by preventing unnecessary compute charges during periods of low activity. These features align with cloud principles of elasticity and pay-per-use. Auto-resize (B) is related to scaling but is managed differently, typically by setting min/max warehouse sizes or enabling scaling policies not directly on creation; whereas auto-drop (A) is not a standard feature of virtual warehouse management. The auto-suspend time can be configured. The auto-resume and auto-suspend features are critical for cost management and efficiency in Snowflake, allowing users to focus on data analysis rather than managing infrastructure.

For further understanding, refer to the Snowflake documentation:

Virtual Warehouse Concepts
CREATE WAREHOUSE

Question: 64 Exam Heist

A Virtual Warehouse's auto-suspend and auto-resume settings apply to:

A.The primary cluster in the Virtual Warehouse

B.The entire Virtual Warehouse

C.The database the Virtual Warehouse resides in

D.The queries currently being run by the Virtual Warehouse

Answer: B

Explanation:

The correct answer is **B. The entire Virtual Warehouse.**

Auto-suspend and auto-resume are features that govern the lifecycle of a Snowflake virtual warehouse, not its underlying components or related databases. These settings dictate when the warehouse is automatically paused (suspended) due to inactivity and when it automatically starts again (resumes) upon receiving a new query. The entire virtual warehouse, including all of its compute resources, is subject to these settings; it is not a granular control that applies to individual clusters within the warehouse (if such are utilized internally), a particular database the warehouse might be querying, or specific queries themselves.

Auto-suspend is crucial for managing costs as it prevents unnecessary compute charges when the warehouse is idle. Conversely, auto-resume ensures that the warehouse is readily available for query execution, minimizing latency. Both features work together to provide a cost-effective and user-friendly experience. It is a key part of Snowflake's strategy to enable on-demand compute resources scaling.

The virtual warehouse is designed to be an abstraction layer over the underlying compute infrastructure, enabling users to focus on data analysis without managing the intricacies of resource allocation. Thus, applying these settings to the warehouse as a whole is consistent with Snowflake's overarching design principles. The settings don't pertain to specific elements within the warehouse but instead control the availability of the whole unit.

For further research, refer to the official Snowflake documentation:https://docs.snowflake.com/en/user-guide/warehouses-auto-suspend-resumehttps://docs.snowflake.com/en/sql-reference/sql/create-warehouseThese links detail virtual warehouse behavior and parameter settings.

Question: 65 Exam Heist

Fail-safe is unavailable on which table types? (Choose two.)

A.Temporary

B.Transient

C.Provisional

D.Permanent

Answer: AB

Explanation:

The correct answer is A. Temporary and B. Transient. Snowflake's Fail-safe feature, designed for data recovery after a specified time travel period, is only applicable to **permanent** tables. Temporary and transient tables are specifically designed for short-term data storage needs, thus they are excluded from Fail-safe protection. Temporary tables exist only within a session, while transient tables are available for longer but lack fail-safe coverage. Fail-safe ensures data is retained for 7 days after time travel, providing an additional safety net beyond traditional time travel. Permanent tables inherently require this extra layer of data safety. Therefore, due to their ephemeral nature, temporary and transient tables do not require and cannot benefit from Fail-safe's long-term data protection capabilities. The core architectural design of temporary and

transient tables is focused on efficiency and short-term storage, making Fail-safe an unsuitable feature.

Authoritative links:

Snowflake Documentation on Understanding Fail-safe: https://docs.snowflake.com/en/user-guide/data-availability.html#understanding-failsafe
Snowflake Documentation on Table Types: https://docs.snowflake.com/en/user-guide/tables-types.html

Question: 66 Exam Heist

Which of the following objects is not covered by Time Travel?

A.Tables
B.Schemas
C.Databases
D.Stages

Answer: D

Explanation:

Time Travel in Snowflake is a powerful feature that allows you to access historical data for a specified period. It enables you to recover deleted or modified objects, providing data protection and auditing capabilities. However, not all objects are covered by Time Travel.

Tables, schemas, and databases are all primary data-bearing objects within Snowflake and are subject to Time Travel. This means that if a table, schema, or entire database is dropped or modified, you can restore it to a previous state within the defined time retention period (which can range from 0 to 90 days for Enterprise Edition or higher).

Stages, on the other hand, are named locations where data files are stored or loaded into Snowflake, external or internal. Stages themselves do not store data in the same way as tables; they point to locations holding file data. Therefore, while the data within files located in a stage can be versioned (depending on how those files are managed outside of Snowflake), the stage object itself is not directly covered by Time Travel. The files in stages are not directly protected by Snowflake's time travel and would require external tools to restore these. Therefore, if a stage is dropped, it can't be recovered via Time Travel. To restore files, you would need a restore mechanism setup for the external storage system.

In conclusion, Tables, Schemas, and Databases have Time Travel functionality, allowing recovery from changes or deletion. Stages do not and are not covered by Snowflake's Time Travel feature.

Authoritative Links for further research:

Snowflake Documentation on Time Travel: https://docs.snowflake.com/en/user-guide/data-time-travel.html
Snowflake Documentation on Stages: https://docs.snowflake.com/en/user-guide/data-load-stages.html

Question: 67 Exam Heist

True or False: Micro-partition metadata enables some operations to be completed without requiring Compute.

A.True
B.False

Answer: A

Explanation:

The statement is **True**. Micro-partition metadata in Snowflake significantly reduces the need for compute resources for certain operations. This is because Snowflake automatically gathers and maintains metadata about the data stored within each micro-partition. This metadata includes information such as the minimum and maximum values for each column, the number of distinct values, and null counts.

When a query is executed, Snowflake's query optimizer uses this metadata to perform data pruning. This means that it can intelligently skip entire micro-partitions that do not contain data relevant to the query's filter conditions. For example, if a query requests data where column_A > 100, and a micro-partition's metadata shows that the maximum value for column_A within it is 50, then the entire partition is skipped without reading its contents.

This efficient data pruning based on metadata allows Snowflake to satisfy many query requests, especially those with filters, by only reading the relevant micro-partitions. This directly translates to reduced I/O operations, decreased data transfer, and significantly faster query performance. More importantly, this entire process often happens without involving any significant compute resource usage. Operations that heavily rely on metadata for optimization include filtering, COUNT DISTINCT, and MIN/MAX queries.

In essence, leveraging micro-partition metadata is a core feature of Snowflake's architecture that provides substantial performance gains and cost optimization by minimizing the necessity for full data scans and expensive compute operations.

Authoritative Links:

Snowflake Documentation on Micro-Partitions: https://docs.snowflake.com/en/user-guide/tables-clustering-micropartitions
Snowflake Documentation on Data Pruning: https://docs.snowflake.com/en/user-guide/querying-data-pruning

Question: 68 Exam Heist

Which of the following commands are not blocking operations? (Choose two.)

A.UPDATE
B.INSERT
C.MERGE
D.COPY

Answer: BD

Explanation:

The question asks for Snowflake commands that are non-blocking. In the context of databases, a blocking operation typically involves waiting for resources or locks to be released before proceeding, potentially hindering other concurrent operations. Conversely, a non-blocking operation does not wait and can immediately proceed or return, even if the requested resource is temporarily unavailable.

UPDATE, MERGE, and COPY commands in Snowflake are generally considered blocking operations. UPDATE and MERGE often require locking specific rows or tables while modifying data, which can create bottlenecks and delays for other operations targeting the same data. COPY, while potentially parallelized for loading files into Snowflake tables, can still be blocking, especially if multiple load jobs target the same table

simultaneously due to its transactional nature.

On the other hand, INSERT operations in Snowflake are typically designed to be non-blocking, particularly when appending data to a table. While there might be temporary contention, for instance, when appending to the same micro-partition as other inserts, Snowflake's architecture, leveraging micro-partitions, minimizes the blocking impact. Hence, while inserts can be part of a transaction, individual append inserts are generally considered non-blocking.

Therefore, INSERT (B) and COPY (D), when considered in the context of non-blocking load operations, especially with modern file loading techniques within Snowflake, are not inherently blocking operations in comparison to UPDATE and MERGE which require more stringent data integrity and lock handling procedures.

Authoritative Links for further research:

Snowflake Documentation on Data Loading: https://docs.snowflake.com/en/user-guide/data-load (Explore sections on COPY command behavior and performance considerations.)
Snowflake Transaction Management: https://docs.snowflake.com/en/user-guide/transactions (Understand how transactions impact concurrency and blocking.)
Snowflake Architecture: https://www.snowflake.com/product/architecture/ (Review Snowflake's micro-partitioning to understand the concept of minimized blocking during inserts).

Question: 69 Exam Heist

Which of the following is true of Snowpipe via REST API? (Choose two.)

A.You can only use it on Internal Stages
B.All COPY INTO options are available during pipe creation
C.Snowflake automatically manages the compute required to execute the Pipe's COPY INTO commands
D.Snowpipe keeps track of which files it has loaded

Answer: CD

Explanation:

Let's analyze why options C and D are correct regarding Snowpipe via REST API, and why A and B are incorrect.

Option C is correct: Snowpipe, regardless of the method of ingestion (REST API or auto-ingest), operates on a serverless compute model. Snowflake dynamically allocates and manages the necessary compute resources to execute the COPY INTO commands defined within the pipe. This relieves users from the burden of managing virtual warehouses for continuous data loading. This serverless nature is a core feature of Snowflake and a key aspect of its ease of use.

Option D is correct: A fundamental function of Snowpipe is to track which files have already been loaded, avoiding duplicate ingestion. This is critical for ensuring data integrity in a continuous loading scenario. The internal tracking mechanism prevents the same data from being loaded multiple times, even if files are re-submitted or appear multiple times in the stage.

Option A is incorrect: Snowpipe via REST API can load data from both internal (Snowflake-managed) stages and external stages (e.g., S3, Azure Blob Storage, Google Cloud Storage). The choice of stage location doesn't restrict REST API usage. Therefore, this is a key capability of Snowpipe and a strong driver for utilizing external stages to ingest from external data lakes.

Option B is incorrect: While Snowpipe utilizes COPY INTO commands, not all COPY INTO options are available

during pipe creation. Some options, such as those related to specific transformations, might require additional considerations or alternative implementation via other mechanisms or features. Therefore, it is important to understand the available options in pipe creation and limitations.

In summary, Snowpipe with the REST API leverages a serverless approach for data loading (C) and maintains meticulous tracking of ingested files to ensure data quality (D). It is not limited to internal stages (A), and does not offer full feature parity with all COPY INTO options (B).

Authoritative Links for Further Research:

1. **Snowflake Documentation on Snowpipe:** https://docs.snowflake.com/en/user-guide/data-load-snowpipe-intro.html
2. **Snowflake Documentation on Snowpipe REST API:** https://docs.snowflake.com/en/user-guide/data-load-snowpipe-rest.html

Question: 70 Exam Heist

Snowflake recommends, as a minimum, that all users with the following role(s) should be enrolled in Multi-Factor Authentication (MFA):

A.SECURITYADMIN, ACCOUNTADMIN, PUBLIC, SYSADMIN
B.SECURITYADMIN, ACCOUNTADMIN, SYSADMIN
C.SECURITYADMIN, ACCOUNTADMIN
D.ACCOUNTADMIN

Answer: D

Explanation:

The correct answer is D, ACCOUNTADMIN. Snowflake, like most cloud platforms, emphasizes securing privileged access. The ACCOUNTADMIN role in Snowflake is the most powerful, possessing complete control over the entire account, including user management, resource provisioning, and data access policies. Therefore, implementing Multi-Factor Authentication (MFA) for users with this role is crucial to safeguard against unauthorized access and potential compromise. It serves as a critical additional layer of security beyond just a username and password, requiring a secondary verification method such as a mobile app code, significantly reducing the risk of account breaches. While roles like SECURITYADMIN and SYSADMIN also have high privileges, the ACCOUNTADMIN role's overarching control makes its protection the utmost priority. Mandating MFA for ACCOUNTADMIN aligns with best practices in cloud security, reinforcing the principle of least privilege and limiting the blast radius of a potential compromise. While MFA is beneficial for all users, its necessity for the ACCOUNTADMIN role is non-negotiable, considering the significant impact a breach could have.

Further reading:

Snowflake Documentation on Multi-Factor Authentication
Snowflake Documentation on Access Control

Question: 71 Exam Heist

When can a Virtual Warehouse start running queries?

A.12am-5am
B.Only during administrator defined time slots

C.When its provisioning is complete

D.After replication

Answer: C

Explanation:

The correct answer is C: "When its provisioning is complete." A virtual warehouse in Snowflake represents the compute resources used to execute queries. It needs to be fully allocated and ready before it can process any data requests. This process is known as provisioning. Option A, specifying a time window, is incorrect as virtual warehouses aren't limited to specific hours, but rather are on-demand and scaled based on need. Option B, administrator-defined time slots, is also incorrect. While administrators control the overall access and configuration of warehouses, there's no functionality that limits query execution to predefined time slots. Option D, after replication, is irrelevant to the core functionality of a warehouse. Replication deals with data availability across different Snowflake regions and accounts, not with the initial readiness of compute resources. The warehouse must be fully created and its compute nodes available before queries can be processed. The provisioning stage includes allocating necessary resources like CPU, memory, and disk space. Once provisioning is complete, the warehouse is in a state ready to accept incoming requests. This is a fundamental concept in cloud computing, where resources need to be allocated before they can perform their designated task.

Here are some authoritative links for further research:

Snowflake Documentation - Virtual Warehouses: https://docs.snowflake.com/en/user-guide/warehouses.html
Snowflake Documentation - Warehouse States: https://docs.snowflake.com/en/sql-reference/info-schema/warehouses.html (Look for the STATE column descriptions, particularly "RUNNING")
Cloud Computing Concepts - Resource Provisioning: [Search for "cloud resource provisioning" on trusted cloud resources like AWS, Azure, or Google Cloud documentation pages] This provides background on the typical lifecycle of compute resources.

Question: 72 Exam Heist

True or False: Users are able to see the result sets of queries executed by other users that share their same role.

A.True

B.False

Answer: B

Explanation:

The statement that users can see the result sets of queries executed by other users sharing the same role in Snowflake is **false**. Snowflake's security model emphasizes data privacy and separation, even among users with the same role. Access control in Snowflake is not just about roles; it also includes object-level privileges. While a role grants permissions to perform actions on specific objects (like tables or views), it doesn't automatically grant access to the result sets of queries run by other users.

Each user's session in Snowflake operates independently. Query results are typically stored within the user's session context and are not visible to others. Snowflake does not save query results persistently by default, beyond what's needed for user experience and performance. For security reasons, sharing sensitive query outputs amongst users is not an inherent feature of the platform.

To achieve any form of data sharing from the results of one user's query with another, a user would need

explicit action like creating a share, a view, a table, or some sort of data replication or data-transfer mechanism.

Granting access to underlying data through roles allows multiple users to query the same source tables/views. The query results, however, remain isolated, reinforcing Snowflake's commitment to data security and access control. Simply having the same role does not grant access to the queries' data of other users. The Snowflake documentation clearly highlights the user-centric security model, where data access requires specific permissions and controls.

For further research, review the official Snowflake documentation on Role-Based Access Control (RBAC) and security concepts, particularly regarding user sessions, object privileges, and data sharing mechanisms. These resources will explain why merely sharing a role does not provide access to other users' query results.

https://docs.snowflake.com/en/user-guide/security-access-control-overviewhttps://docs.snowflake.com/en/user-guide/security-rbac

Question: 73 Exam Heist

True or False: The user has to specify which cluster a query will run on in a multi-cluster Warehouse.

A.True
B.False

Answer: B

Explanation:

The statement is false. In a Snowflake multi-cluster warehouse setup, users do not need to explicitly specify which cluster a query will run on. Snowflake's intelligent query routing automatically distributes queries across available clusters within the warehouse. This distribution is managed internally by Snowflake based on various factors like cluster load and query complexity, aiming to optimize performance and resource utilization. The user submits a query to the warehouse, not a specific cluster within it. Snowflake then handles the underlying resource allocation. This abstraction simplifies query execution for users, freeing them from the complexities of managing individual compute resources and providing a more streamlined and efficient experience. Snowflake's approach enables dynamic scaling and ensures high availability as queries are directed to clusters with capacity. Users benefit from consistent performance regardless of the underlying infrastructure changes. Multi-cluster warehouses are designed for concurrency and high availability, and explicit cluster selection would counteract these principles. Automatic load balancing is a cornerstone of Snowflake's architecture and ensures no single cluster is overwhelmed. Therefore, the responsibility of selecting an appropriate cluster for a query rests entirely on Snowflake's internal mechanisms, not the user.

Authoritative links for further research:

Snowflake Documentation - Multi-cluster Warehouses
Snowflake Documentation - Understanding Query Routing

Question: 74 Exam Heist

True or False: Pipes can be suspended and resumed.

A.True
B.False

Answer: A

Explanation:

The correct answer is **A. True**. Snowflake Pipes, used for continuous data ingestion, can indeed be suspended and resumed. This capability is crucial for managing data loading processes effectively. Suspending a pipe halts the automatic ingestion of data from the specified stage, preventing new files from being loaded into the target table. This is beneficial for maintenance windows, cost control, or when troubleshooting data pipeline issues. Resuming a pipe brings it back online, allowing it to start processing new data from the stage again. This provides operational flexibility and allows for controlled data loading schedules. Snowflake's architecture is designed with such pause/resume functions to offer granular control over resources and processes. The ALTER PIPE command in SQL is used to perform these actions, with syntax like ALTER PIPE <pipe_name> SET PIPE_EXECUTION_PAUSED = true/false;. This allows administrators to seamlessly manage data loading operations. This pause/resume ability avoids manual interventions, and is a core concept to facilitate reliable and scalable data pipelines. The ability to pause provides flexibility, preventing data processing during specific periods. It also allows for controlled data ingestion ensuring data consistency.

Authoritative links:

Snowflake Documentation on ALTER PIPE: https://docs.snowflake.com/en/sql-reference/sql/alter-pipe
Snowflake Documentation on Pipes Overview: https://docs.snowflake.com/en/user-guide/data-load-snowpipe-intro

Question: 75 Exam Heist

Which of the following languages can be used to implement Snowflake User Defined Functions (UDFs)? (Choose two.)

A.Java
B.Javascript
C.SQL
D.Python

Answer: AD

Explanation:

Snowflake supports User-Defined Functions (UDFs) to extend its built-in functionalities. UDFs allow users to implement custom logic within the Snowflake environment. Currently, Snowflake primarily supports UDFs written in Java and Python. Java UDFs leverage the JVM for execution, enabling the use of Java's extensive libraries and capabilities. Python UDFs, on the other hand, are executed through a secure and isolated runtime environment within Snowflake, facilitating the use of popular data science and analysis libraries. SQL, while the primary language for data manipulation within Snowflake, cannot be used to implement the procedural logic required for a UDF itself. Javascript can be used for Stored Procedures, but not for UDF's. Therefore, the correct languages for implementing Snowflake UDFs from the options provided are Java and Python.

The selection of either language depends on factors such as the developer's familiarity, existing libraries, and the specific nature of the function being implemented. For complex computations or integrations with Java-based systems, Java UDFs might be preferred, whereas Python would typically be more suitable for data analysis and transformation tasks using libraries like pandas, numpy, and sklearn. This ability to use multiple languages for custom functions showcases the flexibility and openness of the Snowflake platform.

Refer to the official Snowflake documentation for further clarification on supported languages for UDFs and their respective use cases:

Snowflake UDF Documentation: https://docs.snowflake.com/en/sql-reference/udf-overview
Java UDFs: https://docs.snowflake.com/en/developer-guide/udf/java/udf-java
Python UDFs: https://docs.snowflake.com/en/developer-guide/udf/python/udf-python

Question: 76 Exam Heist

When should you consider disabling auto-suspend for a Virtual Warehouse? (Choose two.)

A.When users will be using compute at different times throughout a 24/7 period
B.When managing a steady workload
C.When the compute must be available with no delay or lag time
D.When you do not want to have to manually turn on the Warehouse each time a user needs it

Answer: BC

Explanation:

The correct answers are **B and C**. Auto-suspend is a feature that automatically shuts down a virtual warehouse when it's inactive for a specified period. While generally beneficial for cost management, there are situations where disabling it is advantageous. Option B, "When managing a steady workload," is correct because a consistent, predictable workload benefits from a continuously running warehouse, avoiding the overhead of repeated start-ups. Frequent auto-suspend and resume cycles introduce latency and can disrupt performance when processing ongoing, consistent tasks. Option C, "When the compute must be available with no delay or lag time," is also correct because auto-suspend incurs a delay when a warehouse needs to restart. For time-sensitive operations or critical applications, the lag from reactivation can be detrimental. Thus, when consistent availability and immediate responsiveness are paramount, keeping the warehouse running is necessary.

Option A, “When users will be using compute at different times throughout a 24/7 period” is incorrect because auto-suspend is designed to manage situations where usage varies throughout the day. The ability to suspend during inactive periods helps to manage costs in such situations. Option D, "When you do not want to have to manually turn on the Warehouse each time a user needs it" is also incorrect as that is the whole point of the auto suspend and auto resume process. It eliminates the need to manually start warehouses and also ensures cost savings.

Here are authoritative links for further research:

Snowflake Documentation on Virtual Warehouses: https://docs.snowflake.com/en/user-guide/warehouses - This is the official Snowflake documentation, providing in-depth information about virtual warehouses and their management, including auto-suspend.
Snowflake Blog on Cost Management: https://www.snowflake.com/blog/snowflake-cost-optimization-guide/ - This article provides information on optimizing costs in Snowflake, which includes using auto-suspend appropriately for various usage patterns.

Question: 77 Exam Heist

Which of the following are valid approaches to loading data into a Snowflake table? (Choose all that apply.)

A.Bulk copy from an External Stage
B.Continuous load using Snowpipe REST API
C.The Snowflake Web Interface (UI) data loading wizard
D.Bulk copy from an Internal Stage

Answer: ABC

Explanation:

The correct answer is ABC. Let's break down why each option is a valid method for loading data into a Snowflake table.

A. Bulk copy from an External Stage: Snowflake supports loading large datasets efficiently from external cloud storage locations (like Amazon S3, Google Cloud Storage, or Azure Blob Storage). This involves using the COPY command, which reads data from a specified stage and loads it into a target table. An external stage points to these external storage locations. This is a standard method for bulk data ingestion, well-suited for initial loading or periodic updates of large datasets.

B. Continuous load using Snowpipe REST API: Snowpipe is Snowflake's continuous data ingestion service. It automatically loads new data files as soon as they are made available in a specified stage. The Snowpipe REST API is used to manage and configure this automated process. This approach is ideal for real-time or near-real-time data streaming use cases, where new data needs to be loaded into Snowflake promptly.

C. The Snowflake Web Interface (UI) data loading wizard: Snowflake provides a user-friendly web interface for various tasks, including loading data. The UI wizard simplifies the data loading process for smaller datasets or ad-hoc tasks. It allows users to upload local files directly or specify data files in an internal stage, guiding them through the steps required to load data into a table.

D. Bulk copy from an Internal Stage: While valid, it is not the primary method for data loading. An internal stage is a location within Snowflake's own storage space. It is generally used as a temporary holding area for data before loading or as a destination for transformed data. Though COPY command can also load from an internal stage, it’s usually a middle step. Therefore, while possible, it's not as direct a method as loading from external stages or using Snowpipe. This is why D is not part of the most voted answers.

Why ABD and BCD are not the best fit: ABD includes an option (D) that is valid but not a common direct method, making ABC a better representation of the primary and popular approaches. Similarly, BCD excludes the significant and common practice of direct bulk loading from an External Stage (A), thus making ABC the best fit.

In summary, ABC represents the core methods for loading data: bulk from external storage, continuous with Snowpipe, and ad-hoc through the UI.

Authoritative Links for further research:

Snowflake Documentation on Loading Data: https://docs.snowflake.com/en/guides-overview-data-loading
Snowflake Documentation on COPY command: https://docs.snowflake.com/en/sql-reference/sql/copy-into-table
Snowflake Documentation on Snowpipe: https://docs.snowflake.com/en/user-guide/data-load-snowpipe
Snowflake Documentation on Stages: https://docs.snowflake.com/en/user-guide/data-load-stages

Question: 78 Exam Heist

If auto-suspend is enabled for a Virtual Warehouse, the Warehouse is automatically suspended when:

A.All Snowflakes sessions using the Warehouse are terminated.
B.The last query using the Warehouse completes.
C.There are no users logged into Snowflake.
D.The Warehouse is inactive for a specified period of time.

Answer: D

Explanation:

The correct answer is **D. The Warehouse is inactive for a specified period of time.** Auto-suspend in Snowflake is a feature designed to optimize cost by automatically suspending a virtual warehouse when it's not actively processing queries. This mechanism is triggered by a defined inactivity period, which is configured at the warehouse level. If a warehouse isn't executing queries or performing other active operations for the specified timeframe, Snowflake initiates the suspension process. This means no more compute resources are utilized, thereby halting consumption of compute credits. Options A, B, and C are incorrect; while they might reflect periods of inactivity, they don't represent the actual trigger for auto-suspend. Termination of Snowflake sessions or completion of the last query only means that the warehouse could become inactive. Similarly, the presence of users logged into Snowflake is irrelevant, as a user may be logged in without issuing queries requiring compute. The core concept underpinning auto-suspend is inactivity-based scaling, a common cost-optimization technique in cloud computing. This ensures resources are allocated and consumed only when actively needed. This practice aligns with pay-as-you-go consumption models and is a key best practice for efficient cloud resource management. By automatically suspending idle warehouses, users avoid incurring unnecessary compute costs.

For further information, please refer to the official Snowflake documentation:

https://docs.snowflake.com/en/user-guide/warehouses-auto-suspend-resume
https://docs.snowflake.com/en/sql-reference/sql/create-warehouse

Question: 79 Exam Heist

True or False: Multi-Factor Authentication (MFA) in Snowflake is only supported in conjunction with Single Sign-On (SSO).

A.True
B.False

Answer: B

Explanation:

The statement that Multi-Factor Authentication (MFA) in Snowflake is only supported with Single Sign-On (SSO) is false. Snowflake provides robust MFA capabilities that can be used independently of SSO. While SSO often incorporates MFA as part of its authentication flow, it's not a prerequisite for using MFA within Snowflake. Snowflake allows users to directly enable MFA using various methods, like authenticator apps, without relying on an SSO provider. This direct MFA setup adds a security layer to individual Snowflake accounts, requiring a secondary verification method in addition to passwords. This enhances account protection, making it significantly harder for unauthorized users to gain access, regardless of whether SSO is in use. The flexibility of Snowflake's authentication approach allows organizations to choose the best security options to fit their needs. It gives organizations the flexibility to apply MFA to a broader range of user access models, not just those routed via SSO. Using MFA directly is a crucial security practice, especially in cases where direct account access is unavoidable. Therefore, the assertion that MFA depends solely on SSO within Snowflake is inaccurate.

https://docs.snowflake.com/en/user-guide/security-mfa.htmlhttps://docs.snowflake.com/en/security/authentication.html

Question: 80 Exam Heist

The number of queries that a Virtual Warehouse can concurrently process is determined by (Choose two.):

A.The complexity of each query
B.The CONCURRENT_QUERY_LIMIT parameter set on the Snowflake account
C.The size of the data required for each query
D.The tool that is executing the query

Answer: AC

Explanation:

The correct answer is A and C. The number of concurrent queries a Snowflake Virtual Warehouse can process is primarily dictated by the complexity of the individual queries (A) and the amount of data they need to process (C). More complex queries, involving intricate joins or aggregations, require more processing resources, potentially limiting concurrency. Similarly, queries acting on very large datasets consume more resources, reducing the number of parallel operations possible. Snowflake's architecture dynamically allocates resources based on the demands of executing queries, which makes the size of data and the complexity directly influential on query concurrency. While Snowflake does have a CONCURRENT_QUERY_LIMIT parameter, it's applied to the whole Snowflake account, not individual warehouses. This parameter acts as a hard cap on the total number of concurrently executing queries across all warehouses within the account, it does not dictate concurrency within a particular warehouse. The tool running the query (D) does not influence warehouse concurrency as that is something that is managed server-side.

Essentially, resource contention within the warehouse is the primary determining factor, and complex queries and large datasets increase the likelihood of that contention. Snowflake dynamically scales resources within a warehouse to a certain extent, but even with scaling, resource limitations will still affect how many queries can run simultaneously.

Further Research:

Snowflake Documentation on Virtual Warehouses: https://docs.snowflake.com/en/user-guide/warehouses
Snowflake Documentation on Query Processing: https://docs.snowflake.com/en/user-guide/querying-concepts
Snowflake Documentation on Resource Monitors: https://docs.snowflake.com/en/user-guide/resource-monitors - While not directly addressing concurrency limits, they show the resource management aspect

Thank you

Thank you for being so interested in the premium exam material.
I'm glad to hear that you found it informative and helpful.

But Wait

I wanted to let you know that there is more content available in the full version.
The full paper contains additional sections and information that you may find helpful, and I encourage you to download it to get a more comprehensive and detailed view of all the subject matter.

Download Full Version Now

Future is Secured

100% Pass Guarantee

24/7 Customer Support

Mail us - support@examheist.com

Verified Updates

Lifetime Updates!

Total: **1254 Questions**

Link: https://examheist.com/papers/snowflake/snowpro-core