DUMPSBOSS.

SnowPro Core Certification Exam

Snowflake COF-C02

Version Demo

Total Demo Questions: 90

Total Premium Questions: 909

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Snowflake Architecture	151
Topic 2, Snowflake Account and Security	223
Topic 3, Data Loading and Unloading	184
Topic 4, Data Transformation	93
Topic 5, Data Protection and Data Sharing	124
Topic 6, Performance Concepts	134
Total	909

QUESTION NO: 1

Which of the following indicates that it may be appropriate to use a clustering key for a table? (Select TWO).

A. The table contains a column that has very low cardinality

B. DML statements that are being issued against the table are blocked

C. The table has a small number of micro-partitions

D. Queries on the table are running slower than expected

E. The clustering depth for the table is large

ANSWER: D E

Explanation:

“Queries on the table are running slower than expected” can indicate that a clustering key may be beneficial, particularly for a large table with frequent, selective queries that filter or range-scan on consistent columns. Effective clustering improves micro-partition pruning, allowing Snowflake to scan fewer micro-partitions and potentially reduce both query elapsed time and compute consumption. Before defining a key, query patterns should be reviewed to identify the columns most commonly used in selective predicates, joins, or ordering-related access patterns.

“The clustering depth for the table is large” is also an important indicator. Clustering depth measures the overlap of micro-partitions for a table’s clustering dimensions. A high average depth means values are spread across numerous overlapping micro-partitions, reducing pruning effectiveness. Defining an appropriate clustering key can help Snowflake organize data around commonly queried dimensions and lower overlap over time through automatic clustering. Snowflake recommends evaluating clustering information and query workload together, especially for large tables where the performance benefit can justify the ongoing compute cost of reclustering. See the Snowflake clustering key documentation and SYSTEM$CLUSTERING_INFORMATION reference.

QUESTION NO: 2

How are serverless features billed?

A. Per second multiplied by an automatic sizing for the job

B. Per minute multiplied by an automatic sizing for the job, with a minimum of one minute

C. Per second multiplied by the size, as determined by the SERVERLESS_FEATURES_SIZE account parameter

D. Serverless features are not billed, unless the total cost for the month exceeds 10% of the warehouse credits, on the account

ANSWER: A

Explanation:

Per second multiplied by an automatic sizing for the job is correct. Snowflake manages the compute resources behind serverless features rather than requiring the customer to select and operate a virtual warehouse. Snowflake automatically determines an appropriate compute size for the workload, can adjust that size as workload demands change, and records the resulting resource consumption in credits. The charge therefore reflects both the automatically selected compute capacity and the time it is used, with usage metered at per-second granularity. This model is used by a range of Snowflake-managed capabilities, including features such as serverless tasks, automatic clustering, search optimization, and materialized-view maintenance. The exact credit rate can vary by feature and cloud region, but users do not set a serverless warehouse size through an account parameter. To monitor these charges, account administrators can use the serverless usage views in the ACCOUNT_USAGE schema, which provide usage history by service type. See Snowflake’s serverless feature credit usage documentation and the SERVERLESS_TASK_HISTORY reference for details on metering and usage visibility.

QUESTION NO: 3

Which of the following Snowflake features provide continuous data protection automatically? (Select TWO).

A. Internal stages

B. Incremental backups

C. Time Travel

D. Zero-copy clones

E. Fail-safe

ANSWER: C E

Explanation:

Time Travel and **Fail-safe** provide Snowflake's automatic, continuous data-protection capabilities. Time Travel preserves historical versions of data after changes, enabling users to query, restore, or clone data that existed at a specified point within the configured retention period. This supports recovery from unintended updates, deletes, or dropped objects without requiring customers to create and manage traditional backups.

After the Time Travel retention period ends, Fail-safe provides an additional non-configurable recovery window for eligible permanent data. During this period, Snowflake maintains the data so that Snowflake support can assist with recovery in disaster-recovery situations. Together, these features form layers of Snowflake-managed data protection: Time Travel is the self-service recovery mechanism during its retention window, and Fail-safe is a subsequent emergency recovery mechanism. Snowflake documentation describes both features as part of its data lifecycle and recovery model, with automatic protection occurring as data is changed. See Understanding and using Time Travel and Understanding Fail-safe.

QUESTION NO: 4

Which Snowflake feature is used for both querying and restoring data?

A. Cluster keys

B. Time Travel

C. Fail-safe

D. Cloning

ANSWER: B

Explanation:

Time Travel is the Snowflake feature used both to query historical data and to restore data that was changed or deleted. Within an object's configured data-retention period, Time Travel preserves prior versions of data and enables SQL queries against a specific point in time, an offset from the current time, or a statement identifier. For example, historical table contents can be queried with the `AT` or `BEFORE` clause. It also supports recovery workflows: an accidentally dropped table, schema, or database can be restored with `UNDROP`, and prior table data can be recovered by creating or replacing a table from a historical point in time. This combination of historical querying and self-service recovery is the core purpose of Time Travel. The retention period depends on the Snowflake edition and object configuration; standard permanent objects generally have one day by default, while higher editions can support longer retention. Snowflake documents Time Travel capabilities and recovery operations at Time Travel and data retention and provides the SQL syntax for historical queries at AT | BEFORE.

QUESTION NO: 5

Which virtual warehouse configuration requires the maximum number of clusters to be equal to the minimum number of clusters?

A. Auto-scale mode

B. Maximized mode

C. Standard scaling policy

D. Economy scaling policy

ANSWER: B

Explanation:

Maximized mode is the multi-cluster warehouse configuration that requires the maximum cluster count to equal the minimum cluster count. In this mode, Snowflake starts and keeps the specified number of clusters available whenever the warehouse is running, rather than dynamically adding or removing clusters in response to demand. Setting the same minimum and maximum values defines a fixed cluster count, ensuring that all configured clusters are provisioned immediately when the warehouse resumes. This approach is intended for workloads where consistently high concurrency is expected and the organization prefers predictable, immediately available compute capacity over demand-based scaling behavior. The warehouse can still suspend according to its auto-suspend configuration; however, while it is running, the fixed number of clusters established by the matching minimum and maximum settings is maintained. Snowflake documents Maximized mode as a multi-cluster warehouse mode and specifies that it is selected by setting the minimum and maximum number of clusters to the same value. See the Snowflake documentation on multi-cluster warehouses and the ALTER WAREHOUSE command configuration parameters.

QUESTION NO: 6

Which of the following features are available with the Snowflake Enterprise edition? (Choose two.)

A. Database replication and failover

B. Automated index management

C. Customer managed keys (Tri-secret secure)

D. Extended time travel

E. Native support for geospatial data

ANSWER: A D

Explanation:

Extended time travel is an Enterprise Edition capability because it increases the maximum Time Travel data-retention period from the standard one-day setting to as many as 90 days for permanent databases, schemas, and tables. This longer recovery window supports restoring or querying historical data further back in time, subject to Snowflake's retention configuration and applicable storage considerations. Snowflake documents the edition-dependent retention capability in its Time Travel documentation: Time Travel and Fail-safe.

Database replication and failover are included in the Enterprise-oriented availability model tested by this question. Replication enables Snowflake to copy eligible database objects and data to another account, supporting data availability and disaster-recovery architectures. A replicated environment can be used as part of a failover design when an outage affects the primary environment. Snowflake's replication documentation describes the database replication concepts, supported objects, and replication architecture: Introduction to replication. Together, these capabilities provide both a substantially expanded historical-data recovery window and a mechanism for maintaining replicated data for continuity planning.

QUESTION NO: 7

Regardless of which notation is used, what are considerations for writing the column name and element names when traversing semi-structured data?

A. The column name and element names are both case-sensitive.

B. The column name and element names are both case-insensitive.

C. The column name is case-sensitive but element names are case-insensitive.

D. The column name is case-insensitive but element names are case-sensitive.

ANSWER: D

Explanation:

The column name is case-insensitive but element names are case-sensitive. When accessing semi-structured data in Snowflake, the name of the column that contains a VARIANT, OBJECT, or ARRAY value follows Snowflake's normal identifier rules. For unquoted identifiers, Snowflake resolves the column name without regard to the letter case used in the query. For example, a column created as `src` can be referenced as `SRC`, `src`, or `Src`. Once traversal moves into the semi-structured value, however, object keys and element names must match their stored case. A JSON key named `customerName` is therefore distinct from `customername` or `CUSTOMERNAME`. This rule applies regardless of whether dot notation or bracket notation is used to access the path. Correct casing is essential because semi-structured fields preserve the names supplied in the source data, allowing JSON and similar formats to retain their original key structure. Snowflake documents this distinction in its guidance for querying semi-structured data and in its identifier rules: Querying semi-structured data and Identifier requirements.

QUESTION NO: 8

Which action will scale-out a Snowflake virtual warehouse?

A. Removing 1 cluster from a multi-cluster warehouse

B. Adding 2 additional clusters to a multi-cluster warehouse

C. Resizing a warehouse from size Small to X-Large

D. Resizing a warehouse from size Large to Medium

ANSWER: B

Explanation:

Adding 2 additional clusters to a multi-cluster warehouse is the action that scales out a Snowflake virtual warehouse. In Snowflake, scale-out means increasing the number of compute clusters available to the warehouse, rather than increasing the size of the individual clusters. Multi-cluster warehouses use identical clusters to handle greater concurrent query workloads. When additional clusters are added, Snowflake can distribute queued and incoming queries across those clusters, improving concurrency and reducing queueing without changing the resources assigned to a single cluster.

This capability is configured through the warehouse's multi-cluster settings, including its minimum and maximum cluster counts and the scaling policy. Snowflake can start clusters automatically as demand rises, up to the configured maximum, and suspend unneeded clusters as demand falls. Each active cluster has the same warehouse size, so adding clusters expands total concurrent compute capacity horizontally. Snowflake documents multi-cluster warehouses as the mechanism for scaling compute resources out to meet concurrent-user and concurrent-query demand. See the Snowflake documentation on multi-cluster warehouses and virtual warehouse overview.

QUESTION NO: 9

A single user of a virtual warehouse has set the warehouse to auto-resume and auto-suspend after 10 minutes. The warehouse is currently suspended and the user performs the following actions:

1. Runs a query that takes 3 minutes to complete

2. Leaves for 15 minutes

3. Returns and runs a query that takes 10 seconds to complete

4. Manually suspends the warehouse as soon as the last query was completed

When the user returns, how much billable compute time will have been consumed?

A. 4 minutes

B. 10 minutes

C. 14 minutes

D. 24 minutes

ANSWER: C

Explanation:

14 minutes is correct. The first query causes the suspended warehouse to auto-resume. It runs for 3 minutes and then remains running but idle for the configured 10-minute auto-suspend period. Therefore, this first warehouse-resume interval consumes 13 minutes of compute time before the warehouse suspends automatically.

When the user returns 15 minutes later, the warehouse is suspended, so submitting the second query causes another auto-resume. Although that query runs for only 10 seconds and the warehouse is manually suspended immediately afterward, Snowflake applies a 60-second minimum billing charge each time a virtual warehouse resumes. This second resume interval is consequently billed as 1 minute. The total billable compute time is 13 minutes plus 1 minute, which equals 14 minutes.

Snowflake bills virtual warehouses by the second while they are running, subject to the one-minute minimum each time a warehouse starts or resumes. Auto-suspend limits the running idle period, while manual suspension stops further compute consumption after the query completes. See Snowflake's warehouse billing documentation and auto-suspend and auto-resume documentation.

QUESTION NO: 10

What are characteristics of the ownership privilege when it is granted on a regular Snowflake schema? (Select TWO).

A. It is automatically granted to the role that creates a database object within the schema.

B. It allows a role to manage grants on the schema.

C. It can be transferred from one role to another for a specific schema.

D. It grants the ability to query data from the schema.

E. It must be granted to a role in order to alter warehouse settings.

ANSWER: B C

Explanation:

"It allows a role to manage grants on the schema." is correct because OWNERSHIP is a special, powerful privilege that gives the owning role full control of the schema. This includes the ability to manage privileges on the schema itself, such as granting or revoking schema-level privileges. Ownership is distinct from ordinary usage or object-access privileges because it represents control of the securable object and is required for many administrative actions involving that object.

"It can be transferred from one role to another for a specific schema." is also correct. Snowflake supports transferring ownership of securable objects, including schemas, by using the `GRANT OWNERSHIP` command with the appropriate object type and target role. A transfer moves the OWNERSHIP privilege to the receiving role; Snowflake documents important considerations around existing outbound privileges when ownership is transferred. In a regular schema, ownership of

individual objects remains independently controlled: the role that creates an object owns that created object, rather than automatically receiving ownership of the schema. See Snowflake's documentation for GRANT OWNERSHIP and its access-control privilege reference.

QUESTION NO: 11

What activities can a user with the ORGADMIN role perform? (Select TWO).

A. Create information_schema in a database

B. View usage information for all accounts in the organization.

C. Enable database cloning for an account in the organization.

D. Enable database replication for an account in the organization.

E. View micro-partition information for all accounts in the organization.

ANSWER: B D

Explanation:

The ORGADMIN role is an organization-level role intended for administration across the Snowflake organization rather than routine object administration within one account. It can access organization-wide usage and billing data, allowing an organization administrator to monitor consumption, costs, and account activity across all accounts in the organization. This data is exposed through the shared SNOWFLAKE database's organization usage views, subject to the role and organization configuration. Snowflake documents this responsibility as part of the ORGADMIN role's ability to view usage information for the organization.

ORGADMIN also has a required role in configuring replication between accounts. In particular, organization-level administration is used to enable replication for the relevant accounts, after which account-level administrators configure replication groups or failover groups and manage replicated objects. Therefore, enabling database replication for an account in the organization is an organization administration capability. These permissions support centralized governance of cross-account data continuity while leaving database-level implementation and access controls to account roles. See Snowflake's ORGADMIN role documentation and its replication and failover documentation.

QUESTION NO: 12

Which types of subqueries does Snowflake support? (Select TWO).

A. Uncorrelated scalar subqueries in WHERE clauses

B. Uncorrelated scalar subqueries in any place that a value expression can be used

C. EXISTS, ANY / ALL, and IN subqueries in WHERE clauses: these subqueries can be uncorrelated only

D. EXISTS, ANY / ALL, and IN subqueries in where clauses: these subqueries can be correlated only

E. EXISTS, ANY /ALL, and IN subqueries in WHERE clauses: these subqueries can be correlated or uncorrelated

ANSWER: A B E

Explanation:

Snowflake supports uncorrelated scalar subqueries wherever a value expression is permitted. Consequently, *Uncorrelated scalar subqueries in WHERE clauses* is supported because a scalar subquery can supply the single value used by a predicate in a WHERE condition. The broader statement, *Uncorrelated scalar subqueries in any place that a value expression can be used*, is also supported and reflects Snowflake's documented rule directly. For example, a WHERE predicate can compare a column to the single value returned by an uncorrelated aggregate subquery.

Snowflake also supports *EXISTS, ANY /ALL, and IN subqueries in WHERE clauses: these subqueries can be correlated or uncorrelated*. A correlated subquery references a column from the outer query and is evaluated in relation to its outer row; an uncorrelated subquery is independent of the outer query. This flexibility allows EXISTS, IN, and comparison predicates using ANY or ALL to express membership, existence, and set-comparison filtering patterns. The question is technically ambiguous because the first two correct statements overlap: the first is a specific instance of the second. Snowflake's subquery documentation describes these supported forms and their placement rules. See Working with Subqueries and WHERE.

QUESTION NO: 13

UESTION NO: 241

What SQL command would be used to view all roles that were granted to user.1?

A. show grants to user USER1;

B. show grants of user USER1;

C. describe user USER1;

D. show grants on user USER1;

ANSWER: A

Explanation:

`SHOW GRANTS TO USER USER1;` is the Snowflake SQL command used to inspect grants assigned directly to the specified user. In Snowflake's role-based access control model, users receive access through roles, and the `SHOW GRANTS TO USER` form returns the roles granted to that user. The result set identifies each granted role and includes grant metadata such as the grantor and the date the role was granted. This makes it the appropriate command for reviewing a user's directly assigned role memberships during access reviews, troubleshooting, or administration. The user name is supplied after the `USER` keyword; unquoted identifiers such as `USER1` are interpreted case-insensitively and stored in uppercase by Snowflake. Executing the statement requires sufficient privileges to view the applicable grants, such as an administrative role with the necessary access-control visibility. For the command syntax and supported `SHOW GRANTS` forms, see the Snowflake SHOW GRANTS documentation. For context on role assignment and Snowflake's access-control model, see Access control overview.

QUESTION NO: 14

Which categories are included in the execution time summary in a Query Profile? (Select TWO).

A. Pruning

B. Spilling

C. Initialization

D. Local Disk I/O

E. Percentage of data read from cache

ANSWER: C D

Explanation:

The Query Profile execution time summary breaks total query execution into named processing categories, helping users identify where a query spent time after compilation. **Initialization** is one of those categories. It represents the execution-stage setup work performed before the query's main processing begins, such as preparing resources and internal execution structures.

Local Disk I/O is also an execution time summary category. It measures time associated with reading data from or writing data to the local disk resources used by the virtual warehouse during query execution. This category is particularly useful when investigating queries whose operators need temporary local storage or must access data that is not already available in memory.

Snowflake presents these timing categories in the Query Profile so that execution time can be attributed to a type of activity rather than viewed only as a single elapsed-time value. The detailed profile then allows investigation of the individual operator nodes that contribute to the summarized timing. See Snowflake's Query Profile documentation and its execution-time guidance for the documented timing breakdown.

QUESTION NO: 15

What is the expiration period tor a file URL used to access unstructured data in cloud storage?

A. The remainder of the session

B. An unlimited amount of time

C. The length of time specified in the expiration_time argument

D. The same length of time as the expiration period for the query results cache

ANSWER: B

Explanation:

An unlimited amount of time is correct because a Snowflake file URL is a permanent URL that identifies a file stored in an internal or external stage. Snowflake generates this type of URL with the `BUILD_STAGE_FILE_URL` function. The URL itself does not contain a time-based expiration and therefore does not become invalid simply because a session ends, a specified duration elapses, or cached query results expire. Access through a file URL still requires the requester to authenticate to Snowflake and have the necessary privileges on the stage and related database objects. Consequently, continued usability depends on authorization and the continued existence of the referenced file and stage, rather than on an expiry timestamp embedded in the URL. Snowflake documents file URLs as one of its URL types for unstructured data and distinguishes their permanent nature from time-limited URL mechanisms. See Snowflake's types of URLs available to access files and the BUILD_STAGE_FILE_URL function reference.

QUESTION NO: 16

What is a fundamental characteristic of Snowflake micro-partitions?

A. They can be read directly as files.

B. They serve as an index for Snowflake tables.

C. They are sized based on Time Travel requirements.

D. Once established they cannot be changed.

ANSWER: D

Explanation:

Once established they cannot be changed. is a fundamental characteristic because Snowflake micro-partitions are immutable storage units. When data is loaded into a table, Snowflake automatically organizes it into compressed, columnar micro-partitions and records metadata about the values contained in each partition. Snowflake does not modify an existing micro-partition in place. Instead, when a row is updated, deleted, or otherwise affected by DML, Snowflake writes the changed data into new micro-partitions and retains the previous versions for the applicable data-retention period. This immutable design supports Snowflake's consistent query processing and is central to features such as Time Travel, which can access prior versions of table data while they remain within the retention window. Automatic clustering or reclustering can improve the physical organization of data over time, but it does so by creating replacement micro-partitions rather than

altering the original ones. Snowflake manages this lifecycle automatically, including eventual cleanup of obsolete partitions after retention requirements are met. See Snowflake's documentation on micro-partitions and data clustering and Time Travel.

QUESTION NO: 17

What are supported file formats for unloading data from Snowflake? (Choose three.)

A. XML

B. JSON

C. Parquet

D. ORC

E. AVRO

F. CSV

ANSWER: B C F

Explanation:

Snowflake supports unloading query results and table data to delimited text, semi-structured JSON, and columnar Parquet files. **CSV** is supported through a named file format or an inline file-format specification; its delimiter, quote handling, compression, null representation, and other output characteristics can be configured. **JSON** is also a supported unload target, allowing semi-structured values to be written as JSON records. **Parquet** is supported for efficient columnar output and is especially useful for downstream analytics systems that read Parquet. In a `COPY INTO <location>` command, Snowflake selects the output representation with `TYPE = CSV`, `TYPE = JSON`, or `TYPE = PARQUET`, either in a named file format object or in the command itself. Snowflake documentation lists these as the valid file-format types for unloading data, along with output-specific behavior such as file splitting and compression. See Snowflake's overview of unloading data and the COPY INTO <location> reference for supported syntax and file-format details.

QUESTION NO: 18

How does a scoped URL expire?

A. When the data cache clears.

B. When the persisted query result period ends.

C. The encoded URL access is permanent.

D. The length of time is specified in the expiration_time argument.

ANSWER: B

Explanation:

When the persisted query result period ends is correct. A scoped URL is generated by the `BUILD_SCOPED_FILE_URL` function and provides access to a staged file through the result of the query that created the URL. Snowflake ties the URL's validity to the persisted result for that query rather than making it indefinitely valid or setting its lifetime directly in the URL. Persisted query results generally remain available for up to 24 hours, so a scoped URL normally expires when that result-cache retention period ends. This design supports controlled, temporary file access in workflows such as loading or retrieving staged files without exposing permanent stage credentials. The precise availability can also depend on the persisted query result continuing to be available under Snowflake's result-caching behavior. See Snowflake's documentation for BUILD_SCOPED_FILE_URL, which states that scoped URLs expire when the persisted query result expires, and the Using persisted query results documentation for the retention period and related behavior.

QUESTION NO: 19

Which of the following Snowflake objects can be shared using a secure share? (Select TWO).

A. Materialized views

B. Sequences

C. Procedures

D. Tables

E. Secure User Defined Functions (UDFs)

ANSWER: D E

Explanation:

Tables can be added to a secure share, allowing consumer accounts to query the provider's data without the provider having to copy or move the data. The shared table remains in the provider account, while consumers access it through a database created from the share. This is a core Snowflake data-sharing capability and supports sharing data with other Snowflake accounts in a governed, read-only manner.

Secure User Defined Functions (UDFs) can also be included in a secure share. A secure UDF enables consumers to use the function while protecting its definition and implementation details from exposure. When a secure UDF is shared, it can be used with shared data objects according to the privileges and object dependencies configured by the provider. Snowflake requires the secure designation for functions made available through shares, consistent with its protections for logic exposed to consumers.

For the authoritative list of objects supported in shares and the requirements for sharing secure functions, see Snowflake's Introduction to Secure Data Sharing and CREATE SHARE reference.

QUESTION NO: 20

How can a user improve the performance of a single large complex query in Snowflake?

A. Scale up the virtual warehouse.

B. Scale out the virtual warehouse.

C. Enable standard warehouse scaling.

D. Enable economy warehouse scaling.

ANSWER: A

Explanation:

Scale up the virtual warehouse. Scaling up means selecting a larger warehouse size, which provides more compute resources, including additional CPU, memory, and local SSD cache capacity, to execute the query. This is the appropriate approach when one large, complex query needs more processing power or can benefit from greater parallelism. Snowflake documentation notes that increasing warehouse size can improve query performance, particularly for larger workloads; the warehouse can subsequently be resized down when the additional capacity is no longer needed to manage cost. Warehouse sizing should be tested with representative queries because the actual gain depends on the query plan, data volume, and whether the workload can execute in parallel. Snowflake also provides Query Profile so users can examine execution details and identify costly operators, spills to local or remote storage, and other bottlenecks while assessing the impact of a larger warehouse. See the Snowflake virtual warehouses overview and Query Profile documentation.

QUESTION NO: 21

In Snowflake's data security framework, how does column-level security contribute to the protection of sensitive information? (Select TWO).

A. Implementation of column-level security will optimize query performance.

B. Column-level security supports encryption of the entire database.

C. Column-level security ensures that only the table owner can access the data.

D. Column-level security limits access to specific columns within a table based on user privileges

E. Column-level security allows the application of a masking policy to a column within a table or view.

ANSWER: D E

Explanation:

Column-level security limits access to specific columns within a table based on user privileges by enabling organizations to apply protections precisely where sensitive values reside, rather than treating every field in a dataset identically. This supports least-privilege access patterns: users can work with the data needed for their responsibilities while sensitive fields receive controls appropriate to the user's active role and the organization's governance rules. In Snowflake, these controls are commonly implemented with policy-based mechanisms that are evaluated at query time, allowing a single governed table to serve different authorized audiences safely.

Column-level security also allows the application of a masking policy to a column within a table or view. A masking policy defines how a column value is presented for a given role or context. For example, an authorized role can receive an unmasked value, while another role receives a partially masked value or a tokenized representation. This helps protect personally identifiable information, financial identifiers, and other confidential attributes without requiring separate copies of the underlying data. Snowflake documents masking policies as a central mechanism for protecting sensitive column data and applying consistent governance controls across data objects. See Snowflake column-level security and Snowflake dynamic data masking.

QUESTION NO: 22

What can be used to view warehouse usage over time? (Select Two).

A. The load HISTORY view

B. The Query history view

C. The show warehouses command

D. The WAREHOUSE_METERING_HISTORY view

E. The billing and usage tab in the Snowflake web UI

ANSWER: D E

Explanation:

`WAREHOUSE_METERING_HISTORY` is the Account Usage view intended for analyzing virtual warehouse credit consumption over time. It returns metering records at hourly granularity, including the warehouse name, time interval, and credits used for warehouse compute and cloud services. This enables administrators to query, aggregate, and visualize usage across a selected date range, individual warehouse, or group of warehouses. The view is particularly useful for creating custom cost-monitoring reports and for attributing consumption to warehouses used by particular teams or workloads. Snowflake documents its columns, retention period, and latency in the WAREHOUSE_METERING_HISTORY reference.

The billing and usage tab in the Snowflake web UI provides a graphical, built-in way to inspect credit usage over time. Authorized users can use the account's usage information to review consumption trends without first writing SQL against an Account Usage view. It complements metering-history queries by supplying an interactive administrative view of usage and costs. Snowflake's cost-management documentation describes the available usage analysis interfaces and account usage data sources in its Understanding overall cost guide.

QUESTION NO: 23

Which privilege grants the ability to set a column-level security masking policy on a table or view column?

A. APPLY

B. CREATE

C. SET

D. MODIFY

ANSWER: A

Explanation:

APPLY is the privilege that authorizes a role to associate a masking policy with a table or view column. In Snowflake, a masking policy is a schema-level object that defines how returned column values are transformed according to the querying role or other context. To assign that policy through commands such as `ALTER TABLE ... ALTER COLUMN ... SET MASKING POLICY` or the corresponding view syntax, the role needs the policy's APPLY privilege. This permission is deliberately separate from policy creation so that policy authors and data-object administrators can be delegated distinct responsibilities. In practice, the role also requires sufficient privileges on the target table or view to alter its column, but APPLY is the policy-specific privilege that permits the policy assignment itself. Snowflake also supports account-level application privileges for centralized governance, but the fundamental privilege named in the question is APPLY. See Snowflake's documentation on assigning masking policies to columns and the access-control privilege reference.

QUESTION NO: 24

What type of query benefits the MOST from search optimization?

A. A query that uses only disjunction (i.e., OR) predicates

B. A query that includes analytical expressions

C. A query that uses equality predicates or predicates that use IN

D. A query that filters on semi-structured data types

ANSWER: C

Explanation:

A query that uses equality predicates or predicates that use IN benefits the most because the search optimization service is designed to accelerate highly selective lookup operations. Rather than scanning every micro-partition that might contain rows for a table, Snowflake maintains additional search-access data that helps identify the small set of micro-partitions likely to contain values matching the lookup condition. This can substantially reduce the amount of data scanned and the elapsed time for point-lookups, particularly on large tables where the predicate returns a small number of rows. Equality conditions such as `customer_id = 123` and membership conditions such as `customer_id IN (123, 456)` are explicit documented use cases for the service. The benefit is strongest when the lookup values are selective and the table is large enough that avoiding broad micro-partition scans has a meaningful effect. Snowflake also documents search optimization support for several other predicate types and data forms, but equality and `IN` lookups are core, high-impact scenarios. See Snowflake's Search Optimization Service documentation and its guidance on queries that can benefit from search optimization.

QUESTION NO: 25

Which Snowflake native tool can be used to diagnose and troubleshoot network connections?

A. SnowSQL

B. Snowflake Python connector

C. Snowsight

D. SnowCD

ANSWER: D

Explanation:

SnowCD is Snowflake's native connectivity diagnostic utility. It is specifically designed to test the network path between a client environment and a Snowflake account and to collect diagnostic information that helps identify connection-related problems. Administrators can run SnowCD from the affected client host to check whether required endpoints can be reached and whether the environment meets Snowflake connectivity requirements. Its output can help isolate issues involving DNS resolution, proxy configuration, TLS certificates, firewall rules, allowlists, or access to Snowflake services and cloud-storage endpoints used by the platform. This makes it the appropriate tool when the objective is diagnosing and troubleshooting the network connection itself, rather than simply connecting to Snowflake to run SQL commands or use a programmatic interface. Snowflake documents SnowCD as a command-line diagnostic tool and provides guidance for obtaining and running it before sharing results with Snowflake Support when needed. See the SnowCD documentation and Snowflake's client connectivity troubleshooting overview.

QUESTION NO: 26

Snowflake's access control framework combines which models for securing data? (Select TWO).

A. Attribute-based Access Control (ABAC 1

B. Discretionary Access Control (DAC)

C. Access Control List (ACL)

D. Role-based Access Control (RBAC)

E. Rule-based Access Control (RuBAC)

ANSWER: B D

Explanation:

Snowflake's access-control framework combines Discretionary Access Control (DAC) and Role-based Access Control (RBAC). RBAC is the central mechanism for administering access: privileges on securable objects, such as databases, schemas, tables, and warehouses, are granted to roles, and roles are then granted to users or other roles. This allows administrators to manage permissions consistently at scale, including through hierarchical role structures.

DAC is also part of Snowflake's model because an object's owner has the ability to grant privileges on that object to other roles. In Snowflake, ownership is represented by the OWNERSHIP privilege; the role holding this privilege has extensive control over the object, including the ability to manage access grants. Together, these models support controlled delegation while retaining role-centered administration. Snowflake documents that its access-control framework combines DAC and RBAC, and identifies the object owner's grant capabilities as the discretionary component. See the Snowflake access control overview and access-control privileges documentation.

QUESTION NO: 27

Which queries will require a running virtual warehouse? (Select TWO).

A. SELECT COUNT(*) FROM ;

B. DELETE FROM ;

C. UPDATE SET = ;

D. DROP TABLE ;

E. SELECT CURRENT_DATABASE();

F. Default Time Travel Retention is set to 90 days.Maximum Time Travel Retention is 7 days.Fail Safe retention time is 356 days.

ANSWER: A B C

Explanation:

`SELECT COUNT(*) FROM <table>;`, `DELETE FROM <table>;`, and `UPDATE <table> SET <column> = <value>;` require a running virtual warehouse because each operation must access and process table data. A `SELECT COUNT(*)` scans or otherwise evaluates table data to calculate the aggregate. `DELETE` and `UPDATE` are DML operations: although Snowflake implements changes by creating new micro-partitions rather than modifying existing ones in place, executing the statements still requires compute resources from a virtual warehouse. Snowflake warehouses provide the compute resources used to execute queries and DML statements, and a suspended warehouse must be resumed (manually or through auto-resume) before these statements can run. The supplied question says “Select TWO,” but it contains three operations that require warehouse compute, so all three are marked correct. In contrast, commands that operate only on object metadata or session context do not require a running warehouse. See Snowflake’s documentation on virtual warehouses and its DML command reference.

QUESTION NO: 28

Which statements are true about secure views? (Select TWO).

A. They can hide the view definition from unauthorized users.

B. They are required when sharing a view through a share.

C. They store a separate physical copy of the underlying table data.

D. They can only reference one base table.

E. They bypass all access-control checks on underlying objects.

ANSWER: A B

Explanation:

A secure view is designed to help protect sensitive data and business logic exposed through the view. Snowflake does not expose the secure view definition to unauthorized users, helping prevent consumers from inspecting the underlying query logic. Secure views are also required when sharing views through Secure Data Sharing.

Secure views can have performance differences from standard views because Snowflake applies additional safeguards to prevent information leakage through query optimization. A secure view does not store a separate materialized copy of the underlying data. See Snowflake documentation on working with secure views.

QUESTION NO: 29

What is the MINIMUM Snowflake edition that must be used in order to see the ACCESS_HISTORY view?

A. Standard

B. Enterprise

C. Business Critical

D. Virtual Private Snowflake (VPS)

ANSWER: B

Explanation:

Enterprise is the minimum required Snowflake edition because the `SNOWFLAKE.ACCOUNT_USAGE.ACCESS_HISTORY` view is an Enterprise Edition feature. This Account Usage view records object-access information for supported SQL statements, allowing administrators and security teams to analyze which users and roles accessed particular tables, views, columns, and other objects. It also provides query and object metadata that supports auditing, governance, and impact analysis. The feature is available to Enterprise Edition accounts and all higher editions, so an account running Enterprise can use the view without requiring Business Critical or Virtual Private Snowflake. As with other Account Usage views, access also depends on granting an appropriate role the required privileges, and data is subject to the documented Account Usage latency and retention behavior. Snowflake documents the view, its fields, supported access types, and availability in the ACCESS_HISTORY reference. The edition-level feature availability is also listed in Snowflake's editions and features documentation.

QUESTION NO: 30

What criteria does Snowflake use to determine the current role when initiating a session? (Select TWO).

A. If a role was specified as part of the connection and that role has been granted to the Snowflake user, the specified role becomes the current role.

B. If no role was specified as part of the connection and a default role has been defined for the Snowflake user, that role becomes the current role.

C. If no role was specified as part of the connection and a default role has not been set for the Snowflake user, the session will not be initiated and the log in will fail.

D. If a role was specified as part of the connection and that role has not been granted to the Snowflake user, it will be ignored and the default role will become the current role.

E. If a role was specified as part of the connection and that role has not been granted to the Snowflake user, the role is automatically granted and it becomes the current role.

ANSWER: A B

Explanation:

When a client initiates a Snowflake session, it can request a role in the connection parameters. If that requested role has been granted to the authenticating user, Snowflake activates it as the session's current role. This makes the role selection explicit and ensures that all subsequent statements run using the privileges available through that granted role.

If the connection does not request a role, Snowflake uses the user's configured default role when one is defined. A user's default role is set with the `DEFAULT_ROLE` user property and provides the normal initial role for sessions where the client does not override it. These role-selection rules allow administrators to establish an appropriate standard working role while still permitting users or applications to select another role that has already been granted to that user. Snowflake documents both the role parameter behavior for connections and the `DEFAULT_ROLE` user property in its role and user-management documentation. See Snowflake active roles documentation and ALTER USER optional parameters.

QUESTION NO: 31

What does the :: operator do when used with a VARIANT value?

A. Converts a VARIANT value into an object.

B. Extracts the raw value from the VARIANT column.

C. Casts the value from a VARIANT to a specified data type.

D. Generates a random sample from the VARIANT column.

ANSWER: C

Explanation:

The `::` operator is Snowflake's cast operator. When it is applied to a value stored in a `VARIANT`, it converts that value to the specified SQL data type, provided the underlying value can be converted. For example, if a semi-structured value is accessed with `src:customer.age`, writing `src:customer.age::INTEGER` returns the extracted value as an `INTEGER`. Similarly, `src:customer.name::VARCHAR` converts a string-like VARIANT value to `VARCHAR`. This shorthand is equivalent to using the `CAST(... AS ...)` syntax and is commonly used after navigating JSON, Avro, ORC, or Parquet-derived data held in VARIANT columns. Explicitly casting is important because values retrieved from semi-structured structures are represented as VARIANT unless they are converted, and typed results support appropriate SQL expressions, comparisons, calculations, and output formatting. Snowflake documents `::` as an alternative syntax for the `CAST` function and shows its use when querying semi-structured data. See CAST and :: and Querying semi-structured data.

QUESTION NO: 32

What is the purpose of multi-cluster virtual warehouses?

A. To create separate data warehouses to increase query optimization

B. To allow users the ability to choose the type of compute nodes that make up a virtual warehouse cluster

C. To eliminate or reduce Queuing of concurrent queries

D. To allow the warehouse to resize automatically

ANSWER: C

Explanation:

To eliminate or reduce Queuing of concurrent queries is correct. A Snowflake multi-cluster warehouse uses additional, identically sized compute clusters to handle increased concurrent query workloads. When demand rises beyond the capacity of a single cluster, Snowflake can start extra clusters, subject to the warehouse's configured minimum and maximum cluster settings. Each cluster accesses the same underlying Snowflake data, enabling more queries to execute simultaneously rather than waiting for computing resources to become available. This capability is designed primarily for concurrency scaling: workloads with many simultaneous users or queries can maintain more consistent performance and reduced queue times. In auto-scale mode, Snowflake dynamically starts and stops clusters based on workload demand; in maximized mode, it keeps all configured clusters running. Multi-cluster warehouses are therefore especially useful where query queuing is caused by high concurrency, rather than where a single query needs more compute resources. Snowflake documents that multi-cluster warehouses are intended to address query concurrency and queueing, while warehouse sizing addresses the resources available to execute individual queries. See Snowflake's multi-cluster warehouse documentation and its virtual warehouse overview.

QUESTION NO: 33

Which copy option is used to include column headers when unloading data to Parquet files?

A. PARSE_HEADER = TRUE

B. SKIP_HEADER = < integer >

C. HEADER = TRUE

D. HEADER = (' < header_1 > ' = ' < value_1 > ')

E. None of these; Snowflake does not support including a header row when unloading to Parquet files.

ANSWER: E

Explanation:

None of these; Snowflake does not support including a header row when unloading to Parquet files. Parquet is a binary, self-describing columnar format rather than a delimited text format. Its files store schema information, including field names, in Parquet metadata, so a separate, row-based header line is neither needed nor part of the file format. Snowflake's `HEADER` copy option controls inclusion of a header row for supported text output formats, but it is not supported when the unload target format is Parquet. Therefore, an unload to Parquet should be configured with the desired columns and aliases in the `COPY INTO` query; those resulting field names are represented through the Parquet schema metadata instead of a textual header record. This distinction matters for downstream readers because Parquet consumers obtain column names from the file schema, not from the first row of data. Snowflake documents the supported unload copy options, including the format restrictions for `HEADER`, in its COPY INTO <location> reference. For format behavior and the schema-oriented nature of Parquet output, see Snowflake's data unloading considerations.

QUESTION NO: 34

A Snowflake user wants to temporarily bypass a network policy by configuring the user object property MINS_TO_BYPASS_NETWORK_POLICY.

What should they do?

A. Use the SECURITYADMIN role.

B. Use the SYSADMIN role.

C. Use the USERADMIN role.

D. Contact Snowflake Support.

ANSWER: C

Explanation:

Use the `USERADMIN` role. This system-defined role is intended for user and role administration and has the capabilities needed to manage user objects that it owns or is authorized to administer. The temporary bypass is configured with `ALTER USER ... SET MINS_TO_BYPASS_NETWORK_POLICY = <number_of_minutes>`. The property defines a limited period during which the specified user can connect without the applicable network-policy restriction; after that period expires, the policy is enforced again automatically. This is useful when an authorized administrator must restore access for a user whose current network location is not permitted by a network policy, while retaining a time-bounded control rather than permanently changing or removing the policy. In practice, the role issuing `ALTER USER` must satisfy Snowflake's access-control requirements for that user object, such as having the required ownership or delegated administrative authority. Snowflake documents this property and its use in the ALTER USER reference. The responsibilities and privileges of the USERADMIN system role are described in Snowflake's system-defined roles documentation.

QUESTION NO: 35

Which Snowflake storage object can be used to store data beyond a single session and will not incur Fail-safe costs?

A. Permanent table

B. External table

C. Temporary table

D. Transient table

ANSWER: D

Explanation:

Transient table is correct because it persists independently of the user session while avoiding Snowflake Fail-safe storage. Unlike a session-scoped object, a transient table remains available until it is explicitly dropped, so it is suitable for intermediate, staging, or reloadable data that must be shared or retained across sessions. A transient table can also have a

Time Travel retention period, subject to the account and object configuration, allowing recent historical data to be queried or restored within that configured retention window.

The defining data-protection tradeoff is that transient tables do not have a Fail-safe period. Consequently, after the applicable Time Travel retention period has ended, Snowflake does not retain the table's data for its additional Fail-safe recovery window. Because no Fail-safe storage is provided for transient tables, there are no Fail-safe storage costs for their data. This makes transient tables appropriate when the data can be recreated from a source system or pipeline and the lower long-term storage protection level is acceptable. Snowflake documents that transient tables persist until dropped and are not protected by Fail-safe. See Temporary and transient tables and Understanding storage costs.

QUESTION NO: 36

What versions of Snowflake should be used to manage compliance with Personal Identifiable Information (PII) requirements? (Choose two.)

A. Custom Edition

B. Virtual Private Snowflake

C. Business Critical Edition

D. Standard Edition

E. Enterprise Edition

ANSWER: B C

Explanation:

Business Critical Edition is appropriate for workloads with sensitive data, including personally identifiable information, because it provides Snowflake's highest standard edition level for organizations with stringent security and compliance requirements. It includes the capabilities of lower editions and adds enhanced security-related features and support for more demanding regulated-workload needs. Snowflake positions this edition for customers that require heightened protection and controls when handling sensitive information.

Virtual Private Snowflake is appropriate where PII compliance requirements require an additional level of isolation. Virtual Private Snowflake is an isolated Snowflake environment designed for organizations with particularly rigorous requirements for security, governance, and compliance. It is available in conjunction with Business Critical Edition and provides isolated compute and storage infrastructure rather than shared multi-tenant infrastructure. This additional isolation can help organizations align their Snowflake deployment with strict internal policies or external regulatory obligations for sensitive data. Selection of the appropriate edition or isolated environment should be combined with governance practices such as role-based access control, data classification, masking policies, and auditing. Snowflake describes its edition capabilities at Snowflake Editions and explains the isolated Virtual Private Snowflake offering at Virtual Private Snowflake.

QUESTION NO: 37

Which Snowflake edition offers the highest level of security for organizations that have the strictest requirements?

A. Standard

B. Enterprise

C. Business Critical

D. Virtual Private Snowflake (VPS)

ANSWER: D

Explanation:

Virtual Private Snowflake (VPS) is the correct choice because it is Snowflake's highest-tier offering for customers with exceptionally stringent security, isolation, and regulatory requirements. VPS provides a Snowflake environment that is isolated from other Snowflake accounts, with dedicated virtual server infrastructure. This added isolation is designed for organizations that need to meet the most demanding governance or compliance obligations, including highly regulated enterprises and public-sector workloads. Snowflake positions VPS above its other editions as an option for customers requiring the strongest level of environment isolation and security controls. The offering is available only to eligible organizations and is delivered through a separate Snowflake deployment rather than the shared multi-tenant environment used by typical accounts. Although security is a shared responsibility and customers must still configure identity, access, network, and data-protection controls appropriately, VPS supplies the foundational deployment model intended for the strictest requirements. Snowflake's edition comparison identifies Virtual Private Snowflake as the edition intended for these specialized security and isolation needs. See the Snowflake editions documentation and Snowflake's cloud platforms and regions documentation for deployment context.

QUESTION NO: 38

Which locations can be specified as targets when unloading table data with the COPY INTO <location> command? (Select TWO).

A. An internal stage

B. An external stage

C. A Snowflake table

D. A standard view

E. A stream

ANSWER: A B

Explanation:

An **internal stage** and an **external stage** can both be targets for unloading data with `COPY INTO <location>`. An internal stage stores unloaded files in Snowflake-managed storage. An external stage references a cloud storage location, such as Amazon S3, Google Cloud Storage, or Microsoft Azure storage.

The command writes query or table results as files to the target stage; it does not load results into a Snowflake table or return them directly to a view. The file format, compression, and partitioning options can be configured as part of the unload operation. See the COPY INTO <location> documentation.

QUESTION NO: 39

How can a Data Exchange Administrator provide a user with account access to a Data Exchange?

A. Grant the user the USERADMIN role.

B. Add the user to the Data Exchange.

C. Enable the IMPORT SHARE privilege and grant this privilege to the user.

D. Create a new database for the Data Exchange and provide access to the user.

ANSWER: B

Explanation:

Add the user to the Data Exchange. A Data Exchange Administrator manages participation in an exchange, including granting eligible users access by adding them as members. This membership-based approach gives the user access to the exchange environment, where they can discover and work with the data products available to them according to their assigned exchange role and the permissions associated with the shared data. The administrator performs this management

in the Data Exchange user interface, rather than through standard object-level database grants. This separation is important because participation in an exchange is an exchange-governance activity: administrators control who can enter and use the exchange, while providers separately control which listings, shares, and data products they publish and make available. Snowflake's data-sharing model similarly distinguishes access to shared data from ownership or copying of the underlying provider data, enabling governed distribution to authorized consumers. For Snowflake's overview of secure sharing and the roles involved in publishing and consuming shared data, see Introduction to Secure Data Sharing. For current documentation on collaboration and sharing workflows in Snowflake, see Collaboration in Snowflake.

QUESTION NO: 40

What happens to the objects in a reader account when the DROP MANAGED ACCOUNT command is executed?

A. The objects are dropped.

B. The objects enter the Fail-safe period.

C. The objects enter the Time Travel period.

D. The objects are immediately moved to the provider account.

ANSWER: A

Explanation:

The objects are dropped. A reader account is a type of managed account that a provider creates and administers to share data with consumers that do not have their own Snowflake account. When the provider executes `DROP MANAGED ACCOUNT`, Snowflake drops the managed account itself. Consequently, the objects contained in that reader account, including any databases, schemas, tables, users, roles, and other account-level resources, are removed with the account. The operation is intended to permanently remove the managed account and its contents rather than transfer those contents back to the provider. Providers should therefore ensure that the reader account is no longer needed and retain any required information outside the account before issuing the command. Snowflake documents that dropping a managed account removes the account and all objects in it. The command must be run by an authorized provider-account role, reinforcing that the provider controls the lifecycle of reader accounts. See the Snowflake DROP MANAGED ACCOUNT reference and the documentation on creating and managing reader accounts.

QUESTION NO: 41

What feature of Snowflake Continuous Data Protection can be used for maintenance of historical data?

A. Access control

B. Fail-safe

C. Network policies

D. Time Travel

ANSWER: D

Explanation:

Time Travel is the Snowflake Continuous Data Protection capability used to maintain and work with historical data. It preserves historical versions of data for a configurable retention period, enabling users to query data as it existed at a specific point in time or before a specified statement or timestamp. This makes it possible to recover data after accidental updates, deletes, or drops, and to compare current data with a prior version when investigating changes. For example, a user can use the `AT` or `BEFORE` clause in a query to access a historical version of a table, or can restore dropped objects within the applicable retention period. The retention period is generally configurable from 0 to 90 days for permanent objects, depending on the Snowflake edition and object settings; transient and temporary objects have more limited retention. Time Travel therefore directly supports historical data maintenance by providing self-service access to prior data states. Snowflake documents Time Travel as the mechanism for accessing, querying, cloning, and restoring historical data, while its broader

continuous data protection model also includes Fail-safe for Snowflake-managed disaster recovery after Time Travel expires. See the Snowflake Time Travel documentation and Continuous Data Protection documentation.

QUESTION NO: 42

Which JSON paths are considered to be equivalent in Snowflake? (Choose two.)

A. src[' customer '][' EMAIL ']

B. src[' CUSTOMER '][' Email ']

C. SRC:Customer.Email

D. src:customer.email

E. SRC:customer.email

ANSWER: D E

Explanation:

`src:customer.email` and `SRC:customer.email` are equivalent because Snowflake treats an unquoted column identifier as case-insensitive. Thus, `src` and `SRC` both resolve to the same VARIANT column. The portion after the colon is a semi-structured-data path. In this syntax, the colon identifies the first-level element in the VARIANT value and the period traverses to a nested element, so both expressions navigate from the same source column to the `customer` element and then to its `email` element. The element names in both paths have identical spelling and case, which is essential because JSON object keys are case-sensitive in Snowflake path traversal. Snowflake documentation specifically distinguishes the SQL column identifier, which follows SQL identifier case rules, from the JSON element names used in path notation. Consequently, changing only the capitalization of the unquoted source-column reference does not change the data selected, while the JSON path itself remains the same. See Snowflake's guidance on traversing semi-structured data and its identifier syntax documentation.

QUESTION NO: 43

What activities can a user with the ORGADMIN role perform? (Select TWO).

A. Create an account for an organization.

B. Edit the account data for an organization.

C. Delete the account data for an organization.

D. View usage information for all accounts in an organization.

E. Select all the data in tables for all accounts in an organization.

ANSWER: A D

Explanation:

The `ORGADMIN` role is the organization-level administrator role in Snowflake. It is intended for centralized administration of the Snowflake organization, which can contain multiple Snowflake accounts. **Create an account for an organization.** is correct because ORGADMIN can create accounts within the organization, including specifying account details such as the account name, cloud platform, and region. This enables an organization to provision and manage its Snowflake account footprint centrally.

View usage information for all accounts in an organization. is also correct. ORGADMIN can access organization-level usage views in the shared `SNOWFLAKE` database, including views in the `ORGANIZATION_USAGE` schema. These views provide aggregated usage, cost, and contract-consumption information across the organization's accounts. This supports

organization-wide monitoring, governance, and cost management without requiring the administrator to operate separately within every individual account.

Snowflake documents ORGADMIN as the role that performs organization administration and identifies organization usage data as available to this role. See Organization-level roles and Organization Usage schema.

QUESTION NO: 44

Which Snowflake tasks are managed in the cloud services layer? Select TWO.

A. Performance tuning

B. Query acceleration

C. Metadata management

D. Query processing

E. Query parsing

ANSWER: C E

Explanation:

Metadata management and query parsing are managed by Snowflake's cloud services layer. This layer acts as the coordination and control plane for Snowflake's architecture, supporting the activities that enable users to securely submit, govern, optimize, and manage work before it is executed. Metadata management includes maintaining the information Snowflake needs about database objects and stored data, such as schemas, tables, views, micro-partitions, and related statistics. This centralized metadata is essential to efficient planning, governance, and operation of the platform.

Query parsing occurs when Snowflake interprets submitted SQL and prepares it for execution. The cloud services layer also performs query optimization, determining an efficient execution plan before compute resources carry out the query. Snowflake virtual warehouses provide the compute resources that execute queries, while the cloud services layer handles services such as authentication, access control, transaction management, metadata management, and query parsing and optimization. This separation allows compute to scale independently from the services that coordinate and manage account activity. Snowflake's architecture documentation describes these cloud services responsibilities and the role of virtual warehouses in query execution. See Snowflake architecture and Virtual warehouses.

QUESTION NO: 45

What is the first step when creating a Data Exchange in Snowflake?

A. Create an organization ID.

B. Run the CREATE LISTING command.

C. Run the CREATE SHARE command.

D. Contact Snowflake Support.

ANSWER: D

Explanation:

Contact Snowflake Support. is correct because a Snowflake Data Exchange is a managed, private data-sharing hub rather than an object that an account administrator independently creates with SQL. Snowflake must first provision and configure the Data Exchange for the organization. This initial engagement establishes the exchange and its administrative setup, including the accounts and roles that can manage the exchange, providers, and listings. Once Snowflake has enabled the Data Exchange, its administrators can perform the operational work within the exchange, such as inviting members and managing listings.

This requirement reflects the governance model for a Data Exchange: it is intended to provide a controlled environment where participating accounts can publish and discover data products under the exchange's policies. The required Snowflake involvement ensures the hub is associated with the appropriate organization and configured correctly before any data product publication activities begin. Snowflake's documentation identifies Data Exchanges as private hubs and directs customers to work with Snowflake to establish one. See the Snowflake documentation on Data Exchanges and the guidance for creating a Data Exchange.

QUESTION NO: 46

What actions can be taken to reduce memory spilling for queries that are too large to fit into memory? Select TWO.

A. Increase the size of the virtual warehouse.

B. Edit the queries to process the data in smaller batches.

C. Increase the size of the virtual warehouse clusters.

D. Suspend and then resume the virtual warehouse.

E. Set a clustering key on the tables used in the queries.

ANSWER: A B

Explanation:

Increasing the size of the virtual warehouse is an appropriate action because a larger warehouse provides more compute resources and memory for query execution. When memory-intensive operations such as joins, aggregations, sorts, or window functions have sufficient memory, Snowflake can retain more intermediate results in memory rather than spilling them to local or remote disk. This generally reduces elapsed query time and avoids the additional I/O associated with spilling.

Editing the queries to process the data in smaller batches is also appropriate because it reduces the peak volume of intermediate data that must be held during a single execution. For example, restructuring a large transformation into staged operations or filtering data earlier can lower memory demand. This approach can reduce spills even when resizing the warehouse is not desirable or when the query's execution plan creates very large intermediate result sets.

Snowflake identifies spilled bytes in the Query Profile and recommends considering a larger warehouse when operators spill data. Query design that reduces the amount of data processed and intermediate result size complements warehouse scaling as a practical performance-tuning measure. See Snowflake's Query Profile documentation and warehouse considerations for query performance.

QUESTION NO: 47

Which privilege is required to use the search optimization service in Snowflake?

A. GRANT SEARCH OPTIMIZATION ON SCHEMA TO ROLE

B. GRANT SEARCH OPTIMIZATION ON DATABASE TO ROLE

C. GRANT ADD SEARCH OPTIMIZATION ON SCHEMA TO ROLE

D. GRANT ADD SEARCH OPTIMIZATION ON DATABASE TO ROLE

ANSWER: C

Explanation:

`GRANT ADD SEARCH OPTIMIZATION ON SCHEMA <schema_name> TO ROLE <role>` is correct because `ADD SEARCH OPTIMIZATION` is the Snowflake schema-level privilege that authorizes a role to enable the Search Optimization Service for eligible tables in that schema. A user operating through that role can then use `ALTER TABLE ... ADD`

`SEARCH OPTIMIZATION` to add a search optimization configuration to a table. This privilege is specifically intended to delegate control of adding search optimization without requiring ownership of every table in the schema. In practice, the role also needs the applicable access privileges to work with the database, schema, and table, but the privilege that directly permits adding the search optimization configuration is `ADD SEARCH OPTIMIZATION`. Snowflake documents this privilege under its access-control privilege reference and lists it as a schema privilege. The Search Optimization Service documentation also identifies the required authorization for adding search optimization to a table. See Snowflake's Access Control Privileges reference and the Search Optimization Service documentation.

QUESTION NO: 48

Which native data types are used for storing semi-structured data in Snowflake? (Select TWO)

A. NUMBER

B. OBJECT

C. STRING

D. VARCHAR

E. VARIANT

ANSWER: B E

Explanation:

Snowflake's native semi-structured data support includes **OBJECT** and **VARIANT**. VARIANT is a tagged universal type designed to hold values from semi-structured formats such as JSON, Avro, ORC, Parquet, and XML. A VARIANT value can contain scalar values, arrays, and object-like structures while preserving the structure needed for later querying. This makes it the common target type when loading semi-structured source files into Snowflake and when querying nested paths with Snowflake's semi-structured-data syntax.

OBJECT is Snowflake's native type for an unordered collection of key-value pairs, analogous to a JSON object. It is useful when the stored value is specifically an object structure and can hold values of various Snowflake-supported types, including nested semi-structured content. Both types are therefore purpose-built for representing structured, nested, and flexible-schema data within Snowflake tables. Snowflake documents VARIANT, OBJECT, and ARRAY as its semi-structured data types, with OBJECT representing key-value collections and VARIANT representing values of arbitrary supported types. See the Snowflake semi-structured data types reference and the introduction to querying semi-structured data.

QUESTION NO: 49

At what level is the MIN_DATA_RETENTION_TIME_IN_DAYS parameter set?

A. Account

B. Database

C. Schema

D. Table

ANSWER: A

Explanation:

The correct answer is **Account**. `MIN_DATA_RETENTION_TIME_IN_DAYS` is an account-level parameter that establishes a floor for Time Travel data retention throughout the account. Its value specifies the minimum number of days for which historical data must remain available for eligible objects, helping an organization enforce a baseline recovery and data-governance policy. Although individual databases, schemas, and tables can have their own `DATA_RETENTION_TIME_IN_DAYS` settings, those object-level settings cannot reduce retention below the account-wide

minimum defined by `MIN_DATA_RETENTION_TIME_IN_DAYS`. This makes the account the appropriate scope: it centrally controls the minimum retention safeguard while still allowing more granular object settings to retain data for longer where business or compliance requirements demand it. The parameter is managed with an account-level `ALTER ACCOUNT SET` statement and is listed by Snowflake among parameters that can be configured for an account. Snowflake documents the parameter and its relationship to Time Travel retention in its parameter reference and explains Time Travel retention behavior in the Time Travel documentation.

QUESTION NO: 50

What is the recommended file sizing for data loading using Snowpipe?

A. A compressed file size greater than 100 MB, and up to 250 MB

B. A compressed file size greater than 100 GB, and up to 250 GB

C. A compressed file size greater than 10 MB, and up to 100 MB

D. A compressed file size greater than 1 GB, and up to 2 GB

ANSWER: A

Explanation:

A compressed file size greater than 100 MB, and up to 250 MB is the recommended sizing range for loading data with Snowpipe. Snowflake advises staging files in this range because it provides an effective balance between load parallelism and per-file processing overhead. Files that are too small can cause unnecessary overhead, since Snowpipe must track and process many individual load operations. Appropriately sized compressed files allow the Snowpipe service to process incoming data efficiently while avoiding the reduced parallelism associated with very large individual files. This guidance applies to the compressed size of the staged data files, which is particularly important for commonly used compressed formats such as gzip. File sizing should be considered alongside a continuous ingestion design: applications should write data to the stage in regularly produced files, and Snowpipe will load them as they become available. Snowflake documents the 100–250 MB compressed-file recommendation specifically in its Snowpipe data-loading guidance. See the Snowpipe introduction documentation and Snowflake's data loading considerations for the file-size and staging recommendations.

QUESTION NO: 51

What are the benefits of the replication feature in Snowflake? (Select TWO).

A. Disaster recovery

B. Time Travel

C. Fail-safe

D. Database failover and fallback

E. Data security

ANSWER: A D

Explanation:

Disaster recovery is a core benefit of Snowflake replication. Replication creates and refreshes a secondary copy of supported objects in another Snowflake account, typically in a separate region or cloud platform. Maintaining this geographically separate copy enables an organization to prepare for a regional service disruption and restore business operations from the secondary environment. Snowflake's replication and failover capabilities are specifically intended to support business-continuity and disaster-recovery architectures.

Database failover and fallback is also a benefit because replicated databases can be promoted when the primary location is unavailable. For more comprehensive designs, replication or failover groups can contain related account objects and data,

allowing coordinated failover to a target account. After the original primary is available again, Snowflake supports returning operations to the original location through failback workflows, after synchronizing changes as required. This lets organizations establish an operational recovery process rather than merely retaining a separate backup copy. Snowflake documents replication, failover, and failback as related mechanisms for protecting and recovering replicated data and account objects. See the Snowflake replication and failover overview and database replication documentation.

QUESTION NO: 52

Which operations are handled in the Cloud Services layer of Snowflake? (Select TWO).

A. Security

B. Data storage

C. Data visualization

D. Query computation

E. Metadata management

ANSWER: A E

Explanation:

Security and **Metadata management** are handled by Snowflake's Cloud Services layer. This layer is the collection of services that coordinates and manages activities across the Snowflake platform rather than performing the underlying storage or compute work. Its security responsibilities include authenticating users and enforcing access control decisions, allowing Snowflake to apply account, role, object, and policy-based permissions before operations are performed.

The Cloud Services layer also manages metadata, which is the information Snowflake maintains about database objects, schemas, tables, views, micro-partitions, users, roles, and other platform resources. Metadata management supports essential query-lifecycle services such as parsing SQL, creating and optimizing query plans, and coordinating query execution. These centrally managed services enable users to interact with Snowflake through SQL while Snowflake handles object definitions, privileges, and query coordination transparently. Snowflake documents the Cloud Services layer as responsible for services including authentication, access control, metadata management, query parsing, and query optimization. See the Snowflake key concepts and architecture documentation and the Snowflake architecture overview.

QUESTION NO: 53

What function can be used with the recursive argument to return a list of distinct key names in all nested elements in an object?

A. FLATTEN

B. GET_PATH

C. CHECK_JSON

D. PARSE JSON

ANSWER: A

Explanation:

`FLATTEN` is the correct function because it expands semi-structured values such as OBJECT and ARRAY data into a set of rows that can be queried using SQL. Its `RECURSIVE` parameter can be set to `TRUE`, causing Snowflake to traverse nested objects and arrays rather than processing only the outermost level. For each expanded element, FLATTEN returns metadata columns including `KEY`, which contains the property name for OBJECT elements. A query can therefore select the `KEY` column from a recursive FLATTEN operation and apply `DISTINCT` to produce a unique list of key names encountered throughout the nested object structure. For example, a lateral FLATTEN call using `INPUT => object_column` and

`RECURSIVE => TRUE`, followed by `SELECT DISTINCT f.KEY`, provides this result. This is a common Snowflake technique for inspecting unknown or evolving JSON-like schemas and identifying fields available at any nesting depth. Snowflake documents the output columns and recursive traversal behavior in its FLATTEN function reference; see also the guidance for querying semi-structured data.

QUESTION NO: 54

Which statements are correct concerning the leveraging of third-party data from the Snowflake Data Marketplace? (Choose two.)

A. Data is live, ready-to-query, and can be personalized.

B. Data needs to be loaded into a cloud provider as a consumer account.

C. Data is not available for copying or moving to an individual Snowflake account.

D. Data is available without copying or moving.

E. Data transformations are required when combining Data Marketplace datasets with existing data in Snowflake.

ANSWER: A D

Explanation:

Data Marketplace listings are shared through Snowflake's secure data-sharing capabilities. When a consumer obtains a listing, the provider grants access to a shared database in the consumer's Snowflake account. This makes the data live and ready to query: queries access the provider-managed data without requiring the consumer to first ingest and maintain a separate physical copy. The provider can continue updating the shared data, allowing consumers to work with the current published dataset. Marketplace data can also be used in the consumer's own Snowflake environment, including in queries, views, and analytics alongside data the consumer already holds, supporting use cases tailored to that consumer's needs. Therefore, "Data is live, ready-to-query, and can be personalized." correctly describes the consumption model. "Data is available without copying or moving." is also correct because Snowflake Secure Data Sharing exposes data through metadata and access controls rather than traditional extract, transfer, and load processes. This architecture reduces data-pipeline overhead while maintaining governance through the provider's sharing configuration. See Snowflake's introduction to Secure Data Sharing and its Marketplace documentation for details on accessing listings and querying shared data.

QUESTION NO: 55

When connecting to Snowflake using SnowSQL, what are ways to explicitly specify the password? (Select TWO).

A. Use public and private key pair authentication

B. Run a web-based authorization flow

C. Use an OAuth token

D. Enter through an interactive prompt

E. Specify using SNOWSQL_PWD environment variables

ANSWER: D E

Explanation:

SnowSQL supports password authentication either by obtaining the password interactively or by receiving it from the `SNOWSQL_PWD` environment variable. With an interactive prompt, a user starts SnowSQL without supplying a password value and securely types the password when SnowSQL requests it. This avoids placing the secret directly in a command-line argument, where it could be exposed in shell history or process listings. The `SNOWSQL_PWD` environment variable provides a non-interactive mechanism for supplying the password, which can be useful for controlled automation and scripts. Snowflake documents this variable specifically as the environment-variable equivalent for the SnowSQL password connection

parameter. In either case, SnowSQL uses the supplied password together with the configured or specified account and user details to establish a password-authenticated session. For operational security, Snowflake recommends protecting credentials and avoiding exposure in scripts or logs; environment variables should therefore be handled carefully and scoped appropriately in automated environments. See Snowflake's SnowSQL configuration documentation and its connection and authentication guidance.

QUESTION NO: 56

The following settings are configured:

THE MIN_DATA_RETENTION_TIME_IN_DAYS is set to 5 at the account level.

THE DATA_RETENTION_TIME_IN_DAYS is set to 2 at the object level.

For how many days will the data be retained at the object level?

A. 2

B. 3

C. 5

D. 7

ANSWER: C

Explanation:

5 is correct. `MIN_DATA_RETENTION_TIME_IN_DAYS` is an account-level parameter that establishes a minimum Time Travel data-retention period for applicable objects in the account. Snowflake applies this configured minimum even when an object's own `DATA_RETENTION_TIME_IN_DAYS` setting is lower. Therefore, although the object explicitly specifies 2 days, the account's minimum of 5 days governs the effective retention period, and the object's historical data is retained for 5 days.

This setting is useful when an organization needs a baseline recovery and audit window that cannot be shortened by object-level configuration. The object-level parameter can request a retention period, but it cannot reduce the period below the account-wide minimum. In this scenario, Snowflake effectively uses the greater applicable value: 5 days from the account minimum rather than 2 days from the object setting. This retention period supports Time Travel operations, such as querying, cloning, or restoring data from within the retained historical window. See Snowflake's documentation for the MIN_DATA_RETENTION_TIME_IN_DAYS parameter and its Time Travel data retention documentation.

QUESTION NO: 57

What transformations are supported in a CREATE PIPE ... AS COPY ... FROM (....) statement? (Select TWO.)

A. Data can be filtered by an optional where clause

B. Incoming data can be joined with other tables

C. Columns can be reordered

D. Columns can be omitted

E. Row level access can be defined

ANSWER: C D

Explanation:

"Columns can be reordered" and "Columns can be omitted" are supported because a Snowpipe definition can use a `SELECT` statement in the `COPY INTO` command's `FROM` clause to transform staged data as it is loaded. The select list explicitly controls both which source fields are included and the order in which their resulting values are supplied to the target table.

For example, the select list can reference staged-file fields in a sequence that differs from their source order, and it can leave unwanted source fields out entirely. This capability is useful when files contain extra attributes, when a target table uses a different column arrangement, or when semi-structured input must be projected into a defined set of target columns. Snowflake documents this as using a select statement to transform data during a Snowpipe load; the supported transformation model is deliberately limited to operations on the rows read from the staged files. The pipe's copy statement therefore provides controlled column projection and ordering while preserving Snowpipe's continuous ingestion design. See Snowflake's Transforming data during a load guidance and the CREATE PIPE reference for the supported pipe definition syntax.

QUESTION NO: 58

If 3 size Small virtual warehouse is made up of two servers, how many servers make up a Large warehouse?

A. 4

B. 8

C. 16

D. 32

ANSWER: B

Explanation:

8 is correct. Snowflake virtual warehouse sizes increase in a doubling progression: an X-Small warehouse represents the baseline compute capacity, a Small warehouse has twice that capacity, a Medium warehouse has twice the capacity of Small, and a Large warehouse has twice the capacity of Medium. Therefore, moving from Small to Large involves two size increases: Small to Medium doubles the two-server premise to four servers, and Medium to Large doubles it again to eight servers.

This same sizing progression is reflected in Snowflake's warehouse credit-consumption rates. A Small warehouse consumes 2 credits per hour and a Large warehouse consumes 8 credits per hour, so a Large warehouse provides four times the compute resources assumed for a Small warehouse. Starting with two servers for Small, four times that capacity is eight servers. Warehouse sizing is designed to let administrators increase available compute for more concurrent work or faster processing while retaining independent storage and compute architecture. See Snowflake's Virtual warehouses overview and Understanding compute cost documentation.

QUESTION NO: 59

A user needs to create a materialized view in the schema MYDB.MYSCHEMA.

Which statements will provide this access?

A. GRANT ROLE MYROLE TO USER USER1;
CREATE MATERIALIZED VIEW ON SCHEMA MYDB.MYSCHEMA TO ROLE MYROLE;

B. GRANT ROLE MYROLE TO USER USER1;
CREATE MATERIALIZED VIEW ON SCHEMA MYDB.MYSCHEMA TO USER USER1;

C. GRANT ROLE MYROLE TO USER USER1;
CREATE MATERIALIZED VIEW ON SCHEMA MYDB.MYSCHEMA TO USER1;

D. GRANT ROLE MYROLE TO USER USER1;
CREATE MATERIALIZED VIEW ON SCHEMA MYDB.MYSCHEMA TO MYROLE;

E. GRANT ROLE MYROLE TO USER USER1;
GRANT CREATE MATERIALIZED VIEW ON SCHEMA MYDB.MYSCHEMA TO ROLE MYROLE;

ANSWER: E

Explanation:

`GRANT ROLE MYROLE TO USER USER1;`
`GRANT CREATE MATERIALIZED VIEW ON SCHEMA MYDB.MYSCHEMA TO ROLE MYROLE;` is the correct privilege-grant pattern. In Snowflake's RBAC model, privileges on securable objects, including schemas, are granted to roles. A user receives those privileges by being granted the role and using that role as their active role. The `CREATE MATERIALIZED VIEW` schema privilege specifically authorizes the role to create materialized views in the named schema. The grant must use the `GRANT` command and must explicitly identify the grantee as a role with `TO ROLE MYROLE`.

For a user to successfully run a complete `CREATE MATERIALIZED VIEW` statement in practice, the active role also needs the applicable `USAGE` privileges on the database and schema, plus `SELECT` access to the source table or view used in the materialized-view definition. Those additional requirements depend on the existing role configuration and the selected source object; the statement shown grants the specific creation privilege requested. Snowflake documents schema-level creation privileges in its schema privileges reference and describes the role-based grant syntax in GRANT <privileges> documentation.

QUESTION NO: 60

Credit charges for Snowflake virtual warehouses are calculated based on which of the following considerations? (Choose two.)

A. The number of queries executed

B. The number of active users assigned to the warehouse

C. The size of the virtual warehouse

D. The length of time the warehouse is running

E. The duration of the queries that are executed

ANSWER: C D

Explanation:

Snowflake computes virtual warehouse charges from the warehouse's configured size and the time that its compute resources are running. The size of the virtual warehouse determines its credit consumption rate: larger warehouses use more credits per hour than smaller ones because they provision more compute resources. Snowflake publishes the credit rate associated with each warehouse size, enabling organizations to estimate and control the cost of a selected configuration.

The length of time the warehouse is running is the other direct charging consideration. While running, a warehouse consumes credits on a per-second basis, subject to a 60-second minimum each time it starts or resumes. Consequently, auto-suspend settings are important for cost management because suspending an idle warehouse stops its consumption of warehouse credits. A query can cause a suspended warehouse to resume, but Snowflake does not calculate warehouse credits as a separate charge per query or according to the individual query's execution duration. Instead, those activities affect costs only insofar as they require the warehouse to remain running. See Snowflake's virtual warehouse overview and understanding compute cost documentation.

QUESTION NO: 61

Which feature is designed to reduce security risks associated with an individual user ' s password authentication?

A. Multi-Factor Authentication, or MFA

B. Standard authentication

C. Snowflake OAuth

D. External OAuth

ANSWER: A

Explanation:

Multi-Factor Authentication, or MFA is designed to reduce the security risks inherent in password-based authentication by requiring users to prove their identity with an additional verification factor when signing in. A password is something a user knows; it can be guessed, reused, phished, stolen, or exposed in a breach. MFA adds a separate factor, typically a time-based one-time passcode generated by an authenticator application, so possession of the password alone is insufficient to access Snowflake. This materially strengthens protection for user accounts, especially privileged accounts, by reducing the likelihood that compromised credentials can be used successfully. In Snowflake, MFA can be enrolled by individual users and administrators can require it through authentication policies, enabling organizations to apply the control consistently according to their security requirements. Snowflake documents MFA as an added layer of security for user sign-ins and supports authenticator applications that generate one-time passcodes. See the Snowflake Multi-Factor Authentication documentation and the Snowflake authentication policies documentation.

QUESTION NO: 62

What is a responsibility of Snowflake's virtual warehouses?

A. Infrastructure management

B. Metadata management

C. Query execution

D. Query parsing and optimization

E. Permanent storage of micro-partitions

ANSWER: C

Explanation:

Query execution is a responsibility of Snowflake virtual warehouses. A virtual warehouse is a cluster of compute resources that customers provision and size independently from storage. When a user submits SQL, Snowflake uses warehouse compute to perform the processing work required to run the query, including reading the relevant data, executing joins, aggregations, sorts, and other operations, and returning results. Warehouses can also execute data-loading and unloading operations, so their role is the scalable compute layer of Snowflake's architecture. Because compute is separated from storage, a warehouse can be started, stopped, resized, or scaled out with multi-cluster capability according to workload demand, without moving the underlying table data. This separation enables independent management of compute capacity and persistent data. Snowflake documents virtual warehouses as the compute resources used to execute queries and other data-manipulation tasks. See the Virtual warehouses overview and Snowflake's key concepts documentation for details on the platform's storage, compute, and cloud-services layers.

QUESTION NO: 63

Which items are considered schema objects in Snowflake? (Select TWO).

A. Pipe

B. File format

C. Resource monitor

D. Storage integration

E. Virtual warehouse

ANSWER: A B

Explanation:

Pipe and **File format** are schema-level objects in Snowflake. A pipe stores the COPY INTO statement and related metadata that Snowpipe uses to continuously ingest data from a stage into a table. Because a pipe is created within, and belongs to, a specific schema, it is addressed using a database.schema.object name and is managed through schema object privileges and ownership.

A file format is likewise a schema object. It defines how Snowflake should interpret staged data files, including attributes such as file type, delimiters, compression, date and time formats, and parsing behavior. Named file formats can be created in a schema and then reused by loading and unloading commands, which provides consistency across ingestion workflows. Snowflake's SQL reference explicitly identifies both object types as objects created in a schema. See the documentation for CREATE PIPE and CREATE FILE FORMAT.

QUESTION NO: 64

What are characteristic of Snowsight worksheet? (Select TWO.)

A. Worksheets can be grouped under folder, and a folder of folders.

B. Each worksheet is a unique Snowflake session.

C. Users are limited to running only one on a worksheet.

D. The Snowflake session ends when a user switches worksheets.

E. Users can import worksheets and share them with other users.

ANSWER: A E

Explanation:

"Worksheets can be grouped under folder, and a folder of folders." is correct because Snowsight provides worksheet organization through folders, including nested folders. This lets users arrange related SQL work, analyses, and administrative queries in a hierarchy rather than maintaining an unstructured list of worksheets. Folder-based organization is especially useful for separating work by project, team, environment, or subject area while retaining worksheets in Snowsight.

"Users can import worksheets and share them with other users." is also correct. Snowsight supports importing worksheet content, helping users bring existing SQL work into the interface. It also provides worksheet-sharing functionality so that a worksheet can be made available to other Snowflake users. Sharing enables collaboration on saved SQL and associated worksheet context without requiring collaborators to manually recreate the work. Access and the ability to execute statements remain subject to each user's Snowflake privileges, role, and available compute resources.

These capabilities are part of Snowsight's worksheet management and collaboration experience. See Snowflake's documentation for worksheets in Snowsight and sharing worksheets.

QUESTION NO: 65

Which service or feature in Snowflake is used to improve the performance of certain types of lookup and analytical queries that use an extensive set of WHERE conditions?

A. Data classification

B. Query acceleration service

C. Search optimization service

D. Tagging

ANSWER: C

Explanation:

The correct answer is **Search optimization service**. Snowflake's Search Optimization Service is specifically designed to speed up selective lookup queries and analytical queries where filters in `WHERE` clauses narrow a large table to a relatively small set of rows. It works by building and maintaining a search access path for enabled tables, allowing Snowflake to avoid scanning unnecessary micro-partitions for supported query patterns. This is especially valuable for point lookups, queries using equality or `IN` predicates, and other highly selective filtering conditions on large datasets. The service can also optimize supported joins, substring searches, and certain semi-structured-data access patterns, but its core value is reducing the work required to locate targeted data. It is enabled at the table level using `ADD SEARCH OPTIMIZATION`, and Snowflake automatically maintains the associated search access paths as table data changes. Because it consumes serverless compute resources for maintenance and query processing, it should be applied to tables and columns with workloads that benefit from selective searches. Snowflake documents the supported query types, configuration, and cost considerations in its Search Optimization Service documentation and its overview and usage guidance.

QUESTION NO: 66

Which of the following are benefits of micro-partitioning? (Select TWO)

A. Micro-partitions cannot overlap in their range of values

B. Micro-partitions are immutable objects that support the use of Time Travel.

C. Micro-partitions can reduce the amount of I/O from object storage to virtual warehouses

D. Rows are automatically stored in sorted order within micro-partitions

E. Micro-partitions can be defined on a schema-by-schema basis

ANSWER: B C

Explanation:

Micro-partitions are immutable objects that support the use of Time Travel. Snowflake automatically organizes table data into immutable micro-partitions. When data is changed, Snowflake creates new micro-partitions rather than modifying existing ones. This preservation of prior partition versions is fundamental to Snowflake's Time Travel capability, allowing eligible historical data states to be queried or restored during the configured retention period.

Micro-partitions can reduce the amount of I/O from object storage to virtual warehouses because Snowflake records metadata for each micro-partition, including value ranges and other statistics for columns. During query execution, the optimizer uses this metadata to identify and eliminate micro-partitions that cannot contain qualifying rows. This micro-partition pruning means the virtual warehouse reads fewer relevant data files from storage, reducing scan volume and improving query performance. These behaviors occur automatically; users do not need to manually create or maintain partitions to receive the benefits. See Snowflake's documentation on micro-partitions and data clustering and Time Travel.

QUESTION NO: 67

Which data types can be used in a Snowflake table that holds semi-structured data? (Select TWO).

A. ARRAY

B. BINARY

C. TEXT

D. VARIANT

E. VARCHAK

ANSWER: A D

Explanation:

ARRAY and **VARIANT** are Snowflake semi-structured data types. VARIANT is Snowflake's universal type for storing values with differing structures and native semi-structured content, including data loaded from formats such as JSON, Avro, ORC, Parquet, and XML. It preserves the hierarchical structure of the source data and allows Snowflake SQL to access nested elements using path notation and functions.

ARRAY stores an ordered collection of values in one column. It is appropriate when a record contains a list of related items, such as tags, line items, or event attributes. Arrays can contain values of different compatible types, including nested semi-structured values, and Snowflake provides functions such as FLATTEN to transform array elements into rows when analysis requires a tabular result. A table may combine these types with conventional relational columns, allowing semi-structured data to be ingested and queried without requiring a fixed relational schema before loading. Snowflake documents ARRAY, OBJECT, and VARIANT as its semi-structured data types; the two requested types present here are ARRAY and VARIANT. See Snowflake's semi-structured data type documentation and its FLATTEN function reference.

QUESTION NO: 68

What Snowflake features allow virtual warehouses to handle high concurrency workloads? (Select TWO)

A. The ability to scale up warehouses

B. The use of warehouse auto scaling

C. The ability to resize warehouses

D. Use of multi-clustered warehouses

E. The use of warehouse indexing

ANSWER: B D

Explanation:

The use of warehouse auto scaling and Use of multi-clustered warehouses enable Snowflake to accommodate high query concurrency by providing additional, independent compute clusters for a single virtual warehouse. A multi-cluster warehouse consists of multiple clusters with the same warehouse configuration. Snowflake can direct concurrent queries across those clusters, reducing queueing when one cluster does not have sufficient capacity to immediately execute all incoming work. This lets the warehouse preserve workload isolation at the compute level while using the same underlying database objects and warehouse settings.

Auto scaling is the dynamic behavior available for a multi-cluster warehouse. When demand rises, Snowflake can start additional clusters up to the configured maximum cluster count; as demand falls, it can suspend unneeded clusters down to the configured minimum. This approach is designed for variable or unpredictable concurrency and helps balance responsiveness with cost, since extra clusters run only when required by the workload. Administrators configure this through the warehouse's multi-cluster settings, including minimum and maximum cluster counts and the scaling policy. Snowflake documents multi-cluster warehouses specifically as the mechanism for managing large numbers of concurrent users and queries. See Multi-cluster warehouses and Virtual warehouses.

QUESTION NO: 69

What are the recommended alternative data types in Snowflake for unsupported large object data types such as BLOB and CLOB? (Select TWO).

A. VARIANT

B. ARRAY

C. BINARY

D. OBJECT

E. VARCHAR

ANSWER: C E

Explanation:

`BINARY` and `VARCHAR` are the appropriate Snowflake replacements for the common large-object patterns represented by BLOB and CLOB. Use `BINARY` for BLOB-style data: it stores arbitrary bytes, including non-text content such as documents, images, encrypted payloads, or other binary streams. Snowflake supports specifying a maximum byte length for a `BINARY` column, subject to its platform limits, while retaining the bytes without interpreting them as character data.

Use `VARCHAR` for CLOB-style data because a CLOB is character data and Snowflake's string types are designed to store variable-length text. `VARCHAR` can store large Unicode text values and is the standard target type when migrating character large objects to Snowflake. The practical migration rule is therefore based on the content rather than the legacy LOB label: binary content belongs in `BINARY`, and textual content belongs in `VARCHAR`. Snowflake documents `BINARY` as its binary-data type and documents `VARCHAR` among its string/text data types. See the Snowflake references for binary data types and text data types.

QUESTION NO: 70

A sales table FCT_SALES has 100 million records.

The following Query was executed

SELECT COUNT (1) FROM FCT__SALES;

How did Snowflake fulfill this query?

A. Query against the result set cache

B. Query against a virtual warehouse cache

C. Query against the most-recently created micro-partition

D. Query against the metadata cache

ANSWER: D

Explanation:

Query against the metadata cache is correct because Snowflake can optimize an unfiltered count of all rows by using table and micro-partition metadata rather than reading the table's stored column data. Snowflake automatically records metadata for every micro-partition, including the number of rows it contains. For a query such as `COUNT(1)`, where the expression is a non-NULL constant and there is no predicate requiring row-level evaluation, the optimizer can aggregate these recorded row counts to produce the result efficiently. This avoids scanning the 100 million rows in the sales table and greatly reduces the processing required. In practical terms, the optimization is based on metadata maintained by Snowflake's cloud services and storage layers, not on a previously saved result or data cached on a virtual warehouse. Snowflake documents that it automatically collects and uses micro-partition metadata for query optimization, and its `COUNT` function documentation describes metadata-based optimizations for applicable count operations. See Micro-partitions and data clustering and COUNT function.

QUESTION NO: 71

What happens to historical data when the retention period for an object ends?

A. The data is cloned into a historical object.

B. The data moves to Fail-safe

C. Time Travel on the historical data is dropped.

D. The object containing the historical data is dropped.

ANSWER: B

Explanation:

The data moves to Fail-safe. For a permanent Snowflake object, data that is retained for Time Travel remains available for historical querying, cloning, and recovery operations during the object's configured data-retention period. When that Time Travel retention period expires, the historical data automatically enters the Fail-safe period. Fail-safe is a Snowflake-managed, seven-day recovery window intended only for emergency data-recovery situations; it is not a user-accessible extension of Time Travel. During this period, Snowflake may be able to recover the data on the customer's behalf, subject to the Fail-safe process. After the Fail-safe period concludes, the historical data is permanently purged. This lifecycle is central to Snowflake's data-protection model for permanent tables and other supported permanent objects. The configured retention period determines how long users can directly access historical versions through Time Travel, while Fail-safe provides an additional platform-managed safeguard after that direct-access window has ended. See Snowflake's Time Travel documentation and storage and Fail-safe documentation for the retention lifecycle details.

QUESTION NO: 72

What is the MINIMUM Snowflake edition that supports the periodic rekeying of encrypted data?

A. Standard

B. Enterprise

C. Business Critical

D. Virtual Private Snowflake (VPS)

ANSWER: B

Explanation:

Enterprise is the minimum Snowflake edition that includes periodic rekeying of encrypted data. Snowflake encrypts customer data by default using a hierarchical key model: data is encrypted with encryption keys that are themselves protected by higher-level keys. Periodic rekeying is an enhanced data-protection capability in which Snowflake rotates the relevant encryption keys on a defined schedule, strengthening long-term cryptographic hygiene without requiring customers to decrypt, export, or reload their data. This feature is included in the Enterprise edition and therefore is also available in higher editions, but Enterprise is the lowest qualifying edition. The edition comparison in Snowflake documentation lists periodic rekeying of encrypted data under Enterprise Edition features. Snowflake's encryption documentation also describes its always-on encryption architecture and centralized management of encryption keys, which enables Snowflake to perform key rotation transparently while maintaining access to protected data. See the Snowflake editions and features documentation and the Snowflake encryption documentation.

QUESTION NO: 73

Which models are supported for Snowflake access control? (Select TWO)

A. UAC (User Access Control)

B. RBAC (Role-Based Access Control)

C. DAC (Discretionary Access Control)

D. OAC (Object Access Control)

E. EBAC (Entity-Based Access Control)

ANSWER: B C

Explanation:

Snowflake supports both **RBAC (Role-Based Access Control)** and **DAC (Discretionary Access Control)** as parts of its access-control framework. RBAC is Snowflake's primary authorization model: privileges on securable objects are granted to roles, and roles are then granted to users or to other roles in a role hierarchy. This approach centralizes administration, supports separation of duties, and lets administrators manage access consistently as users' responsibilities change. Snowflake provides both account roles and database roles to organize privileges at appropriate scopes.

DAC is also supported because object owners, or roles holding the necessary grant authority, can grant privileges on objects to other roles. In particular, a privilege granted with `WITH GRANT OPTION` allows the receiving role to grant that privilege onward to other roles. This discretionary delegation capability is the DAC component of Snowflake access control. Together, RBAC provides the structured role-and-privilege model used for routine governance, while DAC enables authorized privilege delegation when required. Snowflake documents these models and their interaction in its access-control overview and privilege-grant documentation: Access control overview and GRANT <privileges>.

QUESTION NO: 74

Which services does the Snowflake Cloud Services layer manage? (Select TWO).

A. Compute resources

B. Query execution

C. Authentication

D. Data storage

E. Metadata

ANSWER: C E

Explanation:

Authentication and **Metadata** are managed by Snowflake's Cloud Services layer. This layer is the collection of services that coordinates activities across Snowflake rather than supplying the warehouse compute that processes query workloads or the storage layer that persists table data. It handles authentication and access-control-related activities, allowing Snowflake to verify user or client identities before permitting access to account resources. Authentication can use Snowflake-managed credentials as well as supported federated and key-pair-based approaches, but the Cloud Services layer remains responsible for the service-side authentication capability.

The Cloud Services layer also manages metadata: the information Snowflake needs to describe and operate database objects and account resources, such as databases, schemas, tables, views, users, roles, and related object definitions. Centralized metadata enables Snowflake to coordinate object access and query-related operations independently from the data stored in cloud storage. Snowflake documentation describes Cloud Services as the layer supporting functions including authentication, access control, metadata management, query parsing and optimization, and infrastructure management. See Snowflake's key concepts and architecture overview and its authentication documentation.

QUESTION NO: 75

What are valid sub-clauses to the OVER clause for a window function? (Select TWO).

A. GROUP BY

B. LIMIT

C. ORDER BY

D. PARTITION BY

E. UNION ALL

ANSWER: C D

Explanation:

ORDER BY and **PARTITION BY** are valid sub-clauses within a window function's `OVER` clause in Snowflake. `PARTITION BY` divides the query result into independent groups, or partitions, and the window function is evaluated separately for rows in each group. For example, a running total can be restarted for every customer, department, or region by partitioning on that attribute. `ORDER BY` establishes the sequence of rows within each partition for calculations whose outcome depends on row position, including ranking functions, running aggregates, and navigation functions such as `LAG` and `LEAD`. When an ordered window aggregate is used, Snowflake also applies an appropriate window frame unless one is explicitly supplied. Together, these clauses define both the population of rows considered by a window calculation and, where needed, the order in which that calculation proceeds. Snowflake documents the general syntax as `OVER ( [ PARTITION BY ... ] [ ORDER BY ... ] [ windowFrameClause ] )`. See the Snowflake documentation for analytic/window functions and the OVER clause.

QUESTION NO: 76

Which user object property requires contacting Snowflake Support in order to set a value for it?

A. DISABLED

B. MINS TO BYPASS MFA

C. MINS TO BYPASS NETWORK POLICY

D. MINS TO UNLOCK

ANSWER: B

Explanation:

`MINS TO BYPASS MFA` is correct because the corresponding Snowflake user-object property, `MINS_TO_BYPASS_MFA`, has a special administrative restriction: its value can be set only by contacting Snowflake Support. The property controls a temporary MFA bypass period, measured in minutes, after a user has successfully completed MFA authentication. It is intended for narrowly controlled situations in which repeatedly prompting for MFA during a short period would be inappropriate, while still preserving Snowflake's MFA security controls outside that period. Because changing this behavior affects the authentication posture of an account, Snowflake does not make it a normal self-service setting through standard `CREATE USER` or `ALTER USER` administration. Account administrators should treat any request to configure this value as a security-sensitive exception and coordinate with Snowflake Support, following their organization's access-control and security-governance process. Snowflake documents this support requirement directly in the user properties reference: CREATE USER optional parameters. Additional context on configuring and administering multi-factor authentication is available in Snowflake's MFA documentation.

QUESTION NO: 77

The Query Profile shows the metrics of a query that ran on a size Large virtual warehouse.

What steps will improve the query performance? (Select TWO).

A. Increase the MAX_CLUSTER_COUNT.

B. Increase the WAREHOUSE_SIZE.

C. Update the query WHERE clause.

D. Increase the warehouse AUTO_SUSPEND duration.

E. Run the warehouse in Maximized mode.

ANSWER: B C

Explanation:

Increasing the `WAREHOUSE_SIZE` can improve the elapsed time of a single resource-intensive query because a larger warehouse provides more compute resources, including additional CPU and memory, for execution. This is particularly useful when the Query Profile indicates that query operators are compute-bound, processing large data volumes, or spilling data to remote storage because of insufficient memory. Snowflake documents that resizing a warehouse increases the resources available for queries running on it.

Updating the query `WHERE` clause can improve performance by limiting the number of rows and micro-partitions that Snowflake must scan and process. A selective predicate, especially one that aligns with the table's natural clustering or clustering key, enables micro-partition pruning. Reducing scanned data lowers work across downstream operators such as joins, aggregations, and sorts, and is often an effective optimization regardless of warehouse size. Query Profile is intended to help identify these scan and operator costs so that SQL can be refined appropriately.

See Snowflake guidance on virtual warehouse sizing and micro-partitions and pruning.

QUESTION NO: 78

Which stages are used with the Snowflake PUT command to upload files from a local file system? (Choose three.)

A. Schema Stage

B. User Stage

C. Database Stage

D. Table Stage

E. External Named Stage

F. Internal Named Stage

ANSWER: B D F

Explanation:

The correct stages are **User Stage**, **Table Stage**, and **Internal Named Stage**. The Snowflake `PUT` command transfers files from a client machine's local file system into Snowflake-managed internal storage. A user stage is the per-user internal stage referenced with `@~`, making it useful for files associated with the current user. A table stage is an internal stage associated with a specific table and is referenced with `@%<table_name>`; it is commonly used to stage files intended for loading into that table. An internal named stage is a schema-level stage object created with an internal Snowflake storage location and referenced by its stage name, such as `@my_stage`. These three internal stage types are supported `PUT` destinations. Snowflake documents the command syntax as uploading files to an internal stage and identifies user stages, table stages, and internal named stages as the supported targets. For command syntax and examples, see the Snowflake PUT command reference. For the distinctions among Snowflake stage types, see Creating an internal stage for local files.

QUESTION NO: 79

While unloading data into a stage, how can the user ensure that the output will be a single file?

A. Use the copy option files=single.

B. Use the COPY Option SINGLE=TRUE .

C. Use the get option SINGLE-TRUE.

D. Use the GET option FILES-SINGLE.

ANSWER: B

Explanation:

Use the COPY option SINGLE=TRUE. When unloading query or table results to an internal or external stage with the `COPY INTO <location>` command, the `SINGLE` copy option controls whether Snowflake writes the unloaded result as one file. Setting `SINGLE=TRUE` directs Snowflake to produce a single output file rather than splitting the result across multiple files. This is useful when a downstream consumer requires one file, or when the unloaded result is expected to be sufficiently small for one file to be practical. The option is included in the COPY command's unload options, for example: `COPY INTO @my_stage/output FROM my_table SINGLE=TRUE;`. Users should also account for the fact that producing a single file can reduce the parallelism normally available during unload operations, so it is generally best suited to use cases where the single-file requirement outweighs performance considerations. Snowflake documents `SINGLE = TRUE | FALSE` as an available copy option specifically for unloading data to a stage. See the official COPY INTO <location> reference and Snowflake's data unloading considerations.

QUESTION NO: 80

Which feature allows a user the ability to control the organization of data in a micro-partition?

A. Range Partitioning

B. Search Optimization Service

C. Automatic Clustering

D. Horizontal Partitioning

ANSWER: C

Explanation:

Automatic Clustering is correct because it maintains the organization of a table's data across Snowflake micro-partitions according to user-defined clustering keys. A clustering key identifies one or more columns or expressions that Snowflake should use to improve the co-location of related rows in micro-partitions. Once a clustering key is defined, Automatic Clustering continuously and asynchronously reclusters data as needed, helping preserve clustering quality as data is inserted, updated, or deleted.

This capability is especially useful for very large tables whose query patterns commonly filter or range-scan on the clustering-key columns. Better clustering can improve micro-partition pruning: Snowflake can eliminate partitions whose metadata shows they cannot contain qualifying rows, thereby reducing the amount of data scanned. Automatic Clustering is a serverless Snowflake service and consumes credits while it performs reclustering work. The user controls the intended physical data organization by defining the clustering key, while Snowflake handles the underlying micro-partition maintenance automatically. See Snowflake's Clustering Keys documentation and Automatic Clustering documentation.

QUESTION NO: 81

What are the responsibilities of Snowflake's Cloud Service layer? (Choose three.)

A. Authentication

B. Resource management

C. Virtual warehouse caching

D. Query parsing and optimization

E. Query execution

F. Physical storage of micro-partitions

ANSWER: A B D

Explanation:

Snowflake's Cloud Services layer provides the centralized services that coordinate and govern activity across the platform. **Authentication** is handled at this layer because Snowflake must validate user or service identities before allowing access to account resources. **Resource management** is also a Cloud Services responsibility in the sense that this layer manages and coordinates Snowflake resources and service operations, including the infrastructure-level management required to support user requests and virtual warehouses. **Query parsing and optimization** belongs here because submitted SQL is parsed, compiled, and optimized into an execution plan before compute resources process it. The Cloud Services layer also manages metadata, access control, and transaction coordination, allowing Snowflake to maintain a consistent, governed environment independently of any individual warehouse. Snowflake's architecture separates these control-plane capabilities from the compute layer, where virtual warehouses perform the actual query processing, and the storage layer, where persisted table data is maintained. See Snowflake's architecture overview and virtual warehouse overview for details.

QUESTION NO: 82

Which function can be used to convert semi-structured data into a relational representation?

A. FLATTEN

B. OBJECT_KEYS

C. PARSE_JSON

D. ARRAYS_TO_OBJECT

ANSWER: A

Explanation:

`FLATTEN` is a Snowflake table function that transforms semi-structured values—such as JSON stored in `VARIANT`, as well as ARRAY and OBJECT values—into a set of rows. This produces a relational row representation that can be queried with standard SQL. It is used in a `LATERAL` join (or comma-style lateral invocation) against a source table, and emits columns including `KEY`, `INDEX`, `PATH`, `VALUE`, and `THIS`. For example, applying `FLATTEN` to a JSON array returns one output row for each array element, allowing individual nested values to be selected, filtered, joined, and aggregated like ordinary relational data. The optional `RECURSIVE` parameter can expand nested arrays and objects throughout a hierarchy when required. Snowflake documents this function specifically as producing a lateral view of a VARIANT, OBJECT, or ARRAY expression, which is the mechanism for converting semi-structured content to rows. See the Snowflake FLATTEN function reference and its guide to querying semi-structured data.

QUESTION NO: 83

What are valid values for the FIELD_OPTIONALLY_ENCLOSED_BYoption in the copy into < location > command used during data unloading? (Select TWO).

A. Single quote character (')

B. NULL

C. ' NULL '

D. NONE

E. ' NONE '

ANSWER: A D

Explanation:

For CSV data unloaded with `COPY INTO <location>`, `FIELD_OPTIONALLY_ENCLOSED_BY` controls whether output fields are enclosed by a character. A single quote character is valid because Snowflake permits a character literal as the enclosure character. When specifying a single quote in SQL, it must be represented using the appropriate SQL

escaping/quoting syntax so that Snowflake interprets it as the intended one-character delimiter. This setting is useful when produced files must preserve field boundaries where values can contain delimiters, line breaks, or other special content.

`NONE` is also a valid value and is the keyword used when no optional field enclosure character should be written to the unloaded CSV output. It is a configuration keyword rather than a quoted character value. The setting belongs to the CSV file-format options that can be supplied directly in the unload command or defined in a named file format. Snowflake documents the accepted syntax as a character value or `NONE`; see the COPY INTO location format-type options and the CSV file-format option reference.

QUESTION NO: 84

What languages can be used to write User-Defined Functions, or UDFs? Select TWO.

A. Groovy

B. JavaScript

C. Python

D. Spark

E. Golang

ANSWER: B C

Explanation:

JavaScript and Python can both be used to implement Snowflake user-defined functions. A JavaScript UDF uses JavaScript as its handler language, enabling custom logic to be invoked in SQL expressions. Snowflake supports this form of UDF for scalar processing and provides JavaScript APIs that let the handler execute SQL statements when required. Python UDFs are implemented through Snowpark and run Python handler code in Snowflake's managed execution environment. They can use supported Python packages and are suitable for encapsulating reusable transformations, calculations, and other application logic close to governed Snowflake data. Both languages are explicitly included in Snowflake's supported handler languages for UDFs. In addition to these two, Snowflake supports SQL UDFs and Snowpark-based UDF handlers in Java and Scala, but only JavaScript and Python from the supplied choices satisfy the requested selection. For implementation syntax, runtime behavior, package availability, and limitations, consult Snowflake's User-Defined Functions overview and its Python UDF introduction.

QUESTION NO: 85

A Snowflake user has been granted the create data EXCHANGE listing privilege with their role.

Which tasks can this user now perform on the Data Exchange? (Select TWO).

A. Rename listings.

B. Delete provider profiles.

C. Modify listings properties.

D. Modify incoming listing access requests.

E. Submit listings for approval/publishing.

ANSWER: C E

Explanation:

The `CREATE DATA EXCHANGE LISTING` account privilege enables a role to create provider listings in a Snowflake Data Exchange. When a role creates a listing, it receives ownership of that listing, allowing the listing to be managed through its

lifecycle. This includes updating listing metadata and configuration, such as its descriptive properties and the data products associated with the listing. Therefore, *Modify listings properties.* is a supported task for listings owned by the role.

The same listing-management workflow supports submitting a listing for publication. In exchanges that use an approval process, a provider prepares the listing and submits it for approval; after approval, the listing can be published for consumer discovery. Consequently, *Submit listings for approval/publishing.* is also an appropriate capability associated with creating and managing a Data Exchange listing. These actions apply within the scope of listings that the role owns or is otherwise authorized to manage.

Snowflake documents Data Exchange listing creation and provider listing workflows in the Data Exchange listings documentation and the applicable account-level privilege is documented in the GRANT privileges reference.

QUESTION NO: 86

Which clause is used to define a function that may return different values for different rows?

A. IMMUTABLE

B. RETURNS

C. COMMENT

D. VOLATILE

ANSWER: D

Explanation:

`VOLATILE` is the correct clause for declaring a function whose result can vary across evaluations, including when it is invoked for different rows. In Snowflake, function volatility communicates whether the function can be assumed to produce the same output when called with the same input values. A volatile function can depend on changing state or non-deterministic behavior, such as the current time, generated random values, or information that can change between calls. Therefore, Snowflake must treat its result as potentially different for each evaluation rather than relying on a stable, reusable result. This declaration is especially relevant when defining a SQL user-defined function and documenting its expected evaluation behavior for users and query processing. Snowflake documents `VOLATILE` as the function-volatility property used when a function might return different results for the same arguments. See the CREATE FUNCTION reference and Snowflake's SQL UDF documentation for syntax and usage details.

QUESTION NO: 87

Which statements describe benefits of Snowflake ' s separation of compute and storage? (Select TWO).

A. The separation allows independent scaling of computing resources.

B. The separation ensures consistent data encryption across all virtual data warehouses.

C. The separation supports automatic conversion of semi-structured data into structured data for advanced data analysis.

D. Storage volume growth and compute usage growth can be tightly coupled.

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

ANSWER: A E

Explanation:

Snowflake separates persistent data storage from the compute resources used to query and transform that data. Therefore, "The separation allows independent scaling of computing resources." is correct: virtual warehouses can be independently sized to provide the processing capacity appropriate for a workload. A team can use a smaller warehouse for routine reporting and a larger warehouse for an intensive transformation or loading task, while accessing the same centrally stored

data. This enables performance tuning based on workload needs rather than requiring a change to the underlying stored data.

“Compute can be scaled up or down without the requirement to add more storage.” is also correct. A virtual warehouse is a compute cluster; resizing it changes the compute resources available for query execution, not the amount of persistent Snowflake storage. Likewise, data can continue to be retained and grow in Snowflake storage independently of a warehouse’s size or whether a warehouse is running. This architecture supports workload isolation and flexible resource allocation because multiple virtual warehouses can access the same data without copying it. Snowflake describes its architecture and independently scalable compute and storage layers in its key concepts documentation and virtual warehouses overview.

QUESTION NO: 88

Which setting will help control credit consumption for a virtual warehouse that needs to run in Auto-scale mode?

A. Set the warehouse scaling policy to ECONOMY.

B. Set the MIN_CLUSTER_COUNT to 1.

C. Set the AUTO_SUSPEND property of the warehouse to 0.

D. Set the MIN_CLUSTER_COUNT to 2 and set the MAX_CLUSTER_COUNT to 2.

ANSWER: A

Explanation:

Set the warehouse scaling policy to ECONOMY. A multi-cluster virtual warehouse in Auto-scale mode can automatically start additional clusters as concurrent query demand rises and shut them down as demand falls. The ECONOMY scaling policy is specifically intended to conserve credits in this scenario. It applies a more conservative threshold for adding clusters than the default STANDARD policy: Snowflake starts another cluster only when it estimates that the queued workload is sufficient to keep that cluster busy for at least six minutes. By avoiding short-lived cluster starts for temporary spikes in query concurrency, ECONOMY can reduce unnecessary compute usage while retaining Auto-scale capability for sustained demand. This policy is configured through the warehouse’s SCALING_POLICY property and is appropriate when cost control is more important than minimizing query queueing during brief bursts. Snowflake notes that changing the scaling policy applies to subsequent cluster provisioning decisions; running clusters are not immediately affected. For the related configuration syntax and behavior, see Snowflake’s ALTER WAREHOUSE documentation and its multi-cluster warehouse documentation.

QUESTION NO: 89

Which query types will have significant performance improvement when run using the search optimization service? (Select TWO)

A. Range searches

B. Equality searches

C. Substring searches

D. Queries with IN predicates

E. Queries with aggregation

ANSWER: B C D E

Explanation:

Search Optimization Service maintains search access paths that can substantially reduce the number of micro-partitions scanned for supported, sufficiently selective query patterns. Equality searches are a core use case: predicates that look up

specific values can use an equality search access path rather than requiring a broad table scan. Queries with `IN` predicates are included in this point-lookup capability because they search for one or more specific values. Snowflake also supports search optimization for substring searches, allowing qualifying `LIKE`, `ILIKE`, and certain regular-expression patterns to benefit from substring search access paths. In addition, the service can accelerate supported aggregation workloads, including eligible `MIN`, `MAX`, and `COUNT` queries, when the relevant columns are configured for search optimization. The actual improvement depends on selectivity, table size, and the search optimization configuration; it is not a replacement for appropriate warehouse sizing or general query design. Snowflake documents point lookups, substring/regular-expression searches, and aggregation queries as query categories that can benefit from this service. See the Search Optimization Service overview and Snowflake's guidance on search optimization for aggregation queries.

QUESTION NO: 90

What is the maximum Time Travel retention period for a temporary Snowflake table?

A. 90 days

B. 1 day

C. 7 days

D. 45 days

ANSWER: B

Explanation:

The maximum Time Travel retention period for a temporary Snowflake table is **1 day**. Temporary tables are designed for session-scoped, short-lived work such as intermediate processing, staging, and transient calculations. Like transient tables, they do not receive Snowflake Fail-safe protection and support a Time Travel retention period of only zero or one day. This limit applies even when the account edition or a related database/schema configuration permits longer retention for permanent objects. The retention setting is controlled through the `DATA_RETENTION_TIME_IN_DAYS` parameter, which determines how long historical data can be accessed for supported Time Travel operations, including querying prior versions of data and recovering dropped objects where applicable. For a temporary table, the parameter can be set to a maximum of one day; it cannot be extended to the multi-day retention periods available for permanent tables in eligible Snowflake editions. Snowflake also notes that temporary tables are automatically dropped when the session that created them ends, so they are intended for temporary session activity rather than long-term recoverability. See Snowflake's Temporary and transient tables documentation and the Time Travel documentation.