

EXAM PREPARATION MATERIAL

Practice files designed to help you understand exam patterns, question styles and the overall assessment format.

IMPORTANT NOTE



Please note that these files may not be up to date.
However, the questions will help you understand the
exam format and typical question patterns.



Stay connected with our support team
for the latest file updates and exam-related information.

Question: 1

What are computer language considerations when using Snowflake interfaces? (Select TWO).

- A. Notebook cells can be written in Python.
- B. Databases can process queries using Scala.
- C. Notebook cells can be written in SQL.
- D. Worksheets can only be written in SQL.
- E. Dashboards can be written in JavaScript.

Answer: A,C

Explanation:

Snowflake Notebooks support both SQL and Python as executable cell types. This means that users can create notebook cells written in Python for programmatic data processing and modeling, and in SQL for declarative querying and transformation.

Because of this, the statements “Notebook cells can be written in Python” and “Notebook cells can be written in SQL” are both correct.

Snowflake databases do not natively execute queries written in Scala; Scala is supported via Snowpark APIs for application code, not as a direct query language. Worksheets in Snowsight are primarily SQL-based and also support procedural constructs (e.g., Snowflake Scripting), so the phrase “only in SQL” is not a precise or complete characterization. Dashboards in Snowsight are created using SQL-backed visualizations and built-in UI components; JavaScript is not a supported authoring language inside the native dashboarding layer. Therefore, options B, D, and E are not correct in this context.

Question: 2

What Snowflake object is used to store data?

- A. Table
- B. Virtual warehouse
- C. Stored procedure
- D. View

Answer: A

Explanation:

A table is the core Snowflake object used to store structured data. Tables persist rows and columns in Snowflake's storage layer, which internally uses compressed, columnar micro-partitions managed by the platform. Tables can be permanent, transient, or temporary, but in every case, they are the primary objects that hold the actual data.

A virtual warehouse is a compute resource used to process queries and perform operations; it does not store user data. A stored procedure contains executable logic (such as complex workflows or procedural code) but does not serve as a data container. A view is a logical object that presents the results of a saved query; it references underlying tables and does not store its own data. As a result, "Table" is the correct answer.

Question: 3

Where is unstructured data stored in Snowflake?

- A. In the Cloud Services layer
- B. In internal or external stages
- C. In external tables
- D. In tables with a single VARCHAR column

Answer: B

Explanation:

Unstructured data such as PDF files, images, and other binary documents is stored in stages in Snowflake. These stages may be internal stages, which Snowflake manages directly, or external stages, which reference external cloud storage such as Amazon S3, Azure Blob Storage, or Google Cloud Storage. Stages are the designed mechanism for storing and accessing unstructured files so that they can be processed with functions like `PARSE_DOCUMENT` or accessed via directory tables.

External tables are used to query structured or semi-structured data (for example, Parquet or JSON) stored in external locations, not to store raw unstructured binary content. The Cloud Services layer coordinates metadata, security, and query services; it does not store user data. Tables with a single VARCHAR column might be used as an improvised approach for small text blobs, but this is not the native or recommended method for managing unstructured data at scale.

Question: 4

What is the name of Snowflake's default web-based interface?

- A. SnowSQL
- B. Snowpark
- C. Snowsight
- D. Snowflake CLI

Answer: C

Explanation:

Snowsight is the default web-based interface for Snowflake. It provides a graphical, browser-based environment to write and run SQL and Python, manage and explore database objects, create visualizations and dashboards, monitor query and warehouse activity, and collaborate via worksheets. It has replaced the Classic Console as the default UI for new Snowflake accounts.

SnowSQL is a command-line client used to connect to Snowflake from a terminal and execute SQL statements. Snowpark is a developer framework that supports writing data applications in languages like Python, Java, and Scala, but it is not a UI. "Snowflake CLI" refers to command-line tooling and automation utilities, again not a graphical web interface.

Question: 5

What task can be performed on the Snowsight Schema Details page?

- A. Change the schema name.
- B. Share a schema with a different account.

C. Truncate the schema data.

D. Edit the schema metrics.

Answer: A

Explanation:

On the Snowsight Schema Details page, one of the supported operations is renaming the schema. The UI exposes controls that allow users with appropriate privileges to change the schema name, which can help maintain consistent naming conventions or reflect project reorganizations.

Sharing data with other accounts is typically done using secure shares at the database or object level, not from a simple “Share this schema” function on the Schema Details page. Truncation is a table-level operation (e.g., TRUNCATE TABLE), not something that applies at the schema level. Metrics visible on the Schema Details page (such as object counts or storage usage) are informational and not directly editable; they are derived from system metadata.

Question: 6

What is the impact of enabling auto-suspend on a Snowflake virtual warehouse?

A. The size of the warehouse will adjust automatically, improving query performance.

B. The warehouse will be dropped after a specified period of inactivity.

C. The warehouse will be stopped after a specified period of inactivity, reducing costs.

D. The warehouse costs will stabilize from month-to-month.

Answer: C

Explanation:

When auto-suspend is enabled on a virtual warehouse, Snowflake automatically suspends (stops) the warehouse after it has been idle for the configured period of time. Once suspended, the warehouse stops consuming compute credits, thereby reducing costs, since Snowflake bills only for active compute usage. When new queries arrive, auto-resume (if enabled) restarts the warehouse automatically.

Auto-suspend does not change the warehouse size; scaling up or down is a separate configuration. The warehouse is not dropped—its metadata and settings remain intact. Costs may be reduced and made more efficient but are not guaranteed to

“stabilize month-to-month,” as the bill still depends on actual usage and workload patterns.

Question: 7

What parameter is used to define how long Time Travel can be used to access a table?

- A. DATE_OUTPUT_FORMAT
- B. DATA_RETENTION_TIME_IN_DAYS
- C. TIMEZONE
- D. USE_CACHED_RESULT

Answer: B

Explanation:

The DATA_RETENTION_TIME_IN_DAYS parameter controls the Time Travel retention period for Snowflake objects such as tables, schemas, and databases. It specifies the number of days that historical data (prior versions of rows or dropped objects) is retained and accessible via Time Travel. Within this retention window, users can query data “as of” a previous time, restore dropped objects, or clone objects at a historical point.

DATE_OUTPUT_FORMAT determines the display format for date values and is unrelated to historical retention. TIMEZONE affects how timestamps are interpreted and displayed, not how long data history is preserved. USE_CACHED_RESULT governs whether Snowflake may return cached query results, not Time Travel behavior.

Question: 8

What filter can be used when searching for data products on the Snowflake Marketplace?

- A. Size
- B. Compatibility
- C. Price
- D. Performance

Answer: C

Explanation:

On the Snowflake Marketplace, users can apply a Price filter when searching for data products. This filter helps distinguish between free offerings and paid listings, making it easier to align data acquisition with budget and commercial terms. Marketplace metadata includes pricing information that this filter uses to refine search results.

Dataset “Size” is not exposed as a primary Marketplace search filter. “Compatibility” is implied, as all Marketplace products are consumable within Snowflake. “Performance” is not a filterable attribute, because performance is influenced mainly by the consumer’s compute configuration and query design, rather than a fixed property of the Marketplace listing.

Question: 9

What is the purpose of the USE SCHEMA command in Snowflake?

- A. To modify the schema structure in a session
- B. To create a new schema in a database
- C. To take ownership of an existing schema
- D. To set the current schema for a session

Answer: D

Explanation:

The USE SCHEMA command sets the active schema context for the current session. After it is executed, any unqualified object names (for example, SELECT * FROM my_table) are resolved within that schema. This reduces the need to fully qualify object names with database and schema each time and ensures that statements reference the expected logical container.

CREATE SCHEMA is used to create a new schema. ALTER SCHEMA and GRANT OWNERSHIP are used to modify schema properties or transfer ownership, respectively. USE SCHEMA does not alter structure or ownership; it simply changes the context in which subsequent SQL statements are interpreted.

Question: 10

Which Snowflake component is responsible for data encryption?

- A. Database storage
- B. Virtual warehouse
- C. Data loading utility
- D. Cloud services

Answer: A

Explanation:

Snowflake's database storage layer is responsible for encrypting data at rest. All persisted data— whether structured, semi-structured, or unstructured—is encrypted using strong encryption algorithms such as AES-256. This process is automatic and transparent to users, ensuring that files and micro-partitions stored in Snowflake-managed cloud storage are encrypted by default. Data in transit is protected separately through TLS.

Virtual warehouses provide compute resources to execute queries and do not perform storage-level encryption. Data loading utilities (such as COPY INTO and client tools) orchestrate data movement but do not handle at-rest encryption. The Cloud Services layer manages metadata, session management, security policies, and query optimization, but the actual encryption of stored data is

part of the storage subsystem.

Question: 11

What information is available when previewing a table in Snowsight?

- A. The name of the virtual warehouse where the table is stored
- B. The first 100 rows of the table
- C. The table definition
- D. The Entity Relationship (ER) diagram for the table

Answer: B

Explanation:

When you preview a table in Snowsight, Snowflake returns up to the first 100 rows of the table's data. This preview is designed for quick inspection of the table's contents and structure without requiring the user to manually write a SELECT query. The preview interface allows basic exploration such as scrolling, simple sorting, and viewing column metadata.

Tables are not "stored" in a specific virtual warehouse; warehouses provide compute only. The table definition (DDL) can be viewed in the Table Details section but is not the main output of the Preview action. Snowsight does not natively generate an Entity Relationship (ER) diagram for the table as part of the preview.

Question: 12

What is a Markdown cell in Snowflake Notebooks?

- A. A cell that contains data in more than one computer languages
- B. A cell that allows the user to format the cell text
- C. A cell that is nested in other cells
- D. A cell that contains older data

Answer: B

Explanation:

In Snowflake Notebooks, a Markdown cell is a non-executable cell type used for formatted text. It allows the user to format text using Markdown syntax, including headings, lists, tables, emphasis, inline code, and links. This makes it possible to

document the analysis, describe steps, and provide commentary alongside SQL and Python cells, improving clarity and collaboration.

Markdown cells do not execute code in “more than one computer language”; they are not code cells at all. Notebook cells are not nested; each cell exists as a separate element in the notebook. There is no concept of “older data” associated specifically with Markdown cells; they simply store text content defined by the user.

Question: 13

What is the CREATE FILE FORMAT command used for in Snowflake?

- A. To delete a file format
- B. To define the format of data files for processing
- C. To modify an existing file format in a table
- D. To create a new table

Answer: B

Explanation:

The CREATE FILE FORMAT command creates a file format object in Snowflake that defines how data files should be interpreted when loading or unloading. Parameters in the definition include file type (for example, CSV, JSON, PARQUET), field delimiters, compression, quoting rules, and other parsing options. Once created, this file format object can be referenced by COPY INTO commands or external table definitions to ensure consistent handling of files.

Deleting a file format is performed using DROP FILE FORMAT. Modifying an existing file format is done with ALTER FILE FORMAT. Creating a new table is achieved with CREATE TABLE. Therefore, the purpose of CREATE FILE FORMAT is specifically to define how Snowflake should process data files.

Question: 14

What Snowflake object provides a secure connection to external cloud storage?

- A. A directory table
- B. An external table
- C. An external stage
- D. A named file format

Answer: C

Explanation:

An external stage is the Snowflake object that encapsulates a secure connection to external cloud storage such as Amazon S3, Azure Blob Storage, or Google Cloud Storage. It stores the location (URL or bucket path) and, where required, credentials or role-based access configuration, and may also reference a file format. External stages are used as the source or target for COPY INTO operations when loading from or unloading to external storage.

An external table provides a logical SQL interface to data stored externally but relies on a stage for connectivity; it does not itself define the connection. A directory table exposes metadata about files stored in a stage, not the connection. A named file format defines parsing rules (type, delimiter, compression) but has no knowledge of or connection to a specific external storage location.

Question: 15

What are Snowflake customers responsible for?

- A. Managing the hardware on which Snowflake is installed
- B. Software installation and updates
- C. Managing table metadata
- D. Loading, unloading, and managing data

Answer: D

Explanation:

As a fully managed cloud data platform, Snowflake is responsible for infrastructure provisioning, hardware, software installation, platform upgrades, scaling, and internal metadata management such as micro-partitions and statistics. Customers do not manage physical hardware or install Snowflake software.

Customers are responsible for their data and its lifecycle within Snowflake. This includes loading data into tables from internal and external sources, unloading data when required, organizing data structures (databases, schemas, tables), defining access controls, and managing how data is used, transformed, and governed. They design schemas and workloads but do not manage the underlying engine. Therefore, "Loading, unloading, and managing data" correctly describes the customer's responsibility.

Question: 16

What is the purpose of assigning roles to users in Snowflake?

- A. To manage and enforce permissions on objects
- B. To define user tasks
- C. To optimize query execution plans
- D. To control which data types users can query

Answer: A

Explanation:

Snowflake uses a Role-Based Access Control (RBAC) model, where roles are the containers of privileges. Assigning roles to users ensures that permissions on database objects (such as tables, schemas, warehouses, and functions) are enforced consistently and securely. Users do not receive privileges directly; instead, privileges are granted to roles, and roles are assigned to users.

This enables scalable, auditable, and manageable access control.

Roles do not determine tasks, do not affect query optimization, and do not govern which data types a user may query—permissions are object-based, not datatype-based.

Question: 17

What is a key capability of the Snowflake virtual warehouse?

- A. It supports unlimited concurrency.
- B. It can be dynamically scaled up or down.
- C. It can store data permanently.
- D. It can be located on-premises.

Answer: B

Explanation:

A virtual warehouse provides compute resources in Snowflake and can be resized (scaled up or down) at any time. Scaling up increases compute power for intensive workloads, while scaling down reduces cost for lighter workloads.

Virtual warehouses do not store data—that is handled by Snowflake's independent storage layer. They are not on-premises.

and do not provide unlimited concurrency; Snowflake supports high concurrency using multi-cluster warehouses but not boundless concurrency.

Question: 18

What is the name for a Snowflake account holder who shares data with other Snowflake accounts?

- A. A data provider
- B. A data consumer
- C. A data reader
- D. A data administrator

Answer: A

Explanation:

In Snowflake's Secure Data Sharing model, the account sharing data is known as the data provider. Providers create shares that include selected objects (tables, views, secure views, etc.) and grant access to consumer accounts.

A data consumer receives the shared data. Data reader and data administrator are not defined terms in Snowflake's sharing architecture.

Question: 19

What are the key benefits of using Snowflake? (Select TWO).

- A. Independent scaling of compute and storage
- B. Vendor lock-in
- C. Manual infrastructure management
- D. Near-unlimited concurrency
- E. On-premises deployment

Answer: A,D

Explanation:

Snowflake's multi-cluster shared data architecture enables independent scaling of compute and storage, meaning workloads can be sized differently without affecting stored data.

It also provides near-unlimited concurrency, because Snowflake can automatically add compute clusters during peak loads.

Snowflake does not require manual infrastructure management and runs fully in the cloud—not on premises. Multi-cloud support reduces vendor lock-in, not increases it.

Question: 20

What is a benefit of using an external stage to load data into Snowflake?

- A. External stages reduce data storage costs because data is stored outside Snowflake.
- B. External stages provide automatic data purging after successful loads.
- C. External stages are more secure than internal stages for sensitive data.
- D. External stages reduce the number of objects in a database.

Answer: A

Explanation:

External stages point to files in external cloud storage (S3, Azure Blob, GCS). Because the data is not stored inside Snowflake, the user avoids Snowflake storage charges, which can significantly reduce cost for large staging datasets.

External stages do not automatically delete files, are not inherently more secure than internal stages, and still count as database objects even though they reference external storage.

Question: 21

What is the LIST command used for in Snowflake?

- A. Lists all the users assigned to a given role.
- B. Lists all the files in a given stage.
- C. Lists all the tables in a specific schema.

D. Lists all the privileges granted on a specific object.

Answer: B

Explanation:

The `LIST` command displays all files available in a stage—internal, external, user, or table stages.

It is typically used prior to `COPY INTO` to confirm which files are available for loading.

`SHOW TABLES` lists tables; `SHOW GRANTS` shows privileges; `GRANTS` queries show user-role

assignments—not `LIST`.

Question: 22

What is the `SHOW FILE FORMATS` SQL command used for in Snowflake?

- A. Create a list of all of the file formats currently supported by Snowflake.
- B. Get information about how different file formats should be used.
- C. List the file formats in the currently-selected schema.
- D. List all of the files in a database that have data in a given file format.

Answer: C

Explanation:

`SHOW FILE FORMATS` lists file format objects that exist in the currently selected schema (or optionally, a specific database or schema). These objects define how Snowflake interprets staged files during data loading.

It does not list supported built-in formats (CSV, Parquet, etc.).

It does not show example usage information and does not list files—`LIST @stage` handles that.

Question: 23

What Snowflake object is used to organize database objects into logical groupings?

- A. Schemas
- B. Roles
- C. Tables
- D. Stages

Answer: A

Explanation:

Schemas serve as logical containers within a database, grouping tables, views, file formats, functions, and other objects.

Roles control access, not organization.

Tables store data but do not group objects.

Stages store files or point to external storage; they are not organizational containers.

Question: 24

What are the benefits of using the Snowsight data loading interface? (Select TWO).

- A. It creates permanent file formats that can be used to load data in the future.
- B. It allows a user to insert the records of a supported file into a table.
- C. It will try to detect data types.
- D. It allows a user to optimize data loading into a table.
- E. It lets a user merge file rows into the table records.

Answer: B,C

Explanation:

Snowsight's file-loading interface:

- Automatically detects data types by inspecting column values.
- Loads (inserts) file contents directly into a table, either a new table or an existing one.

Snowsight does not automatically create permanent file formats, does not merge rows, and does not optimize warehouse-level performance.

Question: 25

What is the purpose of the SHOW GRANTS SQL command in Snowflake?

- A. To list all roles in the account
- B. To list all roles granted to a user
- C. To list all users in the account
- D. To list all databases in the account

Answer: B

Explanation:

SHOW GRANTS is used to list privileges, including roles assigned to a user.

SHOW GRANTS TO USER <user> reveals all the roles that user inherits.

SHOW ROLES lists roles, SHOW USERS lists users, SHOW DATABASES lists databases—not SHOW GRANTS.

Question: 26

What is the purpose of the COPY INTO SQL command?

- A. To copy data into a virtual warehouse
- B. To copy data from one table into another
- C. To load data from a stage into a table
- D. To duplicate all files on a stage

Answer: C

Explanation:

COPY INTO <table> loads data from staged files (internal or external stages) into a Snowflake table.

It supports pattern matching, file formats, error-handling modes, and parallel loading.

It does not move files, does not load to a warehouse (warehouses only supply compute), and table- to-table copying uses INSERT INTO...SELECT.

Question: 27

Snowflake stores semi-structured data in a column of which data type?

- A. TEXT
- B. VARCHAR
- C. STRING
- D. VARIANT

Answer: D

Explanation:

The VARIANT data type stores semi-structured data (JSON, Parquet, XML, Avro, ORC).

VARIANT allows schema-on-read and supports nested structures.

TEXT/VARCHAR/STRING store plain text only and cannot efficiently query nested structures. STRING is not a Snowflake-native type—VARCHAR is used instead.

Question: 28

What SQL command will return a list of all files in a stage?

- A. PUT
- B. LIST
- C. GET

D. COPY INTO

Answer: B

Explanation:

LIST @stage_name displays all files present in a stage (internal or external).

PUT uploads local files to an internal stage.

GET downloads files from an internal stage.

COPY INTO loads or unloads data but does not list files.

Question: 29

Which Snowflake Cortex LLM function allows a user to specify which model will be used?

A. COMPLETE

B. SENTIMENT

C. SUMMARIZE

D. TRANSLATE

Answer: A

Explanation:

The COMPLETE function is the only Cortex LLM function where the user can explicitly specify the model name (e.g., Mistral, Snowflake Arctic, Meta Llama models).

SUMMARIZE, SENTIMENT, and TRANSLATE are task-specific functions that use built-in, preselected models and do not allow model selection.

Question: 30

What is the Snowsight object browser used for?

- A. To manage virtual warehouses
- B. To create Snowsight worksheets
- C. To modify security settings for objects
- D. To view and filter database objects

Answer: D

Explanation:

The Snowsight object browser is Snowflake's built-in visual interface for exploring and navigating all database objects within an account. It provides a hierarchical view across databases, schemas, and contained objects such as tables, views, stages, pipes, file formats, procedures, and functions. The key purpose of the object browser is to make it easy for users to view and filter database objects without needing SQL commands. Users can search objects by name, apply filters, preview table metadata, and inspect properties such as storage usage or row counts.

The object browser does not manage warehouses; warehouse management is done under the "Warehouses" section. It also does not create worksheets; worksheets are created in the "Worksheets" section. Additionally, the object browser does not modify security settings—permissions and grants are controlled through SQL (GRANT, REVOKE, SHOW GRANTS) or through the Access Control interfaces.

In summary, the object browser is a discovery and navigation layer, allowing users to efficiently find, inspect, and explore Snowflake objects.

Question: 31

What information can be accessed using the Snowsight Monitoring tab?

- A. Virtual warehouse usage metrics
- B. Query execution history
- C. Database Time Travel snapshots
- D. Database schema changes history

Answer: A

Explanation:

The Snowsight Monitoring tab provides a centralized view of virtual warehouse usage metrics, enabling administrators and developers to evaluate how compute resources are being consumed. This includes critical insights such as credit usage, query load, concurrency levels, average queue times, execution durations, and auto-scaling activity (for multi-cluster warehouses). These metrics help determine whether a warehouse is correctly sized, whether concurrency issues are occurring, or whether workloads require scaling up or adding clusters.

Query history is available in a different section—"Activity → Query History"—not under Monitoring. Time Travel snapshots are not visualized within Monitoring; Time Travel is controlled via retention parameters and accessed with SQL (AT/BEFORE clauses). Schema change history is also not part of Monitoring and instead is discoverable through ACCOUNT_USAGE or specific metadata views.

The Monitoring tab exists specifically to help evaluate warehouse performance and resource consumption, enabling optimization of compute spending and better workload management.

Question: 32

Where do tables reside in Snowflake?

- A. Stages
- B. Virtual warehouses
- C. Schemas
- D. File formats

Answer: C

Explanation:

In Snowflake's logical hierarchy, tables reside within schemas, and schemas reside within databases. This structured arrangement allows clear organization, access management, and separation of workloads or business domains. A schema functions as a container that holds tables, views, functions, sequences, stages, and file formats. When users reference a table, they typically qualify it using the full three-part name: `database.schema.table`.

Stages are used to store or reference files, not to hold structured database tables. Virtual warehouses provide compute power for executing queries and performing DML operations but do not store data. File formats define how staged files should be interpreted (CSV, JSON, Parquet, etc.) and are unrelated to where tables physically reside.

Snowflake separates compute and storage, so while compute happens in virtual warehouses, all persistent data—including tables—is managed in Snowflake's centralized storage layer, organized through schemas and databases. This organization ensures consistency, ease of management, and predictable access control.

Question: 33

What is a key characteristic of the Snowflake architecture's Cloud Services Layer?

- A. It stores all customer data.
- B. It manages virtual warehouses.
- C. It handles security and metadata management.
- D. It provides the user interface for Snowsight.

Answer: C

Explanation:

The Cloud Services Layer is the coordination and control layer of Snowflake's architecture. One of its primary responsibilities is managing security, metadata, authentication, and system-wide services. This layer handles user authentication, role-based access control, metadata services (such as table structures, micro-partition metadata, statistics), query parsing, optimization, execution coordination, and transaction management.

It does not store customer data; storage is handled by the Database Storage Layer using micropartitions. It does not manage virtual warehouses directly; warehouses are part of the Compute Layer. While Snowsight is a UI that interacts with the Cloud Services Layer, the interface itself is not part of the architectural layer.

The Cloud Services Layer essentially acts as the “brain” of Snowflake, ensuring the platform is consistent, secure, optimized, and able to scale operations intelligently across compute clusters and cloud-native storage environments.

Question: 34

What is the Snowsight Query Profile used for?

- A. To execute SQL queries
- B. To create new database objects
- C. To manage data loading processes
- D. To visualize and analyze query performance

Answer: D

Explanation:

The Snowsight Query Profile is a powerful diagnostic tool that provides a visual breakdown of how Snowflake executed a query. Its primary purpose is to help users visualize and analyze query performance. It displays execution steps, including scan operations, join strategies, pruning results, aggregation methods, and data movement between processing nodes.

The profile shows metrics such as execution time per step, partition pruning effectiveness, bytes scanned, and operator relationships. This allows developers, analysts, and DBAs to identify bottlenecks—such as unnecessary full-table scans, non-selective filters, or inefficient joins—and tune SQL accordingly.

Query Profile does not execute queries; execution happens in worksheets or programmatic interfaces. It does not create objects or manage data loading; those tasks involve separate SQL commands and UI interfaces.

Overall, Query Profile is essential for performance tuning, helping teams reduce compute costs, optimize warehouse

sizing, and improve query efficiency.

Question: 35

What syntax will enable the use of a Python string variable named myvar in a SQL cell within a Snowflake Notebook?

- A. \$myvar
- B. 'myvar'
- C. myvar
- D. {{myvar}}

Answer: D

Explanation:

Snowflake Notebooks support cross-cell interaction between Python and SQL by using Jinja-style templating syntax. To reference a Python variable inside a SQL cell, you wrap the variable name in double curly braces, like {{myvar}}. During execution, the Notebook engine substitutes the Python variable's value into the SQL statement before sending it to Snowflake.

This mechanism allows dynamic SQL generation, parameterization of queries, incorporating Python logic into SQL workflows, and building interactive analytics pipelines.

Other provided options are invalid in Snowflake Notebooks: \$myvar resembles shell syntax and is not supported; 'myvar' inserts a literal string rather than the variable's value; using myvar alone would cause SQL to interpret it as a column or object name.

Therefore, only {{myvar}} correctly represents Snowflake Notebook variable substitution syntax.

Question: 36

What cell types are available in Snowflake Notebooks? (Select THREE).

- A. Java
- B. R
- C. Scala
- D. SQL
- E. Markdown
- F. Python

Answer: D,E,F

Explanation:

Snowflake Notebooks currently support three primary cell types: SQL, Python, and Markdown. SQL cells allow users to execute SQL queries directly against Snowflake data. Python cells enable computation, data transformation, machine learning, and visualization using Snowpark, pandas-like APIs, and Python libraries. Markdown cells provide rich text formatting to document workflows, add explanations, and create readable narratives within the notebook.

Languages such as Java, Scala, and R are supported by Snowflake outside notebooks—for example, through Snowpark APIs or external integrations—but they cannot be used directly as Notebook cell types. Notebooks are designed to integrate SQL and Python seamlessly while providing a documentation layer, making SQL, Python, and Markdown the correct and only supported options.

Question: 37

What is created in the Cloud Services layer of the Snowflake architecture?

- A. Dashboards
- B. Metadata
- C. Virtual warehouses

D. Micro-partitions

Answer: B

Explanation:

The Cloud Services Layer is responsible for generating and managing metadata, including object definitions, table schemas, micro-partition statistics, column-level profiles, access control information, and query optimization metadata.

Metadata plays a central role in Snowflake's performance and functionality because it informs pruning, query planning, and efficient execution.

Dashboards are created in Snowsight or external BI tools. Virtual warehouses belong to the Compute Layer, providing processing resources. Micro-partitions are created in the Storage Layer, where Snowflake automatically organizes compressed columnar data for efficient access.

Consequently, the Cloud Services Layer is where metadata—not data, not compute resources—is created and managed.

Question: 38

What is the PRIMARY purpose of the use of the PARSE_DOCUMENT function in Snowflake?

- A. To identify any Personally Identifiable Information (PII) in text
- B. To identify data that will benefit from the use of a directory table
- C. To extract text from PDF files
- D. To parse JSON data

Answer: C

Explanation:

The PARSE_DOCUMENT function is part of Snowflake Cortex AI and is designed specifically to extract text, layout

information, and structured elements from unstructured documents, especially PDFs. It supports OCR-based extraction for scanned files and layout-aware extraction to preserve tables, headings, and format structure.

Its purpose is not PII detection; Snowflake does not provide built-in automatic PII identification via `PARSE_DOCUMENT`. It does not identify candidate data for directory tables and is unrelated to JSON parsing—Snowflake uses `PARSE_JSON` for JSON data.

`PARSE_DOCUMENT` is primarily used for workflows such as contract analysis, invoice extraction, document classification, compliance automation, and downstream AI enrichment.

Question: 39

What tasks can be performed using Snowflake Cortex AI? (Select TWO).

- A. Simplify unstructured data workflows.
- B. Share data through the Snowflake Marketplace.
- C. Load semi-structured data.

D. Extract and classify text.

E. Enhanced data security.

Answer: A,D

Explanation:

Snowflake Cortex AI provides built-in AI functions and tools designed to work natively with unstructured and structured data. Two key capabilities are:

- Extract and classify text using functions like `PARSE_DOCUMENT`, `EXTRACT_TEXT`, and classification models. Cortex can process documents, identify relevant fields, and convert unstructured content into usable structured formats.
- Simplify unstructured data workflows by combining document extraction, vector search, summarization, and AI reasoning tools (e.g., Cortex Analyst, Cortex Search) directly inside Snowflake **without external services**.

It does not provide Marketplace data sharing features, which belong to Snowflake's Data Sharing platform. Loading semi-structured data is a core Snowflake capability using `VARIANT` and `COPY INTO`—not Cortex-specific. Enhancing data security is a platform-wide feature, not a Cortex function.

Question: 40

What is a key characteristic of a Snowflake virtual warehouse?

- A. It provides compute resources.
- B. It manages account roles.
- C. It permanently stores data.
- D. It encrypts data.

Answer: A

Explanation:

A virtual warehouse is the compute engine of Snowflake. It provides CPU, memory, and temporary storage needed to execute SQL queries, data loading operations, and DML actions. Warehouses can be sized dynamically and suspended or resumed to optimize cost.

Warehouses don't store data; Snowflake's storage is independent and centralized. Warehouses do not manage roles — access control is handled through Snowflake's RBAC system. Encryption is performed automatically by Snowflake's storage and cloud services, not by warehouses.

Thus, the correct characteristic is that virtual warehouses supply compute.

If you'd like, I can provide Questions 42–55 next, with the same 150–200-word explanations.

You said:

Questions 42–55 next, with the same 150–200-word explanations

ChatGPT said:

Below are Questions 42–55, all formatted exactly per your requirements, with 150–200-word explanations and every original option preserved exactly.

Question: 41

In the Query Profile, what does the Pruning section provide?

- A. Information on how Snowflake removed rows from the query results.
- B. Information on how Snowflake removed columns from the query results.
- C. Information on how Snowflake removed objects from the query plan.
- D. Information on how Snowflake removed micro-partitions from the query scan.

Answer: D

Explanation:

The Pruning section of the Snowsight Query Profile shows how Snowflake eliminated unnecessary micro-partitions from the scan phase of the query. Snowflake stores data in micro-partitions and maintains metadata such as min/max values for each column within each partition. When a query includes filters (e.g., WHERE clauses), Snowflake evaluates this metadata to determine which micro-partitions cannot possibly satisfy the predicate. These partitions are skipped, meaning they are never scanned or read from storage.

This process drastically improves performance because Snowflake minimizes I/O, reduces compute usage, and shortens execution time. Partition pruning is especially impactful on large tables because only a fraction of the stored

micro-partitions typically need to be accessed.

The Pruning section does not show removed rows—that happens during the filter step. It does not show removed columns—column pruning is handled separately by the optimizer. It also does not show removed objects from the plan. Its sole purpose is to document micro-partition elimination and scan reduction.

Question: 42

What are the key benefits of the Snowflake multi-cluster shared data architecture? (Select TWO).

- A. It provides enhanced data security features.
- B. It optimizes data loading for unstructured data.
- C. It allows for independent scaling of compute and storage.
- D. It stores data in a columnar format to improve performance.
- E. It offers near-unlimited concurrency and elasticity.

Answer: C,E

Explanation:

The Snowflake multi-cluster shared data architecture separates compute and storage, enabling independent scaling of each. Compute resources (virtual warehouses) operate independently from the centralized storage layer, allowing users to increase compute power without impacting storage costs—or vice versa. This flexibility provides major advantages for cost optimization and workload performance tuning.

The architecture also supports near-unlimited concurrency and elasticity. When many users or workloads run simultaneously, Snowflake can automatically add additional clusters to a multi-cluster warehouse, ensuring that no queries experience queuing or performance degradation. This capability is crucial for BI dashboards, ETL pipelines, and large organizations with varied usage patterns.

Columnar storage (option D) is indeed a Snowflake feature but is not unique to the multi-cluster architecture. Enhanced security arrives via Snowflake's platform-wide mechanisms, not specifically because of multi-cluster architecture. Unstructured data loading (option B) is also not governed by multi-cluster behavior.

Question: 43

How can Snowsight be used to monitor the performance of a virtual warehouse?

- A. Use the SHOW WAREHOUSES command.
- B. Use the LOAD_HISTORY view.
- C. Use the DESCRIBE WAREHOUSE command.
- D. Use the QUERY_HISTORY view.

Answer: D

Explanation:

To monitor virtual warehouse performance in Snowsight, the correct approach is using the QUERY_HISTORY view or its Snowsight equivalent under the Activity section. QUERY_HISTORY provides detailed metrics on execution time, queueing, credit consumption, compilation time, and warehouse utilization for every executed query. By analyzing this information, users can diagnose issues such as concurrency bottlenecks, under-sized warehouses, long-running queries, or excessive I/O.

SHOW WAREHOUSES returns metadata about warehouse configurations (size, state, scaling policy) but does not provide performance data. LOAD_HISTORY tracks COPY INTO operations only and is not representative of general warehouse workload. DESCRIBE WAREHOUSE shows configuration details but no runtime or historical performance metrics.

Thus, the most accurate and complete way to assess warehouse performance—particularly execution delays, queueing, or excessive credit usage—is by using QUERY_HISTORY.

Question: 44

Which column is returned when the FLATTEN table function is executed?

- A. VALUE
- B. OUTPUT
- C. ROWNUM
- D. LEVEL

Answer: A

Explanation:

The FLATTEN table function expands semi-structured data such as arrays or objects stored in VARIANT columns. Its output includes several metadata columns, and the most important among them is VALUE, which contains the extracted element from the array or object. VALUE is the column that developers typically join on or reference to access the flattened data.

In addition to VALUE, FLATTEN returns KEY (for object keys), INDEX (position in arrays), SEQ (internal ordering), and PATH (hierarchical reference path). These other columns support analytical or transformation logic, but VALUE remains the primary output used for downstream SQL.

The other listed options—OUTPUT, ROWNUM, and LEVEL—are not part of the FLATTEN function's output schema.

ROWNUM and LEVEL may appear in other SQL contexts but not in FLATTEN results. Therefore, the correct answer is VALUE.

Question: 45

What are virtual warehouses in Snowflake?

- A. A compute resource
- B. A worksheet for writing queries
- C. A type of user account
- D. A storage location for data files

Answer: A

Explanation:

A virtual warehouse in Snowflake is a compute resource that supplies CPU, memory, and temporary

storage used to process SQL queries, perform data loading, execute DML operations, and run procedural logic.

Warehouses are elastic, can be resized on demand, suspended to save costs, and automatically resumed when needed.

Virtual warehouses don't store data—data is kept in Snowflake's centralized storage layer. They also are not user accounts; user accounts belong to Snowflake's identity management system.

Worksheets are UI components in Snowsight or Classic Console used to write and run SQL, not compute resources.

Thus, virtual warehouses are Snowflake's compute engine, enabling all processing without managing physical infrastructure.

Question: 46

What types of worksheets can be created in Snowsight? (Select TWO).

- A. SQL
- B. Javascript
- C. Scala
- D. Java
- E. Python

Answer: A,E

Explanation:

Snowsight supports two worksheet types: SQL worksheets and Python worksheets. SQL worksheets allow users to execute queries, create objects, and perform data analysis using ANSI SQL and Snowflake-specific extensions. Python worksheets, powered by Snowpark, allow users to write Python code that interacts directly with Snowflake tables, data frames, and machine learning workflows.

Java, Scala, and JavaScript are supported via Snowpark APIs or UDF development, but they cannot be used as worksheet languages. Worksheets are designed for interactive analysis, visualization, and iterative development, with native runtimes only for SQL and Python.

Thus, only SQL and Python worksheets can be created within Snowsight.

Question: 47

What is the default Time Travel data retention period?

- A. 45 days
- B. 1 day
- C. 7 days
- D. 90 days

Answer: B

Explanation:

The default Time Travel retention period for most objects in Snowflake is 1 day (24 hours). Time Travel enables access to historical versions of data after updates, deletes, or drops. It also allows cloning databases, schemas, and tables at previous points in time, and recovering dropped objects.

While Snowflake Enterprise Edition and higher tiers allow retention periods up to 90 days, this extended window is not the default—administrators must explicitly configure it for each table, schema, or database.

Retention periods of 7, 45, or 90 days are possible only with higher service editions; the default for all accounts and objects remains 1 day unless explicitly overridden.

Question: 48

When loading data into a table using the COPY INTO command, what is the default error-handling setting?

- A. ABORT_STATEMENT
- B. SKIP_FILE
- C. VALIDATE
- D. CONTINUE

Answer: A

Explanation:

By default, Snowflake uses `ON_ERROR = ABORT_STATEMENT` for `COPY INTO`. This means that if a single row fails to load due to issues such as type mismatches or malformed data, Snowflake immediately stops the entire load operation. This behavior ensures data integrity by preventing partial or incomplete loads when errors occur.

Other error-handling modes must be explicitly specified: `CONTINUE` skips erroneous rows, `SKIP_FILE` skips whole files with errors, and `VALIDATE` is part of the validation mode feature, not a default `COPY` behavior.

Thus, unless explicitly changed, `COPY INTO` always uses `ABORT_STATEMENT`.

Question: 49

What Snowflake parameter is configured in the Query Processing layer?

- A. The minimum and maximum serverless compute limits
- B. The types of tables available in an account
- C. The sizing of virtual warehouses
- D. The minimum and maximum micro-partition limits

Answer: C

Explanation:

The Query Processing layer of Snowflake is where virtual warehouses operate, so warehouse sizing parameters (X-Small to 6X-Large) fall under this layer. Warehouse size determines compute power, concurrency, and performance behavior for SQL workloads. Administrators configure warehouse size based on workload intensity, response time requirements, and cost considerations.

Serverless compute limits and micro-partition limits belong to storage and services layers. Table types (permanent, transient, temporary) are storage-level configurations, not part of Query Processing.

Thus, warehouse sizing is the correct parameter configured at the Query Processing layer.

Question: 50

What is the primary benefit of the Snowflake data cloud?

- A. It eliminates the need for data governance.

- B. It provides direct access to underlying infrastructure.
- C. It replaces traditional data warehouses with on-premises solutions.
- D. It enables organizations to unite and share their data.

Answer: D

Explanation:

The Snowflake Data Cloud allows organizations to seamlessly share, access, and collaborate on data across departments and external partners, without copying or moving data. Through secure data sharing, listings, and data clean rooms, Snowflake eliminates data silos and dramatically improves data collaboration.

It does not eliminate the need for governance—Snowflake enhances governance via RBAC, masking policies, and centralized controls. It does not provide access to underlying cloud infrastructure;

Snowflake abstracts that. It is not an on-premises solution; Snowflake is fully cloud-native.

Thus, the primary benefit is unifying and securely sharing data across the ecosystem.

Question: 51

What is the recommended way to provide a role with access to a Snowflake object?

- A. Transfer ownership of the object to the role.
- B. Assign the PUBLIC role.
- C. Grant privileges to the role.
- D. Clone the object and grant the role privileges to the cloned object.

Answer: C

Explanation:

The recommended approach in Snowflake's RBAC model is to grant privileges to roles, not to individual users or through ownership transfers. Privileges such as SELECT, INSERT, USAGE, and MODIFY are granted to roles, and roles are then assigned to users—creating scalable governance.

Transferring ownership is a powerful action and should not be used simply to provide access. The PUBLIC role should not be used to grant sensitive privileges since it is granted to all users by default. Cloning objects is unnecessary and creates new objects, not access to the original.

Thus, granting privileges to the role is the correct RBAC practice.

Question: 52

Which role is a system defined role in Snowflake?

- A. USERADMIN
- B. SNOWFLAKE_ADMIN
- C. SNOWFLAKE_DBA
- D. DATA_ENGINEER

Answer: A

Explanation:

USERADMIN is one of Snowflake's system-defined roles, created automatically in every account. It is responsible for managing users and roles, including CREATE USER, ALTER USER, and role assignment. It is part of Snowflake's default RBAC hierarchy (SYSADMIN, SECURITYADMIN, USERADMIN, etc.).

SNOWFLAKE_ADMIN and SNOWFLAKE_DBA are not Snowflake system roles—they may exist in organizations as custom roles but do not appear by default. DATA_ENGINEER is also user-created and not a built-in role.

Therefore, USERADMIN is the only true system-defined role listed.

Question: 53

What are characteristics of schemas in Snowflake? (Select TWO).

- A. Each schema is available to a single role.
- B. A schema is an immutable object.
- C. Schemas contain virtual warehouses.
- D. Schemas are used to organize database objects.
- E. A schema is contained in a database.

Answer: D,E

Explanation:

Schemas in Snowflake serve as organizational containers for database objects such as tables, views, sequences, stages, file formats, and functions. They exist within a database, forming Snowflake's **logical hierarchy**: account → database → schema → object.

Schemas are not limited to a single role; access depends on granted privileges. Schemas are not immutable—administrators can rename, alter, or drop them. Schemas do not contain virtual warehouses; warehouses exist independently at the account level.

Thus, the correct characteristics are that schemas organize objects and are contained within a **database**.

Question: 54

What is a database in Snowflake?

- A. A logical grouping of schemas.
- B. A single virtual warehouse.
- C. A physical storage location for data files.
- D. A collection of tables and views.

Answer: A

Explanation:

A Snowflake database is a logical container grouping one or more schemas. It does not store data physically in the traditional sense—Snowflake abstracts physical storage into its cloud-native micropartitioned storage layer. Because schemas contain tables, views, and other objects, a database organizes schemas logically, enabling governance, naming conventions, and security boundaries.

A virtual warehouse is compute, not storage. A “collection of tables and views” better describes a schema, not a database. Snowflake separates logical structure from physical storage, so a database is **not** a physical file repository.

Thus, a database is best defined as a logical grouping of schemas.

Question: 55

Which SQL keyword is used to constrain the number of rows returned by a query in Snowflake? (Select TWO)

- A. TOP
- B. SELECT

C. GET

D. LIMIT

E. ROW_NUMBER

Answer: A,D

Explanation:

Snowflake supports both TOP and LIMIT as valid SQL mechanisms for restricting the number of rows returned by a query. They function similarly but are used in different syntactic positions.

LIMIT appears at the end of the query:

```
SELECT * FROM my_table LIMIT 10;
```

This tells Snowflake to return only the specified number of rows.

TOP is used in the SELECT clause:

```
SELECT TOP 10 * FROM my_table;
```

Both approaches are equivalent in functionality and are supported to maintain compatibility with ANSI SQL and other SQL dialects.

The other options are not correct row-limiting mechanisms:

- SELECT is a required keyword but does not limit rows on its own.
- GET is not a SQL keyword for row restriction; GET is used for downloading files from internal stages.
- ROW_NUMBER is a window function used for ranking rows, not constraining row count.

Therefore, TOP and LIMIT are the only correct SQL keywords in Snowflake for controlling how many rows are returned by a query.

Question: 56

What are characteristics of the Snowflake Platform? (Select TWO).

A. Snowflake supports column-level security not row-level security.

B. Snowflake is responsible for all data security.

- C. Snowflake handles platform maintenance, management, and upgrades.
- D. There is no infrastructure to provision and maintain.
- E. Snowflake can run on private infrastructures on-premises.

Answer: C,D

Explanation:

Snowflake is a fully managed, cloud-native platform that automates nearly all infrastructure tasks. Two defining characteristics are:

1. Snowflake handles platform maintenance, management, and upgrades.

Snowflake automatically performs software updates, scaling operations, tuning, optimization, and patching. Users never need to upgrade hardware, install software, or manage clusters.

2. There is no infrastructure for customers to provision and maintain.

All compute, storage, and services run on public clouds (AWS, Azure, GCP). Snowflake abstracts the underlying infrastructure completely, eliminating the need for customers to manage VMs, disks, clusters, or networking.

Incorrect options:

- Snowflake supports both column-level security (via masking policies) and row-level security (via row access policies).
- Snowflake is not responsible for all data security—customers must configure RBAC, masking, and governance.
- Snowflake does not run on-premises; it is exclusively cloud-based.

Thus, options C and D correctly describe Snowflake platform characteristics.

Question: 57

What will be returned when a user runs this command?

```
SELECT count FROM my_table LIMIT 10;
```

- A. A non-deterministic set of 10 values in the count column.
- B. The highest 10 values in the count column.
- C. All values in the count column that are less than or equal to 10.

D. The lowest 10 values in the count column.

Answer: A

Explanation:

When a LIMIT clause is applied without an ORDER BY, Snowflake does not guarantee which rows will be returned. This means the result of:

```
SELECT count FROM my_table LIMIT 10;
```

is a non-deterministic set of 10 rows. Because Snowflake stores data in micro-partitions and distributes processing across compute resources, the natural order of rows is undefined unless explicitly controlled.

To retrieve the lowest or highest values, users must specify ORDER BY count ASC or ORDER BY count DESC respectively. Without ordering, Snowflake simply returns any 10 qualifying rows.

- Option B (highest values) requires ORDER BY count DESC.
- Option D (lowest values) requires ORDER BY count ASC.
- Option C (values ≤ 10) requires a WHERE clause (WHERE count ≤ 10).

Therefore, the query returns a non-predictable sample of 10 rows from the column, making option A correct.

Question: 58

How are privileges inherited in the Snowflake role hierarchy?

- A. Privileges are only inherited by direct child roles in the hierarchy.
- B. Privileges are inherited by any roles at the same level in the hierarchy.
- C. Privileges are inherited by any roles above a given role in the hierarchy.
- D. Privileges are only inherited by the direct parent role in the hierarchy.

Answer: C

Explanation:

Snowflake uses a hierarchical Role-Based Access Control (RBAC) model. When Role A is granted to Role B, the higher-level role (Role B) inherits all privileges of the lower-level role (Role A). This upward inheritance allows broad administrative roles (such as SYSADMIN) to automatically inherit permissions granted to subordinate roles.

Key principle: privilege inheritance flows upward, not downward.

- Parent roles inherit from child roles.
- Child roles do not inherit from parent roles.

Sibling roles do not share privileges unless explicitly granted.

Thus, roles above a given role in the hierarchy inherit privileges, making option C correct.

Options A and D incorrectly describe downward inheritance, which Snowflake does not perform.

Option B (same-level inheritance) also does not occur unless privileges are explicitly granted.

This hierarchical model simplifies governance by allowing higher-level administrative roles to centrally manage access across multiple subordinate roles.

Question: 59

What is the purpose of Time Travel?

- A. To automatically manage timestamp data types
- B. To ensure that users' data can be recovered at any time
- C. To facilitate the loading of historical data into Snowflake
- D. To allow users to access historical data

Answer: D

Explanation:

Time Travel enables Snowflake users to query, clone, or restore historical versions of data. This includes retrieving previous states of tables, schemas, or databases—even after updates, deletes, or drops. Time Travel operates within a retention period (default: 1 day, up to 90 days on higher editions).

Users can query historical data using the AS OF or BEFORE clause, restore dropped objects, and clone databases at

specific points in time for backup or analysis.

Time Travel does not automatically manage timestamp data types. It does not guarantee indefinite recovery—after the retention window expires, data moves into Fail-safe. It also is not primarily designed for loading historical datasets; its purpose is to access past states of Snowflake-managed data.

Thus, the correct purpose is to enable access to historical data inside Snowflake.

Question: 60

What is Snowflake's primary security capability?

- A. Automated data masking
- B. Network perimeter defense
- C. Role-based access control (RBAC)
- D. End-to-end encryption for all data, in transit and at rest

Answer: D

Explanation:

Snowflake's foundational security capability is always-on end-to-end encryption, applied to all customer data both at rest and in transit. Snowflake encrypts data stored in micro-partitions using AES-256, and all network communication uses TLS encryption. Key management is automated through Snowflake's hierarchical key model, providing additional rotation, rekeying, and defense-in-depth mechanisms.

While RBAC, masking policies, and network policies are crucial components of Snowflake security, encryption is the platform's guaranteed, baseline, mandatory protection feature. It applies universally to all data, regardless of workload, role configuration, or user settings.

Option A: automated masking is optional and policy-driven, not a primary universal mechanism.

Option B: network perimeter defense exists but is not the platform's core foundation.

Option C: RBAC controls access but does not protect data itself.

Thus, Snowflake's primary and most universal security capability is end-to-end encryption.

Question: 61

Changing the size of a virtual warehouse from a Medium to a Large refers to what type of warehouse tuning?

- A. Scaling down
- B. Scaling in
- C. Scaling out
- D. Scaling up

Answer: D

Explanation:

Changing a Snowflake virtual warehouse from Medium to Large increases the compute resources allocated to that warehouse—such as CPU cores, memory, and I/O bandwidth. This process is called scaling up, or vertical scaling. It is typically performed to improve performance for complex queries, large joins, or computationally heavy workloads.

Scaling down is the reverse—reducing warehouse size from Large to Medium or similar.

Scaling out refers to adding more clusters to a multi-cluster warehouse, enabling increased concurrency rather than increasing power per cluster.

Scaling in is the reverse of scaling out—reducing the number of active clusters.

Thus, increasing the size from Medium to Large is vertical scaling, also known as scaling up.

Question: 62

What command can be used to load data from an external stage to a table?

- A. INSERT INTO
- B. GET
- C. COPY INTO
- D. PUT

Answer: C

Explanation:

The COPY INTO command is Snowflake's primary bulk-loading mechanism for ingesting data from stages (internal or external) into tables. It can read from Amazon S3, Azure Blob Storage, Google Cloud Storage, or internal Snowflake stages. COPY INTO supports file formats (CSV, JSON, Parquet), pattern matching, error handling, transformations, and validation modes.

PUT uploads files from a local machine to an internal stage—not from stage to table.

GET downloads files from an internal stage to a local system.

INSERT INTO is for row-by-row inserts and is not designed for large file ingestion.

Therefore, COPY INTO is the correct and recommended method for loading stage files into Snowflake tables.

Question: 63

In the Snowsight Query History, what attribute uniquely identifies each query?

- A. Transaction ID
- B. Query ID
- C. Account ID
- D. Session ID

Answer: B

Explanation:

Every query executed in Snowflake is assigned a unique Query ID, a system-generated identifier that allows users to retrieve, analyze, and reference the query. Query IDs support debugging, profile retrieval, audit tracking, workload analysis, and support cases.

Transaction IDs are only assigned to transactional operations and do not uniquely identify all queries.

Session IDs identify user sessions—not individual queries. Many queries can occur in the same session.

Account ID is the identifier for the entire Snowflake account and does not relate to specific queries.

Thus, Query ID is the unique attribute used to identify each query in Query History.

Question: 64

What is a share?

- A. A Snowflake object that can only be accessed by using a reader account.
- B. A Snowflake object that holds all of the information required to share a database.
- C. A Snowflake object that holds all the information required to replicate data into a secondary account.
- D. A Snowflake object that holds the metrics about table data that has been shared outside of Snowflake.

Answer: B

Explanation:

A share in Snowflake is a securable object that contains all the metadata required to provide access to selected data—typically tables, views, and secure views—to other Snowflake accounts. The share object defines which objects are included and what privileges (e.g., SELECT, USAGE) consumers receive.

A share does not store data itself; it provides a reference to the provider's underlying micropartitioned storage, enabling real-time, zero-copy access.

Reader accounts (option A) can consume shares but do not define them.

Replication objects (option C) relate to database replication, not data sharing.

Option D is incorrect because Snowflake does not generate share-specific metrics objects.

Thus, the purpose of a share is to encapsulate all metadata required for secure, controlled data sharing between Snowflake accounts.

Question: 65

What are the three key layers of Snowflake's architecture?

- A. Data Repository, Compute Clusters, Metadata Management
- B. Authentication, Access Control, Cloud Services
- C. Database Storage, ODBC Drivers, Virtual Warehouses
- D. Data Storage, Query Processing, Cloud Services

Answer: D

Explanation:

Snowflake's architecture is built on three primary layers that separate and optimize storage, compute, and services. The Data Storage layer stores all structured and semi-structured data in Snowflake's optimized, compressed columnar format. Snowflake manages file organization, metadata, and optimizations automatically, ensuring efficient storage management. The Query Processing layer is implemented through virtual warehouses that independently execute queries using MPP (massively parallel processing) compute clusters. These warehouses scale elastically based on workload requirements. The Cloud Services layer manages metadata, authentication, access control, query optimization, and infrastructure orchestration. This separation allows independent scaling of compute and storage, optimizing performance while simplifying administration.

Question: 66

How does Snowflake process queries?

- A. With shared-disk architecture
- B. Using MPP compute clusters
- C. By optimizing data in cloud storage
- D. Through third-party connectors

Answer: B

Explanation:

Snowflake processes queries using Massively Parallel Processing (MPP) compute clusters, deployed as virtual warehouses. Each warehouse consists of multiple compute nodes working in parallel to execute queries efficiently. When a query is submitted, Snowflake distributes tasks across nodes, processes data subsets concurrently, and aggregates results. This architecture enables high performance, scalability, and the ability to handle complex analytical workloads. While Snowflake does incorporate elements of shared-disk storage, query execution itself depends on MPP compute clusters. Options such as third-party connectors or storage optimization do not represent the core query processing mechanism.

Question: 67

Which cloud platforms does Snowflake support?

- A. Amazon Web Services (AWS)
- B. Google Cloud Platform (GCP)
- C. Microsoft Azure
- D. All of the options

Answer: D

Explanation:

Snowflake is a fully cloud-native platform that supports deployment across all three major cloud providers: AWS, GCP, and Azure. Snowflake delivers consistent functionality across each provider, allowing organizations to choose the cloud best suited for regulatory, architectural, and business requirements. Although small feature differences may exist between cloud providers, Snowflake's core capabilities—virtual warehouses, storage layers, security, data sharing, and governance—operate uniformly across all three platforms. This multi-cloud support gives Snowflake strong flexibility for hybrid, multi-region, and multi-cloud deployments.

Question: 68

What type of storage format does Snowflake use for its data?

- A. Row-based format
- B. Compressed columnar format
- C. Compressed row-store format
- D. Uncompressed row-store format

Answer: B

Explanation:

Snowflake stores data in a compressed columnar format, which is optimal for analytical workloads. Columnar storage allows Snowflake to read only the relevant columns needed for query execution, improving performance significantly for aggregation and filtering operations. Compression further reduces storage consumption, enabling cost efficiency. This approach differs from row-store formats used in OLTP systems and provides better analytic processing. Snowflake automatically manages partitioning, micro-partitioning, and metadata without user intervention.

Question: 69

What does Snowflake UI called?

- A. Snowsite
- B. SnowGUI
- C. SnowView
- D. Snowsight

Answer: D

Explanation:

The official Snowflake web interface is called Snowsight. Snowsight is the modern, browser-based UI that enables SQL development, Python worksheets, dashboarding, visual exploration, and account monitoring. It replaces the classic UI and is now the default interface. Snowsight provides integrated capabilities such as notebooks, data loading interfaces, dashboards, role and warehouse management, and visual query history. Other options like Snowsite, SnowGUI, and SnowView are not valid Snowflake product names.

Question: 70

Which options are used in Snowflake notebooks for querying data? (Choose any 3 options)

- A. Python

- B. SQL
- C. Markup
- D. Markdown

Answer: A,B,D

Explanation:

Snowflake Notebooks support multiple cell types for development and documentation. Python cells allow users to perform data transformations, analysis, and Snowpark operations. SQL cells enable executing Snowflake SQL queries directly against tables and views. Markdown cells support formatted documentation within notebooks—useful for narrative explanations and commentary. The option “Markup” is not a supported cell type in Snowflake Notebooks. These three supported types allow users to blend analytics, code execution, and documentation in a unified environment.

Question: 71

What is a key feature of Snowflake notebooks?

- A. Command-line interface only
- B. Text-only data display
- C. Integrated environment for writing queries and visualizing results
- D. Manual hardware configuration

Answer: C

Explanation:

Snowflake Notebooks provide an integrated, interactive development environment that enables users to write SQL and Python code, visualize results, and document analysis—all in the same workspace. They support data visualizations, charts, tables, and Streamlit-based interactions. Notebooks run entirely within Snowsight and require no manual hardware configuration. They are not CLI-based and support much richer displays than plain text. This integration streamlines the entire data exploration, transformation, and modeling workflow.

Question: 72

What options are available under the settings for Python worksheets? (Choose any 2 options)

- A. Handler
- B. Return Value
- C. Return Type
- D. Output

Answer: A,C

Explanation:

In Snowflake Python worksheets, settings include the Handler and Return Type. The Handler identifies the specific Python function that serves as the entry point for execution—essential when deploying code as stored procedures or user-defined functions. The Return Type defines how Snowflake interprets the output, such as STRING, VARIANT, or TABLE, ensuring proper result rendering. “Return Value” is not a worksheet setting but rather the outcome produced by the code itself, and “Output” is not a configuration option.

Question: 73

What does a red checkmark next to a cell in a Snowflake notebook signify?

- A. The cell has errors
- B. The cell is currently running
- C. The cell has completed execution successfully
- D. The cell is paused

Answer: A

Explanation:

A red checkmark in Snowflake Notebooks indicates that the cell encountered an error during execution. This visual indicator helps users quickly identify issues so they can inspect the error message and adjust code accordingly. Successful execution shows a green checkmark, while running cells show a spinner. Snowflake notebooks do not include a “paused” state for notebook cells.

Question: 74

How can users integrate Streamlit with Snowflake notebooks?

- A. By writing HTML code
- B. By creating manual scripts
- C. By configuring network settings
- D. By using Streamlit functions within Python code in the notebook

Answer: D

Explanation:

Snowflake Notebooks provide native integration with Streamlit, allowing users to build interactive data apps directly inside Snowflake. Users can call Streamlit functions such as `st.write()`, `st.dataframe()`, `st.button()`, and `st.slider()` within Python cells. No external HTML, scripting, or network configuration is required. Streamlit components execute seamlessly in the notebook environment, enabling dashboards and interactive visualizations on top of Snowflake data.

Question: 75

Which option is available in Snowsight for loading data from a local file?

- A. Drag-and-drop interface
- B. Command-line input
- C. Manual data entry
- D. Only supports cloud storage

Answer: A

Explanation:

Snowsight provides a drag-and-drop interface for uploading data from a local machine into an internal stage. This is done via: Data → Add Data → Load Files into a Stage. Users can upload multiple files and choose the destination stage. This convenient UI eliminates the need for command-line tools such as SnowSQL. Snowsight supports both local file uploads and cloud storage references, not just cloud sources.

Question: 76

How can users filter queries in the Query History view in Snowsight?

- A. By network configuration
- B. By hardware settings
- C. By query status, date range, and user
- D. By text-based search only

Answer: C

Explanation:

Snowsight's Query History view allows filtering queries by status, date range, and executing user. Additional filters include SQL text search, warehouse, duration, statement type, session ID, query ID, query tag, parameterized query hash, and whether a query was generated by a client or task. These filters help users identify long-running, failed, queued, or specific types of queries for troubleshooting and performance optimization. Network or hardware configurations have no role in Query History filtering.

Question: 77

What are the steps to create a new stage in Snowsight?

- A. Data >> Create Schema
- B. Data >> add data >> Create Schema
- C. Data Product >> Create Schema
- D. Data >> Database >> Schema >> Create Stage

Answer: D

Explanation:

To create a new stage in Snowsight, users navigate through: Data → Database → Schema → Create Stage. Within the schema, selecting the "+" icon opens the option to create a stage. Users may configure internal or external stages, encryption settings, and directory table options. Other paths listed reference creating schemas rather than creating stages and are therefore incorrect.

Question: 78

Which are the type of Views we have in Snowflake? (Choose any 3 options)

- A. Materialized
- B. Non-Materialized
- C. UnMaterialized
- D. Secure

Answer: A,B,D

Explanation:

Snowflake supports Non-Materialized Views (standard views), Materialized Views, and Secure Views. Non-materialized views compute results dynamically at query time. Materialized views store precomputed results, improving performance for repeated analytical queries. Secure views protect underlying query logic and metadata from being exposed, ensuring privacy when sharing datasets. The option "UnMaterialized" is not a Snowflake concept and does not exist as a view type.

Question: 79

What a Database in Snowflake cant contain?

- A. Views
- B. Tables
- C. Role
- D. Database Role

Answer: C

Explanation:

A Snowflake database is a top-level container that can include schemas, and within those schemas it can contain objects such as tables, views, stages, functions, procedures, and database roles.

However, global account-level roles (such as SYSADMIN, SECURITYADMIN, PUBLIC, or custom account roles) do not reside inside a database. These roles are created and managed at the Snowflake account level, not within any specific database.

In contrast, database roles are scoped within a specific database and can exist inside it. Views and tables are standard database objects. Therefore, the only option that a Snowflake database cannot contain is an account-level Role.

Question: 80

How do databases and schemas fit into Snowflake's hierarchy? (Choose any 2 options)

- A. A database can contain multiple schemas
- B. A schema can contain multiple databases
- C. A schema can contain multiple tables and views
- D. A table can contain multiple schemas

Answer: A,C

Explanation:

Snowflake's object hierarchy begins with the account, which contains one or more databases. Each database can contain multiple schemas, and each schema contains objects such as tables, views, stages, and functions. The schema is a logical boundary for grouping related database objects. A schema cannot contain databases, and tables cannot contain

schemas. Instead, tables and views are contained within schemas. Therefore, the two correct hierarchical relationships are that a database can contain multiple schemas, and a schema can contain multiple tables and views.

Question: 81

Which of the following are numeric data types supported by Snowflake? (Choose any 2 options)

- A. INTEGER
- B. BOOLEAN
- C. VARCHAR
- D. FLOAT

Answer: A,D

Explanation:

Snowflake supports several numeric data types, including INTEGER (and its aliases INT, BIGINT, SMALLINT, TINYINT) and FLOAT (a floating-point type). INTEGER represents exact whole numbers, and FLOAT supports approximate numerical values with fractional components. BOOLEAN is a logical data type used for TRUE/FALSE values, not numeric. VARCHAR is a string data type, also not numeric. Thus, INTEGER and FLOAT are the correct numeric data types supported by Snowflake.

Question: 82

Which of the following are types of tables in Snowflake? (Choose any 3 options)

- A. Permanent
- B. Temporary
- C. Transient
- D. Virtual

Answer: A,B,C

Explanation:

Snowflake provides three main table types: Permanent, Transient, and Temporary. Permanent tables support full Time Travel and Fail-safe, and are designed for long-term storage. Transient tables do not include Fail-safe and are best suited for intermediate or less durable datasets. Temporary tables exist only for the duration of the user session that created them, making them ideal for staging or ephemeral workloads. Snowflake does not have a table type called "Virtual"; therefore, the correct options are Permanent, Temporary, and Transient.

Question: 83

What most accurately describe Snowflake?

- A. A data warehouse software
- B. An ETL Tool
- C. An AI data cloud Platform
- D. A Transactional database

Answer: C

Explanation:

Snowflake is best described as an AI Data Cloud Platform. While Snowflake originated as a cloud data warehouse, its capabilities have expanded to include secure data sharing, application development, machine learning workloads, unstructured data processing, Snowpark, and integrated LLM/AI functions (such as Snowflake Cortex and Streamlit). This broad ecosystem supports analytic workloads, governance, multi-cloud architecture, and AI-driven applications. It is not simply an ETL tool or a transactional database.

Question: 84

Which SQL command is used to create a new database in Snowflake?

- A. USE DATABASE database_name;
- B. CREATE DATABASE database_name;
- C. CREATE DB database_name;
- D. USE DB database_name;

Answer: B

Explanation:

The correct SQL command for creating a new database in Snowflake is `CREATE DATABASE database_name;`. This initializes a new logical container for schemas and objects. `USE DATABASE` switches the active database context but does not create one. Snowflake does not support the shorthand `CREATE DB` or `USE DB` syntax. Therefore, `CREATE DATABASE` is the correct and only valid command for creating a database.

Question: 85

Which of the following are key components of Snowflake's RBAC system? (Choose any 3 options)

- A. Permissions
- B. Roles
- C. Privileges
- D. Users

Answer: B,C,D

Explanation:

Snowflake's RBAC security model revolves around `Users`, `Roles`, and `Privileges`. `Users` represent individual identities or service accounts. `Roles` group privileges and are assigned to users. `Privileges` define specific actions that can be performed on objects (e.g., `SELECT`, `INSERT`, `OWNERSHIP`). "Permissions" is not a formal RBAC component—permissions are effectively the result of privileges assigned to roles. Therefore, the three core RBAC components are `Users`, `Roles`, and `Privileges`.

Question: 86

What is the purpose of a role hierarchy in Snowflake?

- A. To define the sequence of SQL queries
- B. To organize roles and grant inherited privileges

C. To manage network settings

D. To store raw data

Answer: B

Explanation:

Role hierarchy in Snowflake allows one role to inherit the privileges of another. By granting roles to other roles, Snowflake enables scalable, maintainable access control. Higher-level roles grant privileges downward, allowing administrators to create layered access structures. This hierarchy simplifies permission management across teams and environments. It has no relation to SQL sequencing, network settings, or data storage.

Question: 87

What is a key feature of custom roles in Snowflake? (Choose any 2 options)

A. They cannot be granted to users

B. They can be tailored to specific organizational requirements

C. They have predefined privileges

D. They are created by Snowflake administrators

Answer: B,D

Explanation:

Custom roles in Snowflake allow administrators to design access patterns aligned with organizational security requirements. They are created by Snowflake administrators and do not have predefined privileges; instead, privileges must be granted explicitly. Custom roles can be assigned to users or other roles, just like system-defined roles. Thus, the key attributes are that they are created by administrators and tailored to meet specific organizational needs.

Question: 88

How can roles access objects in Snowflake? (Choose any 2 options)

A. By being directly assigned privileges on the objects

B. By inheriting privileges from parent roles

C. By configuring network settings

D. By storing raw data

Answer: A,B

Explanation:

Roles gain access to objects in Snowflake through direct privilege grants or through inheritance within a role hierarchy. A role may be explicitly granted privileges such as SELECT, OWNERSHIP, or USAGE. Alternatively, if a role is assigned to another role, it inherits all privileges of its parent role. Neither network settings nor data storage processes determine role access.

Question: 89

Which command is used to grant privileges to a role in Snowflake?

A. ASSIGN

B. CREATE

C. REVOKE

D. GRANT

Answer: D

Explanation:

The GRANT command is used in Snowflake to assign privileges to a role. GRANT allows administrators to authorize roles to perform actions on objects such as tables, warehouses, schemas, or databases. The REVOKE command removes privileges, CREATE creates objects, and ASSIGN is not a valid Snowflake command. Therefore, GRANT is the correct command for privilege assignment.

Question: 90

What is the primary purpose of the LIMIT clause in a Snowflake query?

A. To sort the result set

B. To filter the result set

C. To restrict the number of rows returned

D. To group the result set

Answer: C

Explanation:

The LIMIT clause limits the number of rows Snowflake returns from a query. It is commonly used to preview datasets, improve performance when exploring large tables, and paginate results. LIMIT may be combined with OFFSET to skip a number of rows. Sorting is performed with ORDER BY, filtering with WHERE, and grouping with GROUP BY—not with LIMIT. Therefore, restricting returned rows is the primary purpose of LIMIT.

Question: 91

To exclude certain columns from a SELECT query, you should:

A. Explicitly list the columns you want to include

B. Use the EXCLUDE keyword

C. Use a REMOVE function on the table

D. Use the OMIT clause

Answer: B

Explanation:

Snowflake supports the EXCLUDE keyword to simplify queries when excluding certain columns from a SELECT * operation. SELECT * EXCLUDE (column1, column2) reduces verbosity and enhances maintainability, especially when table schemas evolve. Explicitly listing all columns is possible but inefficient. Snowflake does not support REMOVE functions for columns nor an OMIT clause. EXCLUDE is the correct and official mechanism.

Question: 92

What is a fully qualified name in Snowflake used for?

A. To store data in tables

B. To manage user permissions

C. To configure network settings

D. To uniquely identify database objects

Answer: D

Explanation:

A fully qualified name uniquely identifies Snowflake objects by specifying database.schema.object.

This prevents ambiguity when multiple schemas or databases contain objects with identical names.

Fully qualified names ensure that SQL statements operate on the intended object. They are not used for storing data, managing permissions, or configuring network settings. Their core purpose is precise object identification.

Question: 93

What happens when you execute the following command in Snowflake?

```
DROP SCHEMA IF EXISTS marketing_schema;
```

- A. An error is returned if the schema does not exist
- B. The schema "marketing_schema" is created if it does not exist
- C. The schema "marketing_schema" is dropped only if it exists
- D. The schema "marketing_schema" is dropped even if it contains tables

Answer: C

Explanation:

The SQL command `DROP SCHEMA IF EXISTS marketing_schema;` instructs Snowflake to drop the specified schema only if it already exists in the database. The inclusion of the `IF EXISTS` clause prevents Snowflake from returning an error when the schema is not found. If the schema does exist, Snowflake drops it along with all objects it contains, unless a `RESTRICT` clause is enforced (`RESTRICT` is the default when not explicitly specified, but Snowflake also supports `CASCADE` for forcing deletion of all objects). Importantly, this command never creates a schema; it strictly performs a conditional drop. In addition, Snowflake supports Time Travel, so the dropped schema can typically be restored—using `UNDROP SCHEMA`—within the defined retention window. Therefore, the correct interpretation is that the schema is dropped only if it exists, and no error is thrown otherwise.

Question: 94

What is the purpose of the Public Schema in Snowflake? (Choose any 2 options)

- A. Storing user-specific data
- B. Providing a default schema for new objects
- C. Schema to store Account level meta data
- D. Storing shared resources accessible by all users

Answer: B,D

Explanation:

The Public schema is created by default within every Snowflake database and serves primarily as a shared workspace that is accessible to all users who have the necessary privileges. One of its main roles is to act as a default schema for objects created without explicitly specifying a schema name. This ensures usability and prevents object creation errors when

users forget to define a schema. Additionally, the Public schema is commonly used to store shared or common resources that multiple users or teams need to access. However, it should not be used for sensitive or user-specific data because access can be broad. It does not store account-level metadata; that information resides in INFORMATION_SCHEMA views or account-level metadata functions. Overall, the Public schema acts as both a convenient default location and a shared-access area within a Snowflake database.

Question: 95

What is the primary purpose of Snowflake's Cloning feature?

- A. To back up data in real-time
- B. To create an exact copy of a database, schema, or table without duplicating the storage
- C. To manage user permissions
- D. To create a backup of the entire Snowflake account

Answer: B

Explanation:

Snowflake's Cloning feature provides the ability to create a zero-copy clone of tables, schemas, or entire databases. This means that the cloned object references the same underlying micro-partition data as the original, so no duplicate storage costs are incurred upon creation. Instead, Snowflake uses metadata pointers to reference the existing data. If changes occur in the original or cloned objects, Snowflake performs copy-on-write operations to maintain isolation while still optimizing storage. This cloning mechanism is extremely valuable for development, testing environments, analytics validation, and troubleshooting historical states. It is not a real-time backup tool; backup and restore are handled through Time Travel and Fail-safe. Cloning does not grant or manage permissions nor copy the entire Snowflake account; it works solely at the database, schema, and table object levels.

Question: 96

Which of the following can be cloned using Snowflake's Cloning feature? (Choose any 3 options)

- A. Databases
- B. Schemas
- C. Roles

D. Tables

Answer: A,B,D

Explanation:

Snowflake supports zero-copy cloning at the database, schema, and table levels. Cloning creates a new metadata reference to existing data without physically duplicating micro-partitions, significantly reducing storage cost and clone creation time. Cloning a database replicates all schemas and tables within it at that moment in time. Cloning a schema replicates all objects inside that schema, including tables and views. Cloning a table creates an isolated version of that table for development, testing, or analysis. Importantly, roles cannot be cloned because they are account-level security objects, not data objects. Privileges must be manually granted or recreated for cloned objects. Cloning enables rapid, efficient creation of consistent environments for workload isolation and testing.

Question: 97

What is the maximum duration of data retention using Time Travel in Snowflake for standard accounts?

A. 0

B. 1

C. 7

D. 90

Answer: B

Explanation:

For Snowflake Standard Edition accounts, the maximum Time Travel retention period is 1 day (24 hours). Time Travel allows users to query historical data, restore dropped objects, and clone objects at a point in time. Although the retention period can be adjusted at the account, database, schema, or table level, Standard Edition restricts the upper limit to 1 day. A value of 0 disables Time Travel entirely. A 7-day retention period applies to Fail-safe, not Time Travel, and Fail-safe is a disaster recovery mechanism managed exclusively by Snowflake. The 90-day Time Travel retention is available only for Enterprise Edition and above. Therefore, the correct maximum retention period for Standard Edition is 1 day.

Question: 98

What is a "provider" in the context of the Snowflake Marketplace or Data Exchange?

- A. A user who only consumes data
- B. A user who manages Snowflake infrastructure
- C. A user or organization that shares data products or data listings
- D. A user who is a Snowflake employee

Answer: C

Explanation:

In Snowflake Marketplace and Data Exchange, a provider is any Snowflake user or organization that publishes and shares datasets, data products, or applications with other Snowflake consumers. Providers control access to their listings, configure data sharing models, and can offer either free or paid data products. They manage listing descriptions, update schedules, and refresh logic, ensuring customers or subscribers receive up-to-date and reliable information. Providers can be enterprises, analytics vendors, public data contributors, or research institutions. They are distinct from consumers, who access or use the shared data. Providers also do not handle Snowflake platform infrastructure—that is Snowflake's responsibility. Snowflake employees are not categorized as providers unless they publish official Snowflake datasets. Thus, the primary purpose of a provider is to share curated data offerings on the Marketplace or within private exchanges.

Question: 99

Which of the following is true about data listings in the Snowflake Marketplace?

- A. They always require payment for access.
- B. They are limited to static data snapshots.
- C. They can be free or paid, depending on the provider's terms.
- D. They are only accessible to Snowflake administrators.

Answer: C

Explanation:

Data listings in the Snowflake Marketplace can be offered as free or paid depending on how the provider chooses to distribute the data. Providers can use usage-based pricing, subscription models, or make datasets available at no cost. Listings are not restricted to static data; Snowflake supports live data sharing, enabling instant access to continuously updated datasets without copying or ETL. Listings are accessible to any Snowflake user with the appropriate marketplace access—not exclusively administrators. This flexibility makes the Marketplace a powerful ecosystem where organizations can seamlessly obtain external datasets for analytics, machine learning, and reporting needs. Providers define terms of use, pricing, and refresh schedules, while Snowflake manages the secure data-sharing infrastructure.

Question: 100

What is the key difference between a "private listing" and a "public listing" in a Snowflake Data Exchange context?

- A. Public listings require payment, while private listings are free.
- B. Public listings are only accessible to Snowflake employees, while private listings are for external users.
- C. Private listings are shared with specific Snowflake accounts, while public listings are available to all Snowflake users.
- D. Private listings are for sharing code, while public listings are for sharing data.

Answer: C

Explanation:

A private listing is shared only with explicitly selected Snowflake accounts, giving providers fine-grained control over who can access their data. This is ideal for partner organizations, internal business units, or customers requiring restricted access. A public listing, on the other hand, is accessible to all Snowflake users via the Marketplace, making it available for broad distribution. Pricing models for both types can be free or paid; neither category inherently requires payment. Both listing types share data, not code. Public listings are not restricted to Snowflake employees—they are open to the entire Snowflake ecosystem. The distinction lies entirely in access scope: targeted (private) vs. universal (public).

Question: 101

How is the initial activation of the Snowflake Data Exchange feature, which is necessary for establishing private exchanges, typically accomplished?

- A. Any Snowflake account administrator can enable Data Exchange through the web interface with a single click.
- B. Activation of the Data Exchange feature generally requires a request to Snowflake Support, who then enable it for the specific account.
- C. Data Exchange is automatically enabled for all Snowflake accounts upon creation.
- D. Activation of the Data Exchange feature requires a specific SQL command to be executed by the ACCOUNTADMIN role.

Answer: B

Explanation:

The Data Exchange feature is not automatically enabled for all Snowflake accounts. Instead, enabling this capability—particularly for private exchanges—typically requires submitting a request to Snowflake Support. Snowflake evaluates account type, region, and compliance requirements before enabling the Data Exchange functionality. Snowflake administrators cannot independently enable this through Snowsight, nor is there a SQL command available to turn it on. Once activated, account administrators can create private exchanges, manage listings, and define which Snowflake accounts can participate. This activation step ensures secure, intentional use of data-sharing capabilities at an account level.

Question: 102

What does "warehouse scaling in/out" refer to in Snowflake?

- A. Changing the size of the warehouse (e.g., from Small to Medium or vice versa).
- B. Moving data between different storage locations.
- C. Changing the region of the warehouse.
- D. Adjusting the number of clusters in a multi-cluster warehouse.

Answer: D

Explanation:

Scaling in/out in Snowflake refers to modifying the number of compute clusters associated with a multi-cluster virtual warehouse. Scaling out increases cluster count to accommodate higher concurrency or workload spikes, allowing more queries to run simultaneously without queuing. Scaling in reduces cluster count during periods of lower demand, optimizing compute usage and costs. This is distinct from scaling up/down, which refers to changing warehouse size (e.g.,

Small, Medium). Scaling does not involve data movement or region changes; warehouse compute is stateless and operates independently of storage. Multi-cluster warehouses allow Snowflake to automatically add or remove clusters based on demand when auto-scale policies are configured.

Question: 103

Which of the following are benefits of using a multi-clustered warehouse in Snowflake? (Choose any 3 options)

- A. Improved query performance during peak load
- B. Automatic scaling of clusters based on query demand
- C. Reduced storage costs
- D. Increased compute capacity

Answer: A,B,D

Explanation:

A multi-cluster warehouse raises Snowflake's capability to handle concurrent workloads efficiently. When large numbers of users or queries hit the system, Snowflake can automatically start additional clusters (scaling out) to distribute workloads, reducing queuing and improving performance. As query demand drops, Snowflake scales in by shutting down clusters to conserve credits. This dynamic auto-scaling provides significantly increased compute capacity during peak usage periods and enhances responsiveness. Importantly, multi-cluster warehouses affect only compute, not storage, so they do not reduce storage costs. Their primary benefits revolve around performance stability, concurrency handling, workload isolation, and compute elasticity.

Question: 104

Which command is used to create a new virtual warehouse in Snowflake?

- A. CREATE VIRTUALWAREHOUSE warehouse_name;
- B. CREATE DATABASE warehouse_name;
- C. CREATE SCHEMA warehouse_name;
- D. CREATE WAREHOUSE warehouse_name;

Answer: D

Explanation:

Snowflake uses the SQL command `CREATE WAREHOUSE warehouse_name` to create a new virtual warehouse.

Warehouses provide the compute resources for executing SQL queries, loading data, and

performing transformations. When creating a warehouse, additional parameters may be specified, such as warehouse size, auto-suspend timeout, auto-resume, scaling policy, or multi-cluster mode. The options involving “`CREATE VIRTUALWAREHOUSE`” are invalid syntax in Snowflake. `CREATE DATABASE` and `CREATE SCHEMA` create storage containers, not compute resources. Therefore, `CREATE WAREHOUSE` is the only correct command for provisioning compute.

Question: 105

What is the purpose of the "auto-suspend" feature in Snowflake virtual warehouses?

- A. To automatically increase the size of the warehouse during peak load
- B. To automatically shut down the warehouse after a period of inactivity
- C. To automatically add more nodes to the warehouse
- D. To automatically reduce the number of clusters

Answer: B

Explanation:

The auto-suspend feature saves compute costs by automatically suspending a warehouse when it becomes idle for a configured period. Since Snowflake charges for compute time while a warehouse is running, auto-suspend prevents unnecessary credit consumption by stopping the warehouse when no queries are executing. The warehouse can resume automatically when a new query is submitted, provided auto-resume is enabled. Auto-suspend does not change warehouse size (scaling up/down) nor adjust clusters in a multi-cluster warehouse (scaling in/out). It strictly controls when compute resources turn off due to inactivity, making it an essential cost-optimization feature.

Question: 106

Which of the following is true about Snowflake's "auto-scale" feature for multi-clustered warehouses? (Choose any

2 options)

- A. It automatically adjusts the number of clusters based on query demand
- B. It requires manual intervention to add or remove clusters
- C. It helps optimize resource utilization during variable workloads
- D. It is only available for standard warehouses

Answer: A,C

Explanation:

Auto-scale enables Snowflake to dynamically start or stop clusters in a multi-cluster warehouse depending on workload pressure. When concurrency rises, Snowflake adds clusters; when demand decreases, it scales back down. This provides seamless performance during peak workloads and efficient resource usage during off-peak times. Auto-scale does not require manual action—it is fully automated once configured. It is not limited to standard warehouses; multi-cluster mode is available in Enterprise Edition and above. Ultimately, auto-scale ensures elasticity and cost-efficient compute allocation.

Question: 107

Which types of stages are available in Snowflake? (Choose any 2 options)

- A. Internal
- B. External
- C. Public
- D. Private

Answer: A,B

Explanation:

Snowflake provides two main types of stages: Internal stages and External stages. Internal stages reside inside Snowflake and include user stages, table stages, and named internal stages. These offer convenient, secure storage within Snowflake's managed environment. External stages reference cloud storage locations such as AWS S3, Azure Blob

Storage, or Google Cloud Storage, enabling integration with external data repositories. Snowflake does not define a stage type called “Public” or “Private,” although stages can be granted secure access. Thus, the correct types are Internal and External stages.

Question: 108

Which of the following are examples of semi-structured data formats? (Choose any 3 options)

- A. JSON
- B. CSV
- C. Parquet
- D. XML

Answer: A,C,D

Explanation:

Semi-structured data refers to formats that do not follow a rigid relational schema but still contain structural tags or hierarchies, allowing flexible representation of nested or irregular data. In Snowflake, JSON, Parquet, and XML are all considered semi-structured because they incorporate metadata, hierarchical fields, or tagged data that can vary across records. JSON offers key–value pairs and nested arrays, making it ideal for logs and API responses. Parquet, being a columnar file format containing both schema definitions and metadata, is optimized for analytics while still supporting semi-structured capabilities such as variable field nesting. XML uses tags and attributes to represent hierarchical content, making it semi-structured as well. On the other hand, CSV represents strictly structured, row/column-based data without inherent metadata or hierarchy, so it is not considered semi-structured. Snowflake treats semi-structured formats by loading them into the VARIANT data type, enabling powerful SQL-based exploration using path notation.

Question: 109

How can you load JSON data into a Snowflake table from stage?

- A. Use the INSERT INTO command
- B. Use the COPY INTO command
- C. Use the LOAD DATA command
- D. Use the UPLOAD JSON command

Answer: B

Explanation:

Snowflake uses the COPY INTO <table> command as the primary mechanism for bulk loading semistructured data, including JSON, from internal or external stages. COPY INTO is optimized for large file ingestion and supports file format specifications, such as TYPE=JSON, enabling Snowflake to parse, validate, and store JSON into the VARIANT data type. Unlike INSERT INTO, which is suitable for row-level inserts, COPY INTO supports large-scale ingestion with parallelization and error-handling parameters. Commands such as LOAD DATA or UPLOAD JSON are not part of Snowflake's SQL interface. COPY INTO works seamlessly with both internal stages (USER, TABLE, or NAMED stages) and external cloud storage stages (S3, Azure Blob, GCS). Once the JSON is loaded, users can query nested fields using dot or bracket notation, flatten arrays for relational use, and build transformations directly in SQL. Therefore, the COPY INTO command is the correct and supported method for loading JSON files from a Snowflake stage.

Question: 110

What is the purpose of the PARSE_JSON function in Snowflake?

- A. Storing JSON data in its PARSE format
- B. Converting JSON data to XML format
- C. Loading JSON data from external stages
- D. Parsing JSON data into relational table format

Answer: D

Explanation:

The PARSE_JSON() function ingests a string containing JSON text and converts it into Snowflake's VARIANT data type, enabling the JSON to be queried, navigated, and transformed using SQL. Snowflake does not store JSON in its raw textual representation; instead, VARIANT allows Snowflake to apply optimized parsing, indexing, and querying operations against semi-structured content. This function is particularly useful when JSON arrives inline (e.g., supplied directly within SQL statements or loaded from CSV files containing JSON strings). PARSE_JSON does not perform data loading from stages—that is handled through COPY INTO—nor does it convert JSON into XML.

Once JSON is converted to VARIANT, Snowflake allows access to nested structures using dot notation, bracket notation, and functions like FLATTEN(). Thus, the function serves as a bridge between raw JSON strings and Snowflake's relational and analytical capabilities.

Question: 111

How can you query nested fields in semi-structured data in Snowflake? (Choose any 2 options)

- A. Using dot notation (.) to access nested fields
- B. Using Snowflake functions
- C. Using bracket notation [] to access nested fields
- D. Using Cloud Functions

Answer: A,C

Explanation:

Snowflake provides intuitive mechanisms to navigate semi-structured data stored in VARIANT columns, especially JSON, XML, Parquet, and Avro. Two primary access methods are dot notation and bracket notation. Dot notation, such as `column.nested.key`, allows straightforward access to nested elements using a hierarchical reference. Bracket notation, such as `column['key name']`, allows access to elements with special characters, spaces, or numeric array indices. While Snowflake also offers powerful functions such as `FLATTEN()`, `OBJECT_CONSTRUCT()`, and `ARRAY_LENGTH()`, these operate on semi-structured types but do not serve as direct field-access mechanisms. Cloud Functions play no role in querying semi-structured fields inside Snowflake. Dot and bracket notation work together to provide full expressiveness, enabling users to traverse complex hierarchical structures directly in SQL, making Snowflake an ideal platform for semi-structured data analytics.

Question: 112

Which of the following are examples of unstructured data? (Choose any 3 options)

- A. JSON
- B. Image

C. Video

D. Pdf

Answer: B,C,D

Explanation:

Unstructured data refers to content that lacks a predefined schema or consistent structural pattern. It cannot be represented naturally in tabular or hierarchical formats without extraction or transformation. Snowflake supports unstructured data through external tables and native features allowing storage and machine learning analyses. Images, videos, and PDFs fall into this category because they contain raw multimedia or document content without intrinsic row/column structure, often requiring OCR, video/audio processing, or content extraction. In contrast, JSON is semistructured, containing keys, hierarchical structures, and metadata that Snowflake's VARIANT type can interpret directly. Unstructured data is becoming increasingly important for AI-driven workloads, enabling use cases such as OCR with PARSE_DOCUMENT, deep-learning analysis of images and video, and document intelligence tasks.

Question: 113

When loading CSV data, what FILE FORMAT option allows you to skip the header row?

A. SKIP_FILE_HEADER = FALSE

B. IGNORE_HEADER = TRUE

C. HEADER_SKIP = 1

D. SKIP_HEADER = 1

Answer: D

Explanation:

Snowflake's file format options allow granular control over CSV ingestion. The SKIP_HEADER = 1 parameter instructs Snowflake to ignore the first line of the file, typically containing column headers. This is the correct and official parameter within FILE FORMAT objects or inline COPY INTO options. Other options listed—such as SKIP_FILE_HEADER, IGNORE_HEADER, or HEADER_SKIP—are not valid

Snowflake parameters. When SKIP_HEADER is applied, Snowflake begins reading data rows starting from line 2. This ensures proper alignment between file contents and table structure during ingestion. SKIP_HEADER can also be set to

values greater than 1 if multiple header rows exist. This behavior is essential when loading structured data where the header row is not intended for table consumption.

Question: 114

What information can be obtained by describing a table in Snowflake? (Choose any 3 options)

- A. Column names and data types
- B. Table constraints
- C. Table Origin
- D. Indexes and keys

Answer: A,B,D

Explanation:

The DESCRIBE TABLE (or DESC TABLE) command in Snowflake provides core metadata about table structure. This includes column names, data types, nullability, and default values. It also returns constraints if applied, such as PRIMARY KEY, UNIQUE, and CHECK constraints. Snowflake additionally displays information regarding clustering keys and other table characteristics. The command does not show table origin; Snowflake does not track lineage information directly through DESCRIBE TABLE. Indexes in Snowflake are not traditional B-tree indexes—Snowflake uses micropartition pruning instead—but DESCRIBE TABLE can show clustering keys, which function similarly by enabling optimized data skipping. Therefore, column definitions, constraints, and keys are all valid outputs of DESCRIBE TABLE.

Question: 115

Which command is synonymous with the LIST command in Snowflake?

- A. LS
- B. SHOW
- C. DESC
- D. LIS

Answer: A

Explanation:

Snowflake allows users to list files in a stage using the LIST command, and it provides a shorthand alias: LS. Both commands return identical results, including file name, size, last modified timestamp, and other metadata stored in the stage. This is relevant for internal stages (user, table, named stages) and external stages backed by cloud storage. SHOW and DESC are unrelated commands—SHOW lists Snowflake metadata objects such as tables, schemas, or roles, while DESC describes object structures. “LIS” is not a Snowflake command. Therefore, LS is the exact shorthand synonym for LIST and is used identically in queries.

Question: 116

Which of the following data format supported to unload semi-structured data from Snowflake? (Choose any 2 options)

- A. JSON
- B. CSV
- C. Parquet
- D. XML

Answer: A,C

Explanation:

Snowflake supports unloading semi-structured data specifically into JSON and Parquet formats. JSON preserves semi-structured hierarchy and nested fields, making it ideal for interchange, backup, and integration with downstream systems. Parquet provides columnar, compressed, and metadata-rich storage well-suited for analytical reprocessing in data lakes or other systems. CSV is not semistructured and cannot represent nested data accurately. XML, although a semi-structured format, is not supported by Snowflake as an unload format. When performing COPY INTO <location>, users can

specify FILE_FORMAT with TYPE=JSON or TYPE=PARQUET to export semi-structured content efficiently.

Question: 117

Which function would you use to determine the sentiment of a customer review in Snowflake Cortex?

- A. TRANSLATE
- B. COMPLETE
- C. SENTIMENT
- D. PARSE_DOCUMENT

Answer: C

Explanation:

The SENTIMENT function in Snowflake Cortex analyzes text and returns a numerical sentiment score between -1 and 1. This value indicates whether the text expresses negative, positive, or neutral sentiment. It accepts English-language text and uses Snowflake's managed AI models to infer polarity based on contextual understanding. A negative score represents negative emotion, a positive score represents positive emotion, and values near zero indicate neutrality. It is ideal for customer reviews, feedback analysis, and text mining. TRANSLATE is used for language translation, COMPLETE for text generation, and PARSE_DOCUMENT for extracting text from documents—not sentiment analysis. SENTIMENT provides fast, in-database AI inference without external model hosting or integration.

Question: 118

What is the purpose of the COMPLETE function in Snowflake Cortex?

- A. To parse documents
- B. To translate languages
- C. To classify text
- D. To generate text completions based on a given prompt

Answer: D

Explanation:

The COMPLETE function enables natural language generation within Snowflake Cortex. It accepts a prompt and produces

a text-based completion using an underlying large language model (LLM). COMPLETE can generate single-turn or multi-turn responses, depending on how the prompt is structured. It is used for tasks such as drafting summaries, generating recommendations, writing structured output such as JSON, or extending existing text. It is not a classification tool—that role is served by CLASSIFY_TEXT—and it does not perform linguistic translation (TRANSLATE) or document extraction (PARSE_DOCUMENT). COMPLETE allows developers and analysts to embed AI-driven generative capabilities directly within SQL pipelines or Snowpark processes, eliminating the need for external LLM services.

Question: 119

When using the TRANSLATE function, what type of input is expected for the source and target languages?

- A. Full language names (e.g., "English", "French")
- B. Standard language codes (e.g., "en", "fr")
- C. File paths to language dictionaries
- D. Numerical language identifiers

Answer: B

Explanation:

The TRANSLATE function in Snowflake Cortex expects ISO-standard language codes such as "en" for English or "fr" for French. These are two-letter or multi-letter codes defined by international standards. Snowflake uses these codes to instruct its translation model which languages to interpret and produce. If the source language is unknown, users can pass an empty string to trigger automatic language detection. Full-language names, dictionary file paths, or numeric identifiers are not

required or supported. TRANSLATE operates entirely with text input and language codes, ensuring easy integration into SQL workflows for multi-language data pipelines, localization tasks, and multilingual analytics.

Question: 120

Which of the following file formats can be processed by the PARSE_DOCUMENT function?

- A. PDF
- B. PPTX

C. XLSX

D. DOCX

Answer: A,B,D

Explanation:

PARSE_DOCUMENT extracts text and, depending on mode, layout information from supported document formats. The function currently supports PDF, PPTX, and DOCX files. For PDFs, it can perform OCR to extract scanned text or use LAYOUT mode to detect tables and structured regions. For PPTX files, it extracts slide text and content. For DOCX files, the function captures text and applies layout analysis. XLSX files are not supported—Excel content requires different extraction methods.

PARSE_DOCUMENT enables automated document intelligence workflows entirely inside Snowflake, eliminating the need for external OCR or parsing services.

Question: 121

What type of mode available in PARSE_DOCUMENT function? (choose two)

A. OCR

B. OMR

C. CONTENT

D. LAYOUT

Answer: A,D

Explanation:

PARSE_DOCUMENT supports two processing modes: OCR and LAYOUT. OCR mode performs Optical Character Recognition, extracting raw text from scanned documents, images within PDFs, or low-quality text-based documents. It is ideal for scenarios like contract ingestion, receipt processing, or older scanned documents. LAYOUT mode extracts structured layout elements—tables, paragraphs, lines, bounding boxes—preserving the original document's spatial organization. This enables downstream analytical tasks such as table reconstruction or semantic segmentation of document content. OMR (Optical Mark Recognition) is not a supported feature, and "CONTENT" is not a valid mode. By supporting OCR and LAYOUT modes, Snowflake Cortex provides robust document intelligence capabilities directly within the Snowflake environment.

Question: 122

Which of the following parameters can be used with the COPY INTO <TABLE> command to specify the file format?

- A. FILE_FORMAT
- B. DATA_TYPE
- C. FILE_TYPE
- D. LOAD_FORMAT

Answer: A

Explanation:

The FILE_FORMAT parameter is the correct and official Snowflake option used with the COPY INTO <TABLE> command to define the structure and characteristics of the source data files. When loading data, Snowflake needs to understand how to interpret the incoming file—its delimiter, compression, data type (CSV, JSON, PARQUET, etc.), and any additional parsing rules. The FILE_FORMAT parameter can reference either a named file format created earlier (CREATE FILE FORMAT) or an inline definition directly within the COPY INTO statement. Parameters like TYPE, FIELD_DELIMITER, SKIP_HEADER, and ESCAPE can be specified within FILE_FORMAT. None of the other options—DATA_TYPE, FILE_TYPE, LOAD_FORMAT—are valid COPY INTO parameters in Snowflake. DATA_TYPE relates to

column definitions, FILE_TYPE is not a supported keyword, and LOAD_FORMAT is not recognized by Snowflake. Using FILE_FORMAT ensures proper data ingestion, prevents parsing errors, and enables Snowflake to correctly interpret semi-structured and structured files during bulk load operations.

Question: 123

Which option can be used to specify the delimiter for a CSV file in the COPY INTO <TABLE> command?

- A. DELIMITER
- B. SEPARATOR
- C. FIELD_DELIMITER
- D. COLUMN_DELIMITER

Answer: C

Explanation:

The correct Snowflake parameter for defining a CSV delimiter during data loading is `FIELD_DELIMITER`. This option can be set within a named file format or specified inline in a `COPY INTO` command, such as `FIELD_DELIMITER = ','` for comma-separated files or `FIELD_DELIMITER = '\t'` for tab-delimited data. Snowflake does not recognize `DELIMITER`, `SEPARATOR`, or `COLUMN_DELIMITER` as valid parameters. `FIELD_DELIMITER` ensures that Snowflake accurately splits each row into separate fields based on the chosen delimiter. Without the right delimiter definition, the `COPY INTO` operation would treat entire rows as a single string, causing misalignment with the table structure. `FIELD_DELIMITER` is essential when handling CSV or TSV files, particularly when ingesting data from external systems, cloud storage, or partner feeds. Snowflake's flexibility with file formatting—combined with parallel loading and automatic error-handling—makes `COPY INTO` an optimized solution for structured data ingestion.

Question: 124

Which of the following settings can be configured for a Snowflake Virtual Warehouse? (Choose any 3 options)

- A. Cloud provider region
- B. Auto-suspend time
- C. Auto-resume
- D. Warehouse size

Answer: B,C,D

Explanation:

Snowflake Virtual Warehouses support several configuration parameters that directly influence compute behavior, performance, and cost control. Auto-suspend time determines how long the warehouse should remain idle before Snowflake automatically suspends it to save credits. Auto-resume enables automatic warehouse reactivation whenever a new query is submitted, ensuring seamless user experience without manual intervention. Warehouse size determines the compute resources available (e.g., X-SMALL, SMALL, MEDIUM, LARGE). Larger warehouses provide more CPU, memory, and parallel processing ability. Conversely, the cloud provider region cannot be configured at the warehouse level; it is determined when the Snowflake account is created and applies globally across the account. These warehouse settings enable efficient workload management, dynamic compute scaling, and cost optimization, allowing Snowflake users to tailor compute behavior to their analytics and data processing needs.

Question: 125

Which SQL command is commonly used to load structured data from a stage into a Snowflake table?

- A. INSERT INTO
- B. COPY INTO
- C. LOAD DATA
- D. IMPORT DATA

Answer: B

Explanation:

The COPY INTO <table> command is Snowflake's primary and optimized mechanism for loading large volumes of structured data from either internal or external stages. Unlike INSERT INTO, which is used for row-level inserts or small datasets, COPY INTO is designed for high-throughput operations, parallel loading, and performant ingestion from files stored in S3, Azure Blob Storage, GCS, or Snowflake-managed stages. The COPY INTO command supports detailed file

formatting options (such as TYPE=CSV, FIELD_DELIMITER, SKIP_HEADER), error-handling rules (ON_ERROR), and transformations using SQL expressions inside the COPY statement. Snowflake does not support LOAD DATA or IMPORT DATA as SQL commands; these exist in other database systems but not within Snowflake. COPY INTO ensures efficient, resilient, fault-tolerant data ingestion and is heavily used for ETL/ELT, data lake ingestion, and large batch ingestion pipelines.

Question: 126

Which of the following Snowflake functions is used to generate pre-signed URLs for accessing files in a stage?

- A. GET_STAGE_LOCATION()
- B. BUILD_STAGE_FILE_URL()
- C. GET_PREIGNED_URL()
- D. GET_RELATIVE_PATH()

Answer: C

Explanation:

GET_PREIGNED_URL() is the official Snowflake function used to create a temporary, secure, time-limited pre-signed URL, which provides controlled-access retrieval of a file stored in a Snowflake internal or external stage. Pre-signed URLs are essential for secure file sharing without exposing credentials or granting direct stage access to external users. The function accepts the stage name and file path and optionally allows specifying an expiration period. GET_STAGE_LOCATION() retrieves metadata about where a stage points, not access URLs. BUILD_STAGE_FILE_URL() constructs a basic URL but does not sign it, meaning it cannot authorize secure download access. GET_RELATIVE_PATH() provides path-only details, not a secure download link. GET_PREIGNED_URL ensures that external consumers can access staged files safely, making it an important feature for secure distribution workflows, data export pipelines, partner integrations, and temporary file sharing scenarios.

Question: 127

Which of the following system-defined roles exist in Snowflake? (Choose any 3 options)

- A. SECURITYADMIN
- B. DATABASEADMIN
- C. ACCOUNTADMIN
- D. SYSADMIN

Answer: A,C,D

Explanation:

Snowflake provides a predefined set of system-defined roles that enforce Role-Based Access Control (RBAC). These roles ensure structured governance and centralized privilege management across accounts. The primary system-defined roles include:

ACCOUNTADMIN, the highest-privileged role, responsible for global account-level activities such as billing, governance, replication, and cross-region/cloud configuration. It has implicit ownership of all objects.

SECURITYADMIN manages users, roles, MFA, and all privilege grants. This role ensures operational control over user lifecycle management while supporting separation of duties from ACCOUNTADMIN.

SYSADMIN manages objects such as databases, schemas, tables, warehouses, and other compute objects. It is the default role for data engineering and data platform teams needing full control of object creation and maintenance.

DATABASEADMIN does not exist as a system-defined role—it is typically user-created for customization. System roles form a foundational security model for controlled privilege escalation and governance.

Question: 128

How can you search for listings on Snowflake Marketplace? (Choose any 2 options)

- A. Browsing categories and tags
- B. Contact Snowflake Support
- C. User Universal Search
- D. Using the search bar at the top of the marketplace

Answer: A,C,D

Explanation:

Snowflake Marketplace provides multiple intuitive discovery mechanisms to help users explore available listings efficiently. The search bar at the top of the Marketplace is a primary entry point, enabling keyword-based discovery across providers, industries, and dataset types. This facilitates targeted searching for specific topics or data categories.

Universal Search, available in Snowsight, extends search beyond databases and worksheets to include Marketplace listings. This unified search experience allows users to locate datasets without navigating through menus, improving productivity and discoverability.

Browsing categories and tags supports exploratory search by grouping listings under predefined themes such as financial data, geospatial data, demographics, and more. These tags promote structured navigation, especially useful when users are exploring rather than seeking something specific.

Contacting Snowflake Support is not required for accessing or discovering listings and is irrelevant to Marketplace search workflows.

Question: 129

What is the key difference between a Standard Warehouse and a Multi-cluster Warehouse?

- A. Multi-cluster Warehouses can automatically scale horizontally by adding more clusters.
- B. Standard Warehouses can only be used for data loading.
- C. Multi-cluster warehouses cannot be auto suspended.
- D. Standard Warehouses provide better performance for complex queries.

Answer: A

Explanation:

The central difference between a Standard Warehouse and a Multi-cluster Warehouse lies in horizontal scalability and concurrency handling. A Standard Warehouse consists of one compute cluster. Though it can scale up or down vertically (e.g., Small → Medium), it cannot automatically add more clusters during periods of heavy workload. This means that under peak concurrency, queries can queue until compute resources are available.

A Multi-cluster Warehouse, by contrast, supports automatic horizontal scaling, allowing Snowflake to add or remove clusters dynamically based on query demand. This ensures consistent performance under simultaneous workloads, making it essential for high-concurrency analytical environments or BI dashboards.

Both warehouse types share characteristics such as auto-resume, auto-suspend, and compatibility with all query workloads. They differ only in concurrency scaling. Multi-cluster warehouses can still auto-suspend at the warehouse level; the belief they cannot is incorrect. Standard Warehouses are fully functional for all workloads—they are not restricted to loading tasks.

Question: 130

What type of mode available in PARSE_DOCUMENT function? (Select two)

A. CONTENT

B. LAYOUT

C. OMR

D. OCR

Answer: B,D

Explanation:

The `PARSE_DOCUMENT` function offers two primary processing modes: `OCR` and `LAYOUT`, each designed for different document-intelligence tasks.

`OCR` (Optical Character Recognition) mode extracts raw text from scanned or image-based documents. This is ideal for contracts, scanned PDFs, receipts, manuals, and any scenario where the primary goal is text extraction. `OCR` focuses on textual content without preserving document structure.

`LAYOUT` mode extracts both text and structural context, including tables, section groupings, paragraph blocks, and visual positioning. This provides layout-aware data, making it vital for downstream use cases such as Retrieval-Augmented Generation (RAG) pipelines, structured extraction, and content summarization while retaining relationships between document components.

`CONTENT` is not a mode—it is a JSON key in output. `OMR` (Optical Mark Recognition) is not supported within `PARSE_DOCUMENT`.

Question: 131

Which of the following are valid Snowflake Virtual Warehouse sizes? (Choose any 3 options)

A. Small

B. Tiny

C. Medium

D. Xsmall

Answer: A,C,D

Explanation:

Snowflake warehouse sizes follow T-shirt sizing, each representing a doubling of compute resources relative to the previous size. Valid Snowflake warehouse sizes include X-Small, Small, Medium, Large, and up to 6X-Large.

X-Small is the smallest supported warehouse size and is commonly used for light data processing, development tasks, and low-volume operations.

Small doubles the compute capacity of X-Small, providing better performance for moderate workloads.

Medium continues scaling upward, offering increased CPU, memory, and parallelism for heavier analytics or transformation workloads.

Tiny is not a Snowflake warehouse size. Snowflake never exposes a warehouse smaller than X-Small, ensuring consistent minimum compute capabilities.

These standardized size tiers allow predictable resource scaling and cost forecasting.

Question: 132

Which feature does Snowsight provide for query management?

- A. Manual data storage configuration
- B. Limited to simple queries
- C. Pre-built queries only
- D. An intuitive SQL editor for writing and running queries

Answer: D

Explanation:

Snowsight includes a modern, fully capable SQL editor designed for rich query development, execution, and analysis. The editor offers features such as syntax highlighting, intelligent autocomplete, contextual error display, and result-set visualization. Users can organize work with worksheets, folders, tagging, and versioning-like behavior.

Snowsight is built for both simple and complex SQL workloads. It supports multi-statement execution, script-like workflows, stored procedure development, materialized views, and advanced analytics queries.

Unsupported options include:

Manual data storage configuration: Snowflake abstracts storage management completely.

Pre-built queries only: Snowsight allows fully custom SQL.

Limited to simple queries: It supports enterprise workloads, BI query chains, and operational SQL.

Question: 133

Which data type in Snowflake is commonly used for storing semi-structured data?

- A. BLOB
- B. JSON
- C. VARIANT
- D. VARCHAR

Answer: C

Explanation:

Snowflake uses the VARIANT data type as the native mechanism for storing semi-structured data

formats such as JSON, XML, Avro, ORC, and Parquet. VARIANT supports hierarchical, nested structures and allows querying via dot or bracket path notation without requiring schema-on-write.

JSON is not a Snowflake data type—JSON documents are stored inside VARIANT.

VARIANT allows:

Mixed structured and semi-structured content

Efficient storage through Snowflake's columnar compression

Schema-on-read flexibility

Querying nested structures without flattening

VARCHAR stores plain text only.

BLOB is reserved for binary data and not suitable for interactive querying.

This design eliminates the need for preprocessing or schema enforcement at ingestion time.

Question: 134

Which command can be used to query the Information Schema in Snowflake to retrieve a list of tables?

- A. `SELECT * FROM PUBLIC.INFORMATION_SCHEMA;`

B. SELECT * FROM INFORMATION_SCHEMA.COLUMNS;

C. SELECT * FROM INFORMATION_SCHEMA.TABLES;

D. SELECT * FROM PUBLIC.TABLES;

Answer: C

Explanation:

Snowflake's INFORMATION_SCHEMA contains metadata views about databases, schemas, tables, columns, constraints, and other objects. To retrieve a list of tables, the correct view is INFORMATION_SCHEMA.TABLES, which returns:

Table name

Schema name

Table type (BASE TABLE, VIEW, MATERIALIZED VIEW)

Creation timestamp

Owner and retention properties

The query:

SELECT * FROM INFORMATION_SCHEMA.TABLES;

returns metadata for all tables in the current database context.

INFORMATION_SCHEMA.COLUMNS returns column-level metadata, not table listings.

PUBLIC.INFORMATION_SCHEMA is invalid—INFORMATION_SCHEMA is not inside PUBLIC schema.

PUBLIC.TABLES is not a system object unless created manually.

Question: 135

What is the purpose of the ACCOUNTADMIN role in Snowflake? (Choose any 2 options)

A. To grant and revoke privileges across the account

B. To monitor query performance

C. To create and manage databases

D. To manage all aspects of the Snowflake account

Answer: A,D

Explanation:

The ACCOUNTADMIN role is Snowflake's highest-privileged system-defined role. It provides complete administrative authority across the entire Snowflake account. Its functions include:

Managing global account parameters, replication settings, business continuity, failover groups, and region configurations

Administering billing, resource monitoring, and governance

Granting and revoking privileges across all objects and roles

Overseeing role hierarchy, including SECURITYADMIN and SYSADMIN

It is typically reserved for platform owners and security/governance teams, following least-privilege principles.

"Create and manage databases" is primarily a SYSADMIN responsibility.

"Monitor query performance" can be accomplished by roles with MONITOR privileges; it is not exclusive to ACCOUNTADMIN.

Question: 136

What is Snowflake Data Exchange used for?

- A. Storing raw data
- B. Sharing data between specific Snowflake accounts
- C. Managing user permissions
- D. Sharing data with all Snowflake accounts

Answer: B

Explanation:

Snowflake Data Exchange enables secure, governed sharing of live data between Snowflake accounts.

It lets providers publish datasets to specific consumers—partners, vendors, departments, or customers—without copying or transferring data. Consumers query shared data instantly, leveraging Snowflake's secure data sharing architecture.

Key capabilities include:

Targeted sharing to selected accounts

Fine-grained access management and auditing

Live, real-time data sharing without data movement

Ability to customize listings and visibility

Data Exchange is different from the Marketplace, which supports public or broad distribution.

It does not store raw data—storage occurs in provider accounts.

It does not manage user permissions (that is the role of RBAC).

Question: 137

Which of the following types of data can be found on Snowflake Marketplace? (Choose any 3 options)

- A. Public data sets
- B. Financial market data
- C. Proprietary data sets
- D. Third-party data sets

Answer: A,B,D

Explanation:

Snowflake Marketplace serves as a centralized catalog where providers publish ready-to-use datasets, models, and applications across industries. It includes:

Public datasets, such as demographics, weather, healthcare statistics, and government data.

Financial market datasets, including equities, commodities, macroeconomic indicators, and pricing feeds from major financial data vendors.

Third-party datasets supplied by external organizations specializing in geospatial intelligence, retail analytics, marketing insights, healthcare, census data, and more.

“Proprietary datasets” are internal to organizations and typically not published on the public

Marketplace unless the owner chooses to list them. Marketplace listings are curated and governed.

Question: 138

What does a green checkmark next to a cell in a Snowflake notebook signify?

- A. The cell has completed execution successfully
- B. The cell has errors
- C. The cell is currently running
- D. The cell is paused

Answer: A

Explanation:

In Snowflake Notebooks, a green checkmark is a visual indicator that the cell has successfully executed, meaning the computation completed without errors and results are available. This helps users verify execution state quickly across multi-step workflows.

Other visual states include:

A spinning icon for cells that are currently executing

A red error symbol for failed executions

Plain, unmarked cells that have not yet run

Snowflake Notebooks support SQL and Python cells with integrated compute, output visualization, and cell-level execution state tracking. There is no “paused” state for notebook cells.

Question: 139

Before below options what you need for executing INSERT command in Snowflake worksheet?

- A. Active Warehouse
- B. Schema
- C. Database
- D. All of them

Answer: D

Explanation:

Executing an INSERT command in a Snowflake worksheet requires three components to be explicitly selected:

Active Warehouse– Provides compute resources to process the query. Without an active warehouse, DML operations cannot execute.

Database– The table receiving the INSERT must reside in the active database context.

Schema– The targeted table must be within a selected schema.

All three define the fully qualified context for locating and writing to the correct object. Snowflake's compute/storage separation ensures compute is only used when warehouses are active. Schema and database selection ensure correct namespace resolution.

Question: 140

What is the typical syntax for using the LIMIT clause to retrieve the first 10 rows of a table named employees?

- A. SELECT * FROM employees LIMIT 10;
- B. SELECT LIMIT 10 * FROM employees;
- C. LIMIT 10 SELECT * FROM employees;
- D. SELECT * FROM employees WHERE LIMIT 10;

Answer: A

Explanation:

The LIMIT clause is used to restrict the number of rows returned by a query. The correct syntax in Snowflake is:

```
SELECT * FROM employees LIMIT 10;
```

This retrieves the first 10 rows of the result set. LIMIT appears at the end of the SELECT statement. Snowflake follows a SQL syntax similar to MySQL/PostgreSQL regarding LIMIT usage.

Incorrect forms include:

Using LIMIT before SELECT

Embedding LIMIT inside WHERE

Reversing the order of SELECT and LIMIT

LIMIT is frequently used for sampling, record previewing, debugging, and exploratory analysis.

Question: 141

Which layer of Snowflake's architecture provides security features like encryption and authentication?

- A. Compute Layer
- B. Data Storage Layer
- C. Cloud Services Layer
- D. Query Processing Layer

Answer: C

Explanation:

The Cloud Services Layer is responsible for Snowflake's global security, coordination, and metadata management. It orchestrates all authentication and authorization processes, including username/password login, MFA, OAuth, SSO, key-pair authentication, and integration with enterprise identity providers. It also manages role-based access control (RBAC), privilege enforcement, session management, and secure query orchestration.

This layer centralizes encryption policy enforcement, including the management of Snowflake-managed and customer-managed encryption keys (Tri-Secret Secure). While all data stored and transmitted in Snowflake is encrypted by default, it is the Cloud Services Layer that handles the key lifecycle, rotation, and hierarchy.

The Compute Layer only runs queries.

The Storage Layer keeps compressed, encrypted columnar data but does not manage security controls.

The Query Processing Layer executes SQL but does not manage authentication or RBAC.

Thus, the Cloud Services Layer is the authoritative control plane for Snowflake security.

Question: 142

What file formats can be loaded into Snowflake using the COPY INTO command? (Choose any 3 options)

- A. JSON
- B. XLS
- C. Parquet
- D. CSV

Answer: A,C,D

Explanation:

The COPY INTO command supports loading structured and semi-structured data from various Snowflake stages. Supported file formats include CSV, JSON, Parquet, Avro, ORC, and others. Each format integrates with Snowflake's file format definitions, enabling configuration for delimiters, compression, headers, and parsing rules.

CSV is widely used for tabular datasets and supports flexible parsing via FILE_FORMAT options such as FIELD_DELIMITER, SKIP_HEADER, and NULL_IF.

JSON is loaded natively into VARIANT columns, allowing nested structures to be queried immediately through path notation.

Parquet, a high-efficiency columnar format, is optimized for analytic workloads and provides excellent compression and schema evolution benefits.

XLS is not supported, as Snowflake does not support Excel spreadsheets directly. XLS/XLSX files must be converted to CSV or Parquet before loading.

The COPY INTO architecture ensures performant ingestion across multi-node compute clusters.

Question: 143

Query History can display data up to how many days?

A. 28

B. 14

C. 21

D. 7

Answer: B

Explanation:

In Snowsight, the Query History user interface retains and displays executed query information for up to 14 days. This includes query text, status, start/end timestamps, duration, warehouse usage, and

result statistics. Snowsight also offers grouping by session, user, warehouse, and time windows for detailed analytics and troubleshooting.

It is important to distinguish Snowsight UI limits from SQL-based history functions. The table function `INFORMATION_SCHEMA.QUERY_HISTORY()` returns only 7 days of history, whereas Snowsight visually retains 14 days. Snowflake's Account Usage views (e.g., `QUERY_HISTORY`) retain data for longer periods but are meant for programmatic auditing rather than UI display.

Incorrect options:

7 days refers only to Information Schema.

21 or 28 days are not supported by Snowsight's UI.

Thus, the correct visibility window in the Snowsight interface is 14 days.

Question: 144

Which of the following is a potential use case for the `CLASSIFY_TEXT` function?

- A. Generating SQL code from natural language description
- B. Categorizing customer support tickets based on topic
- C. Translating a user manual into multiple languages
- D. Extracting data from scanned invoices

Answer: B

Explanation:

`CLASSIFY_TEXT` is a Snowflake Cortex task-specific function designed to categorize free-form text into predefined buckets. It operates by applying LLM-driven classification logic based on category labels or descriptions provided by the user. A primary and practical use case is automatically categorizing customer support tickets based on content, such as identifying billing issues, login problems, or product defects. This enables automated routing, prioritization, and operational analytics.

The function outputs structured results with the predicted label and confidence score. It is optimized for domain classification rather than open-ended text generation.

Incorrect options:

Generating SQL code is handled by the `COMPLETE` function.

Translation is handled by `TRANSLATE`.

Extracting structured fields from documents is the role of `PARSE_DOCUMENT`.

Thus, classification of incoming text streams is the natural use case of CLASSIFY_TEXT.

Question: 145

Which of the following is a key feature of Snowflake's Data Storage Layer? (Select 2 options)

- A. In-memory storage
- B. Internal optimized, compressed, columnar format
- C. Elastic storage
- D. Elastic Cloud

Answer: B,C

Explanation:

The Data Storage Layer in Snowflake stores all data in a compressed, encrypted, columnar format designed for analytical efficiency. Snowflake automatically reorganizes ingested data into micro-partitions optimized for pruning and high-performance scans. The compression and structuring are fully automated, requiring no tuning or indexing by the user.

Another foundational feature is elastic storage, made possible by Snowflake's cloud-native architecture. Storage automatically scales up or down based on data volume without user intervention, and is separated from compute. This ensures cost efficiency and immediate availability of capacity for large datasets.

Incorrect options:

In-memory storage is not a Snowflake feature; long-term data is persisted in cloud object storage.

Elastic Cloud is not a Snowflake architectural term.

The Data Storage Layer is designed for durability, scalability, and optimal query performance.

Question: 146

How can Python variable substitution be performed in Snowflake notebooks?

- A. By manually editing data files
- B. By configuring network settings
- C. By using placeholders in SQL queries

D. By writing HTML code

Answer: C

Explanation:

In Snowflake Notebooks, Python and SQL operate in an integrated environment. Python variable substitution is performed using string placeholders or f-string interpolation within SQL statements. This allows dynamic query construction where Python values are embedded directly into SQL code executed through the Snowpark session.

Example:

```
table_name = "CUSTOMERS"
```

```
session.sql(f"SELECT * FROM {table_name}").show()
```

This method enables parameterized, context-aware SQL generation—ideal for iterative development, automation, and pipeline construction. It avoids manual text editing and provides consistency across Python and SQL execution layers.

Incorrect responses:

Network settings and HTML are unrelated to variable handling.

Editing data files does not apply to dynamic query parameterization.

Variable substitution is essential for combining logic, iteration, and conditional flows within notebooks.

Question: 147

Which of the following are benefits of using a multi-cluster warehouse in Snowflake? (Choose any 2 options)

- A. Reduced storage costs
- B. Improved concurrency for handling many simultaneous queries
- C. Automatic scaling to meet varying query loads
- D. Faster data loading

Answer: B,C

Explanation:

A multi-cluster warehouse provides Snowflake's highest level of concurrency support by allocating multiple compute clusters behind a single warehouse definition. It automatically adds clusters during peak workload to eliminate query queuing and maintain stable performance for dashboards, BI tools, and multi-user environments.

Improved concurrency is the primary benefit—large numbers of simultaneous queries are distributed across clusters.

Automatic scaling (scaling out and in) ensures compute cost efficiency, as Snowflake adds clusters only when needed and removes them when load decreases.

Incorrect options:

Storage costs are unaffected because compute and storage are independent.

Faster data loading may occur if warehouse size increases, but multi-cluster design is driven by concurrency, not ingestion performance.

The design enables smooth operation in high-demand enterprise environments.

Question: 148

What file extension is commonly used for Snowflake notebooks?

- A. .sql
- B. .ipnb
- C. .ipynb
- D. .txt

Answer: C

Explanation:

Snowflake notebooks use the .ipynb file extension, the standard format for Jupyter notebooks. This format stores executable code, markdown, metadata, and cell outputs in a structured JSON layout. Snowflake adopts this format to ensure compatibility with the broader Python ecosystem, thereby enabling seamless migration between Snowflake and external notebook environments.

The .ipynb structure allows mixed SQL and Python cells, visualizations, Streamlit components, documentation, and stepwise development within Snowsight. It supports reproducibility, collaboration, and integration with Snowpark and Cortex.

Incorrect formats:

.ipnb is a misspelling and invalid.

.sql is used for SQL scripts only.

.txt cannot represent notebook metadata or cell structure.

Thus, .ipynb is the correct and only supported notebook format.

Question: 149

What can a schema in Snowflake contain? (Choose any 3 options)

- A. Views
- B. Tables
- C. Stage
- D. Virtual warehouses

Answer: A,B,C

Explanation:

A schema in Snowflake is a logical container inside a database that groups related objects. It can contain tables, views, stages, file formats, sequences, streams, tasks, and other database objects.

Schemas allow structured organization, privilege management, and separation of workloads.

Tables store structured data.

Views are virtual tables defined by queries on other objects.

Stages store data files for loading/unloading.

Virtual warehouses are compute resources, not database objects, and therefore exist outside schemas. Warehouses belong to the account-level namespace and are managed independently of databases and schemas.

Schemas play a key role in RBAC and object naming through fully qualified names: database.schema.object.

Question: 150

How do databases, schemas, tables, and views interact in Snowflake's hierarchy? (Choose any 2 options)

- A. A view can contain multiple databases
- B. Tables and views store data and provide virtual tables
- C. A database contains schemas, which contain tables and views

D. Schemas manage network settings

Answer: B,C

Explanation:

Snowflake's object hierarchy maintains a structured namespace. At the top is the database, which serves as a container for related schemas. Each schema organizes logical groupings of tables, views, stages, file formats, and other database objects.

Tables store persistent, structured data.

Views provide virtual representations of data through stored SQL queries. They do not store data themselves but reference underlying tables.

The hierarchical structure is:

<database>.<schema>.<object>

Incorrect statements:

Views do not contain databases—they reference objects but do not encapsulate them.

Schemas do not manage network or cloud settings; those are defined at account or cloud service layers.

This hierarchy enables clear object governance, secure sharing, and proper namespace organization.

Question: 151

Which statement is true about Snowflake Data Exchange? (Choose any 2 options)

- A. It is limited to internal data sharing only
- B. It requires complex ETL processes to transfer data
- C. It supports data sharing between different regions and cloud providers
- D. It allows organizations to securely share live, governed data

Answer: C,D

Explanation:

Snowflake Data Exchange provides governed, real-time data collaboration between Snowflake accounts. It enables

providers to publish live datasets while consumers query that data without copying or moving it. Because Snowflake uses secure data sharing primitives at the metadata layer, no ETL pipeline or data duplication are required.

A key advantage is support for cross-region and cross-cloud sharing, allowing collaboration across AWS, Azure, and GCP regions seamlessly.

Data Exchange listings support controlled visibility, entitlement management, and auditing. Providers maintain full control over updates since consumers always access the live, authoritative version of the dataset.

Incorrect statements:

It is not limited to internal sharing—external sharing is a major feature.

ETL is not required because Snowflake's architecture exposes shared objects directly.

Question: 152

What are Snowflake notebooks primarily used for?

- A. Storing raw data
- B. User authentication
- C. Writing and executing SQL queries, Python commands, building visualizations, and documenting analysis
- D. Managing cloud storage

Answer: C

Explanation:

Snowflake Notebooks provide an interactive environment where developers and analysts can combine SQL, Python, visualization libraries, and documentation. Use cases include exploratory analysis, ETL prototyping, data engineering workflows, machine learning model development, and dashboard creation using Streamlit components.

Notebooks allow mixing of SQL and Python cells with shared session state, enabling smooth transitions between Snowpark DataFrames, SQL queries, visual charts, and markdown explanations. They enhance reproducibility and collaboration, supporting versioning, parameterization, and seamless Snowflake compute integration.

Incorrect options:

Raw data is stored in Snowflake tables, not in notebooks.

User authentication is handled by the Cloud Services Layer.

Managing cloud storage is automatic and not a notebook responsibility.

Thus, Snowflake notebooks are an end-to-end development and analytics interface.

Question: 153

What are compute resources called in Snowflake?

- A. Data Nodes
- B. Virtual Warehouses
- C. Compute Clusters
- D. Virtual Machines

Answer: B

Explanation:

Snowflake compute resources are referred to as Virtual Warehouses. A virtual warehouse is a cluster of compute nodes that executes SQL queries, performs DML operations (INSERT/UPDATE/DELETE), and runs data loading or transformation tasks.

Virtual Warehouses provide:

Dedicated compute isolation

Independent scaling (resize at any time)

Concurrency support through multi-cluster mode

Auto-suspend and auto-resume for cost efficiency

While Virtual Warehouses consist of compute clusters under the hood, Snowflake abstracts the underlying VM and node architecture, exposing only the warehouse construct to users. This ensures simplicity and avoids operational burdens such as node management.

Incorrect terms like Data Nodes or Virtual Machines represent underlying infrastructure concepts not exposed to end users.

Question: 154

How do you enable a Directory Table for an external stage in Snowflake?

- A. By default, Directory Tables are enabled
- B. Using the command CREATE DIRECTORY TABLE ...

- C. By setting the `ENABLE_DIRECTORY_TABLE` property to `TRUE` on the external stage using `ALTER STAGE ... SET ENABLE_DIRECTORY_TABLE = TRUE`
- D. By creating a view on the external stage

Answer: C

Explanation:

Directory tables allow Snowflake to expose metadata about files stored in an external stage, enabling users to query file listings using SQL. To enable a directory table, the correct approach is to update an existing external stage using:

```
ALTER STAGE my_stage SET DIRECTORY = (ENABLE = TRUE);
```

This enables automatic synchronization of file metadata from cloud providers such as AWS S3, Azure Blob, or Google Cloud Storage.

Incorrect options:

Directory tables are not enabled by default.

There is no `CREATE DIRECTORY TABLE` command in Snowflake.

Creating a view cannot enable directory table functionality.

Once enabled, users can query:

```
SELECT * FROM DIRECTORY(@my_stage);
```

to retrieve file names, sizes, MD5 hashes, and timestamps.

Question: 155

Which of the following are date and time data types in Snowflake? (Choose any 3 options)

- A. `TIMEDATE`
- B. `TIMESTAMP`
- C. `DATE`
- D. `DATETIME`

Answer: B,C,D

Explanation:

Snowflake provides multiple date and time data types to handle various temporal workloads. The `DATE` type stores only the calendar date (year, month, day), suitable for dimensional modeling, slowly changing dimensions, and calendar-related analytics.

`TIMESTAMP` represents a point in time and includes sub-second precision. Snowflake supports multiple variants:

`TIMESTAMP_NTZ`(no time zone)

`TIMESTAMP_LTZ`(local time zone)

`TIMESTAMP_TZ`(explicit time zone)

These options allow flexibility for global operations, event logs, and time-series analytics.

`DATETIME` is an alias for `TIMESTAMP_NTZ`, meaning it stores both date and time but without timezone awareness. It is commonly used in ETL, application logs, and system-generated events.

`TIMEDATE` is not a valid Snowflake data type and does not exist in Snowflake's type system.

These temporal types support extensive built-in datetime functions, automatic casting, and integration with semi-structured data through `VARIANT`.

Question: 156

Which SQL command is used to view the definition of a Table?

- A. `SHOW TABLE`
- B. `SELECT GET_DDL('TABLE', '<table_name>');`
- C. `SELECT * FROM INFORMATION_SCHEMA.TABLES`
- D. `DESCRIBE TABLE`

Answer: B

Explanation:

The `GET_DDL` function retrieves the Data Definition Language (DDL) script that was used to create a given object.

Executing:

`SELECT GET_DDL('TABLE', 'MY_TABLE');`

returns the full DDL including column definitions, comments, clustering keys, constraints, masking policies, and other metadata. This makes it essential for schema migration, replication, compliance audits, and environment synchronization.

`DESCRIBE TABLE` shows only column-level metadata—names, types, nullability—but does not return the full DDL.

`SHOW TABLE` is not a valid Snowflake command; tables are listed using `SHOW TABLES`.

`INFORMATION_SCHEMA.TABLES` lists table metadata, but the provided option “ABLES” is invalid and still would not provide DDL even if corrected.

Thus, `GET_DDL()` is the only method that returns the complete table definition directly.

Question: 157

Which of the following are examples of system-defined roles in Snowflake? (Choose any 3 options)

- A. USERADMIN
- B. CUSTOM_ROLE
- C. ACCOUNTADMIN
- D. SECURITYADMIN

Answer: A,C,D

Explanation:

Snowflake includes several system-defined RBAC roles to provide structured access control across the platform.

ACCOUNTADMIN is the highest-privileged role, governing the entire account, including billing, replication, failover groups, and object ownership inheritance.

SECURITYADMIN is responsible for user, role, and privilege management, including granting and revoking roles, password policies, and MFA enforcement. It is essential for identity lifecycle governance.

USERADMIN manages user creation, role creation, and user authentication administration. It is commonly assigned to identity administrators.

CUSTOM_ROLE is not a Snowflake system role; it is a user-created role, allowing organizations to design fine-grained, domain-specific access policies.

These predefined roles form the foundation of Snowflake’s security model and support least privilege.

implementations.

Question: 158

Which of the following statements are true about Cortex functions?

- A. They support multiple languages
- B. They can be combined to perform complex data processing tasks
- C. They are limited to a specific domain
- D. They can handle structured and semi-structured data

Answer: A,B,D

Explanation:

Snowflake Cortex functions enable AI-powered operations directly within SQL and Python, supporting multiple languages via translation, classification, summarization, and document extraction functions. They operate seamlessly on structured and semi-structured data stored in VARIANT columns, allowing LLMs to process text, JSON, documents, and rich content.

Cortex functions can be combined within SQL pipelines—for example, using PARSE_DOCUMENT to extract data, followed by CLASSIFY_TEXT to categorize extracted fields, and then SUMMARIZE to produce narrative summaries. This composability enables sophisticated multi-stage workflows for document intelligence, support ticket analysis, compliance automation, and semantic enrichment.

They are not limited to a single domain; Cortex functions apply to finance, retail, support operations, healthcare, legal analysis, and general AI-assisted data processing.

Thus, Cortex functions are versatile, multi-domain, multi-language tools integrated with Snowflake's compute and governance layers.

Question: 159

What does "warehouse scaling in/out" refer to in Snowflake?

- A. Moving data between different storage locations
- B. Changing the region of the warehouse
- C. Adjusting the number of clusters in a multi-cluster warehouse.

D. Changing the size of the warehouse (e.g., from Small to Medium or Vice Versa).

Answer: C

Explanation:

Scaling in/out refers to horizontal scaling, which is available only in multi-cluster warehouses. When Snowflake scales out, additional compute clusters are added automatically to handle high concurrency, preventing query queuing. When load decreases, Snowflake scales in, reducing cluster

count to save compute credits.

This differs from scaling up/down, which refers to changing warehouse size (X-Small → Small → Medium). Vertical scaling increases compute resources for a single cluster but does not improve concurrency for simultaneous workloads.

Scaling in/out is valuable for BI dashboards, workloads with many simultaneous users, peak-hour query loads, and applications requiring consistent performance under high concurrency.

Changing regions is an account-level configuration and unrelated to warehouse scaling.

Question: 160

What is the primary purpose of the CLASSIFY_TEXT function in Cortex?

- A. Translating text to different languages
- B. Generating text completions based on a given prompt
- C. Extracting data from documents
- D. Classifying text into predefined categories

Answer: D

Explanation:

The CLASSIFY_TEXT function performs category assignment for free-form text. Users provide a list of predefined labels, and the function predicts the most appropriate category using a Snowflake-optimized LLM. It is ideal for automatically tagging customer feedback, routing support tickets, labeling product reviews, classifying email content, and segmenting operational logs.

The function outputs both the predicted label and an associated confidence score, enabling downstream filtering,

dashboarding, and automated processing.

Other options belong to different Cortex functions:

TRANSLATE handles multilingual translation.

COMPLETE generates natural language completions.

PARSE_DOCUMENT extracts structured information from documents.

Thus, CLASSIFY_TEXT is task-specific and optimized specifically for classification workloads.

Question: 161

How do you specify a custom delimiter for a CSV file when using COPY INTO <TABLE>?

- A. Using the FIELD_DELIMITER file format option
- B. Using the FIELD_SEPARATOR stage property.
- C. Using the DELIMITER_CHAR table property.
- D. Using the COLUMN_SEPARATOR query parameter

Answer: A

Explanation:

When loading CSV files, Snowflake allows full control over parsing behavior using a FILE_FORMAT object or inline file format options. The FIELD_DELIMITER option specifies the character that separates fields in the CSV—such as comma (,), pipe (|), semicolon (;), or tab.

Example:

```
COPY INTO my_table
```

```
FROM @my_stage
```

```
FILE_FORMAT = (TYPE='CSV' FIELD_DELIMITER='|');
```

This ensures Snowflake interprets each row correctly based on the custom delimiter.

Incorrect options:

FIELD_SEPARATOR does not exist as a stage property.

DELIMITER_CHAR is not a Snowflake file or table option.

COLUMN_SEPARATOR is not a recognized query parameter.

FIELD_DELIMITER is the correct and only method for specifying CSV delimiters in COPY INTO operations.

Question: 162

What is the function of the SKIP_HEADER file format option when loading CSV data?

- A. To include the header row as data.
- B. To ignore empty rows in the data file.
- C. To skip a specified number of header rows.
- D. To automatically create column names from the header row.

Answer: C

Explanation:

The SKIP_HEADER option instructs Snowflake to ignore a defined number of initial rows, typically the column header row in CSV files. For example:

```
CREATE FILE FORMAT my_fmt
```

```
TYPE='CSV'
```

```
SKIP_HEADER=1;
```

When used with COPY INTO, Snowflake loads only the actual data rows while excluding header metadata. This prevents header values from being inserted as data and ensures consistent row structure.

Other options:

SKIP_BLANK_LINES controls handling of empty rows, not headers.

Snowflake does not auto-generate column names from header rows; table schemas must be predefined.

Including header rows as data is undesirable and incorrect in most pipelines.

Thus, SKIP_HEADER is used exclusively to omit header rows during ingestion.

Question: 163

Which of the following is true about Snowflake clones?

- A. Clones consume the same storage as the original object.

- B. Clones initially consume minimal/no storage, with storage only being used for changes made to the clone or the original
- C. Clones can only be created from databases, not tables or schemas.
- D. Clones are automatically synchronized with the original object.

Answer: B

Explanation:

Snowflake's zero-copy cloning creates a metadata-only copy of an object—database, schema, table, or many other object types. At creation time, the clone shares the same underlying micro-partitions as the source, consuming virtually no additional storage.

Storage consumption occurs only when either the clone or the original object is modified, a behavior known as copy-on-write. This makes cloning ideal for development, testing, QA, troubleshooting, backup creation, or controlled sandboxing.

Incorrect statements:

Clones do not duplicate storage at creation.

Cloning applies to tables, schemas, databases, and other supported objects.

Clones do not synchronize with the original; after creation they diverge independently.

This architecture enables quick, cost-efficient replication of environments and data.

Question: 164

What does "warehouse scaling up/down" refer to in Snowflake?

- A. Moving data between different storage locations
- B. Changing the region of the warehouse
- C. Changing the size of the warehouse (e.g., from Small to Medium or Vice Versa).
- D. Adjusting the number of clusters in a multi-cluster warehouse.

Answer: C

Explanation:

Scaling up or down refers to vertical scaling, meaning the warehouse's compute size is increased or decreased. For example, moving from Small → Medium → Large increases CPU, memory, and I/O capacity, enabling faster processing for compute-intensive workloads.

Vertical scaling improves single-query performance, large ETL jobs, complex joins, or transformations. It does not improve concurrency unless multi-cluster mode is also used.

Horizontal scaling (scaling out/in), by contrast, adjusts the number of clusters and is used for concurrency.

Region selection is fixed at account creation and cannot be changed by resizing a warehouse. Storage movement is unrelated to compute rescaling.

Question: 165

What is the purpose of a FILE FORMAT object in Snowflake?

- A. To manage virtual warehouse sizes.
- B. To create data backups.
- C. To specify the format and structure of data files being loaded (e.g., CSV delimiter, JSON compression)
- D. To define security policies for data files

Answer: C

Explanation:

A FILE FORMAT object defines how Snowflake interprets files during loading and unloading operations. It includes settings such as:

Type (CSV, JSON, Parquet, Avro)

FIELD_DELIMITER, SKIP_HEADER, NULL_IF

Compression (GZIP, BZ2)

Time and date format handling

Encoding and escape character rules

By encapsulating these parsing rules, organizations standardize ingestion logic across stages and COPY INTO commands.

FILE FORMATS improve data consistency, reduce errors, and simplify maintenance by centralizing file parsing configurations. They do not manage compute resources, create backups, or define data security policies—those belong

to warehouses, fail-safe/Time Travel, and RBAC respectively.

Question: 166

Which SQL function is used to parse a string as JSON data within a Snowflake query?

- A. CONVERT_JSON()
- B. TO_JSON()
- C. PARSE_JSON()
- D. EXTRACT_JSON()

Answer: C

Explanation:

The PARSE_JSON() function converts a valid JSON string into a VARIANT value. This allows Snowflake to store and query nested, hierarchical data using dot and bracket notation.

Example:

```
SELECT PARSE_JSON('{"name":"John","age":30}') AS data;
```

After parsing, fields can be accessed like:

data.name or data['age']

Incorrect options:

TO_JSON() converts VARIANT to a JSON string (opposite direction).

CONVERT_JSON and EXTRACT_JSON are not Snowflake functions.

PARSE_JSON is essential for dynamically loading, transforming, or analyzing JSON content from files, streams, or external applications.

Question: 167

How does Snowflake's compute layer handle query execution?

- A. By optimizing data in cloud storage
- B. With shared-disk architecture

- C. Using MPP (massively parallel processing) compute
- D. Using single-threaded processing

Answer: C

Explanation:

Snowflake's compute layer uses Massively Parallel Processing (MPP), meaning queries are divided into smaller tasks distributed across multiple compute nodes in the Virtual Warehouse. Each node processes a portion of the data simultaneously, maximizing parallelism and drastically reducing query times.

Although Snowflake uses a central storage layer (shared-disk model), the compute engine behaves like a shared-nothing MPP system, where each node handles local processing independently, minimizing contention.

Incorrect options:

Snowflake does not rely on single-thread execution.

Storage optimization occurs at the Storage Layer, not compute.

Snowflake does not use traditional shared-disk execution; compute nodes work in parallel independently.

This architecture enables high performance for large analytical workloads.

Question: 168

Given a table named MY_TABLE, which SQL statement would create a clone named MY_TABLE_CLONE?

- A. COPY TABLE MY_TABLE TO MY_TABLE_CLONE;
- B. CREATE TABLE MY_TABLE_CLONE CLONE MY_TABLE;
- C. BACKUP TABLE MY_TABLE TO MY_TABLE_CLONE;
- D. RESTORE TABLE MY_TABLE TO MY_TABLE_CLONE;

Answer: B

Explanation:

The correct SQL syntax to create a zero-copy clone of an existing table is:

```
CREATE TABLE MY_TABLE_CLONE CLONE MY_TABLE;
```

This command instantly creates a new table that references the same underlying micro-partitions as the original. Because

of Snowflake's metadata-only cloning, no storage is consumed at the time of creation. Storage only increases when either the original or the clone diverges through DML operations, following a copy-on-write model.

Cloning is available for multiple object types—tables, schemas, databases, stages, streams, tasks, and more. This capability enables rapid creation of development sandboxes, QA environments, rollback copies, or controlled experimentation without duplicating data.

Incorrect options:

“COPY TABLE” is not a valid Snowflake command.

BACKUP/RESTORE are not Snowflake SQL commands.

RESTORE applies only to Time Travel or Fail-safe, not to cloning.

Thus, the CLONE keyword is the only correct method for zero-copy duplication.

Question: 169

What is a key benefit of using Streamlit for data visualization in Snowflake notebooks?

- A. Manual data storage
- B. Simplified code execution
- C. Fixed-size compute clusters
- D. Real-time interactivity and dynamic visualizations

Answer: D

Explanation:

Streamlit inside Snowflake Notebooks enables users to build interactive, real-time visualizations and data applications directly within Snowsight. Because Streamlit is event-driven, users can create UI components such as sliders, dropdowns, date pickers, buttons, and filters that dynamically update charts, tables, and computed outputs.

Data remains inside Snowflake, processed through Python with Snowpark, eliminating data movement to external systems. This allows rapid prototyping of dashboards, analytical workflows, and AI-driven apps. Streamlit also supports layout customization, reactive components, and seamless integration with Snowflake compute for scalable analytics.

Incorrect options:

Streamlit does not manage data storage.

Streamlit does not simplify code execution; it enhances visualization and interactivity.

Warehouse clusters are not fixed-size—Snowflake warehouses can scale.

Thus, its primary benefit is enabling rich interactive analytics directly within Snowflake.

Question: 170

Which statement is true regarding internal stages?

- A. Internal stages can only be used for semi-structured data
- B. Internal stages are accessed with external cloud provider credentials
- C. Internal stages are managed and stored within Snowflake
- D. Internal stages can only be used for structured data

Answer: C

Explanation:

Internal stages are fully Snowflake-managed storage locations used to temporarily store data files for loading or unloading operations. Examples include:

User stages (@~)

Table stages (@%table_name)

Named internal stages (@my_internal_stage)

Because they reside inside Snowflake, users do not need AWS, Azure, or GCP credentials to access them. Internal stages support all file types Snowflake can process, including CSV, JSON, Parquet, Avro, ORC, images, PDFs, and other semi-structured formats.

Incorrect statements:

They are not limited to structured or semi-structured data—they accept all supported formats.

External cloud credentials are only needed for external stages (S3, Blob, GCS).

Internal stages simplify secure data ingestion and operational pipelines because Snowflake handles encryption, metadata, and lifecycle management.

Question: 171

What types of privileges can be granted to a role in Snowflake?

- A. INSERT
- B. SELECT
- C. MONITOR
- D. All of them

Answer: D

Explanation:

Snowflake uses role-based access control (RBAC), where privileges are assigned to roles rather than individual users. All of the listed privileges (SELECT, INSERT, MONITOR) can be granted to a role.

SELECT allows reading data from tables, views, or materialized views.

INSERT allows adding new rows into a table.

MONITOR allows viewing query history, warehouse status, and resource usage.

In addition, roles can receive privileges such as CREATE TABLE, USAGE, MODIFY, OPERATE, OWNERSHIP, and many others. Users inherit all privileges assigned to the roles they possess, enabling scalable security design.

Thus, all listed privileges can be granted to Snowflake roles.

Question: 172

Which package is needed for Python worksheet?

- A. NumPy
- B. Snowpark
- C. Matplotlib
- D. Pandas

Answer: B

Explanation:

Snowflake's Python Worksheets require the Snowpark for Python package. Snowpark integrates deeply with Snowflake's compute engine, enabling DataFrame operations, UDF creation, ML workflows, and direct execution of Python logic

inside Snowflake compute.

While additional libraries like Pandas, NumPy, and Matplotlib may be available in worksheets, they are not required to enable the Python execution environment. Snowpark is the core computational interface that connects Python to Snowflake's data and compute layers.

It enables:

Distributed data processing

Transformations expressed in Python but executed inside Snowflake

Access to Snowflake tables as DataFrames

Integration with Snowpark ML and Cortex functions

Thus, Snowpark is the essential package for Python worksheet support.

Question: 173

What is the maximum size limit for a single VARIANT value in Snowflake?

- A. 1MB
- B. 8MB
- C. 32MB
- D. 16MB

Answer: D

Explanation:

A single VARIANT value in Snowflake has a maximum size limit of 16 MB (uncompressed). This limit applies to semi-structured data stored in VARIANT, such as JSON, XML, Avro, and Parquet objects.

This limit ensures efficient storage, query performance, and micro-partitioning. Exceeding the size results in load or processing errors. For ingestion, when loading large JSON files, it's recommended to break them into multiple objects or use JSON Lines formatting.

Snowflake has introduced an optional behavior change bundle (2025_03), increasing the limit to 128 MB for accounts that enable it, but the standard default remains 16 MB.

Thus, under default behavior, 16 MB is the correct maximum value size.

Question: 174

Which function in Snowflake Cortex LLM are Task Specific Function? (Select two)

- A. TRANSLATE
- B. CLASSIFY_TEXT / AI_CLASSIFY
- C. COUNT_TOKENS
- D. PARSE_DOCUMENT

Answer: A,B,D

Explanation:

Snowflake's Cortex LLM includes task-specific functions, meaning each performs a well-defined AI operation with predictable outputs. Examples include:

TRANSLATE— Converts text between languages; deterministic and domain-independent.

CLASSIFY_TEXT / AI_CLASSIFY— Assigns text to predefined categories, ideal for sentiment, topics, or routing tasks.

PARSE_DOCUMENT— Extracts structured information from documents (PDFs, invoices, receipts, contracts) including layout-aware content.

These functions are optimized for reliability, reproducibility, and governance, making them suitable for production pipelines.

COUNT_TOKENS is not task-specific—it's a utility function used to estimate LLM token usage rather than perform a primary AI task.

Thus, TRANSLATE, CLASSIFY_TEXT, and PARSE_DOCUMENT are the correct task-specific functions.

Question: 175

What are the steps to create a new schema in Snowsight?

- A. Data >> Add Data >> Create Schema
- B. Data >> Create Schema
- C. Data >> Database >> Create Schema
- D. Data Product >> Create Schema

Answer: C

Explanation:

To create a new schema in Snowsight, users must follow the hierarchical navigation of Snowflake's object structure.

The correct sequence is:

Data → Database → Create Schema

Steps:

Open the **Data** tab in the left navigation.

Select a **Database** where the schema will be created.

Choose **Create** → **Schema**.

Enter the schema name and (optionally) description.

Click **Create**.

Incorrect options:

"Add Data" is used for loading files, not creating schema objects.

"Data → Create Schema" is incomplete because Snowflake must know which database the schema belongs to.

"Data Product" is unrelated to object creation.

Thus, Snowflake requires explicit database context before schema creation.

Question: 176

Which SQL clause is used to query historical data in Snowflake using Time Travel?

- A. AT | BEFORE (TIMESTAMP => ...)
- B. BACKUP
- C. HISTORY
- D. RESTORE

Answer: A

Explanation:

Snowflake's Time Travel enables querying historical data as it existed at a prior point in time. This is accomplished using the `AT` or `BEFORE` clause within the `FROM` statement:

```
SELECT * FROM my_table AT (TIMESTAMP => '2025-06-18 10:00:00');
```

```
SELECT * FROM my_table BEFORE (OFFSET => -3600);
```

Time Travel supports access via timestamp, statement ID, or time offset. This is essential for auditing, recovery, and reconstructing past states for analysis.

Incorrect options:

`RESTORE` is used to recover dropped objects, not to query history.

`BACKUP` is not a Snowflake SQL clause.

`HISTORY` is not a valid SQL clause.

Thus, `AT` | `BEFORE` enables precise historical queries.

Question: 177

What is a key benefit of using Snowsight for loading data into Snowflake?

- A. Requires external tools for data loading
- B. Manual data storage configuration
- C. Simplified and intuitive data loading process
- D. Limited to specific file formats

Answer: C

Explanation:

Snowsight provides an intuitive UI-based workflow for loading data, enabling drag-and-drop file upload, schema inference, previewing records, automatic table creation, and loading into existing tables. Users can load CSV, JSON, Parquet, AVRO, ORC, and other supported formats without writing `COPY INTO` commands manually.

Snowsight also generates the equivalent SQL code, helping users understand and reuse ingestion scripts in pipelines. It eliminates the need for external tools such as SnowSQL CLI or custom ETL scripts.

Incorrect statements:

External tools are not required.

Manual storage configuration is unnecessary; Snowflake abstracts all storage.

Snowsight supports a wide variety of formats, not limited ones.

Thus, the key advantage is asimple, guided data-loading workflow, especially useful for new Snowflake users.

Question: 178

What is a primary function of a view in Snowflake?

- A. Providing a virtual table based on the result set of a query
- B. Executing network configurations
- C. Managing user authentication
- D. Storing raw data

Answer: A

Explanation:

A view in Snowflake represents a virtual table whose contents are defined by a stored SQL query.

Views allow users to encapsulate transformation logic, simplify complex joins, enforce column-level security, and provide curated datasets for downstream consumers.

Views do not store data themselves; the underlying tables store the data, and Snowflake executes the view's query each time the view is referenced. This makes views ideal for abstraction layers, semantic modeling, and separating compute costs across user groups.

Incorrect options:

Snowflake views do not store raw data.

Views have no role in authentication or network configuration.

Views help streamline business logic and control data access efficiently.

Question: 179

How can you query semi-structured data (e.g., JSON) stored in a Snowflake VARIANT column?

- A. Using standard SQL WHERE clauses on the entire VARIANT column.
- B. Using path notation to access specific elements within the VARIANT
- C. By converting the VARIANT column to a relational table first.

D. Only through stored procedures.

Answer: B

Explanation:

Semi-structured data like JSON stored in a VARIANT column can be queried directly using path notation, either dot notation or bracket notation.

Examples:

Dot notation: `SELECT data:customer.name FROM table;`

Bracket notation: `SELECT data['customer']['name'] FROM table;`

Snowflake automatically interprets the JSON structure, making relational extraction unnecessary.

Complex fields can be accessed through functions such as `FLATTEN`, `OBJECT_KEYS`, `ARRAY_SIZE`, and `TYPEOF`.

Incorrect options:

Converting JSON to a relational table is optional, not required.

Stored procedures are not necessary for JSON querying.

`WHERE` on the full VARIANT cannot precisely extract fields.

Thus, path notation is the native, efficient method.

Question: 180

Which SQL command is used to create a new SCHEMA in Snowflake? (Choose any 3 options)

A. `CREATE SCHEMA Schema_name;`

B. `USE SCH Schema_name;`

C. `USE SCHEMA Schema_name;`

D. `CREATE SCH Schema_name;`

Answer: A

Explanation:

The only valid Snowflake SQL command among the options for creating a schema is:

`CREATE SCHEMA schema_name;`

This command creates a schema within the currently active database. Optional modifiers like `IF NOT EXISTS` or `OR REPLACE` can be used to control behavior.

Other options:

`USE SCHEMA` switches to an existing schema; it does not create one.

`USE SCH` is invalid because `SCH` is not a Snowflake keyword.

`CREATE SCH` is invalid; `SCH` is not a valid abbreviation.

Thus, the only valid schema creation command is `CREATE SCHEMA`.

Question: 181

What information can users view in the Query History option in Snowsight?

- A. Only successful queries
- B. Data storage configuration
- C. Network settings
- D. Details of executed queries, including status, duration, and resource usage

Answer: D

Explanation:

Snowsight's Query History interface provides comprehensive visibility into executed queries. Users can view:

Query text

Execution status (running, succeeded, failed)

Duration and start/end timestamps

Warehouse used and credits consumed

Partition pruning details

Error messages

Query profile for performance diagnostics

This allows developers, analysts, and administrators to troubleshoot slow queries, identify bottlenecks, and monitor compute usage.

Incorrect options:

Query History shows failed and running queries, not just successful ones.

Storage configuration is unrelated to query operations.

Network settings are managed by the Cloud Services Layer, not Query History.

Thus, Query History is a key operational tool in Snowsight.

Question: 182

How do you drop a schema named "temp_schema" in Snowflake?

- A. DROP SCHEMA temp_schema;
- B. DROP DATABASE temp_schema;
- C. DELETE SCHEMA temp_schema;
- D. DROP VIEW temp_schema;

Answer: A

Explanation:

The correct SQL command for removing a schema in Snowflake is:

DROP SCHEMA temp_schema;

This command deletes the schema and all objects contained within it, including tables, views, stages, file formats, and sequences. Snowflake performs this operation atomically, ensuring metadata consistency during the drop process. Users can also include `IF EXISTS` or the `CASCADE` keyword to handle dependencies more explicitly:

DROP SCHEMA IF EXISTS temp_schema CASCADE;

This safely handles scenarios where the schema may not exist or contains objects that would normally block deletion.

Incorrect options:

`DROP DATABASE temp_schema` removes an entire database, not a schema.

`DELETE SCHEMA` is not valid SQL; SQL uses `DROP` for schema removal.

DROP VIEW temp_schema applies only to removing a view object.

Dropping a schema requires USAGE and OWNERSHIP privileges, typically granted to roles such as SYSADMIN or ACCOUNTADMIN.

Question: 183

Which of the following components are included in a fully qualified name in Snowflake? (Choose any 3 options)

- A. Database name
- B. Role name
- C. Schema name
- D. Table or view name

Answer: A,C,D

Explanation:

A fully qualified name in Snowflake specifies the exact location of an object and follows this structure:

<database_name>.<schema_name>.<object_name>

This enables precise referencing, especially in environments with multiple databases or schemas.

Components:

Database name— the top-level container.

Schema name— the logical grouping of related objects.

Table or view name— the specific object being referenced.

Example:

SALES_DB.PUBLIC.ORDERS

Roles are not part of object names. Roles control authorization, while fully qualified names handle namespace resolution.

Using fully qualified names helps avoid ambiguity in pipelines, stored procedures, views, and cross-database queries.

Snowflake also supports partially qualified names depending on active context (current database and schema).

Question: 184

What does the SELECT * statement do in a Snowflake query?

- A. Retrieves all columns from a specified table or view
- B. Retrieves only distinct values
- C. Retrieves only the primary key column
- D. Retrieves a limited number of rows

Answer: A

Explanation:

The SELECT * statement instructs Snowflake to return all columns from the referenced table or view. This is commonly used during data exploration, debugging, initial data profiling, and validation steps. It allows users to quickly view the complete dataset structure without manually specifying each column name.

However, while SELECT * retrieves all columns, it does not limit the number of rows. To restrict rows, developers must include a LIMIT clause (e.g., SELECT * FROM table LIMIT 10;).

The query does not automatically apply DISTINCT or primary key filtering—Snowflake returns all rows exactly as stored unless additional filtering, WHERE conditions, or ordering are provided.

Though SELECT * is convenient, Snowflake best practices recommend explicitly selecting columns in production workloads to optimize performance and avoid unnecessary scanning of unused fields.

Question: 185

Which Snowflake architectural layer is responsible for persistent storage of customer data?

- A. Database Storage Layer
- B. Cloud Services Layer
- C. Virtual Warehouse Layer
- D. Query Processing Layer

Answer: A

Explanation:

The Database Storage Layer is responsible for persistently storing all customer data in Snowflake. This layer automatically manages micro-partitioning, compression, and metadata. Data is stored in optimized, columnar formats inside cloud

object storage (AWS S3, Azure Blob Storage, or GCP Storage). Users cannot manipulate physical storage directly; access is only through SQL operations. Other layers serve different purposes: Cloud Services manages metadata and orchestration, Query Processing Layer executes workloads, and Virtual Warehouses provide compute but do not store data.

Question: 186

What is the primary benefit of Zero-Copy Cloning in Snowflake?

- A. It allows creating writable copies of databases or tables without duplicating stored data.
- B. It creates a full physical copy of data for archival.
- C. It enables real-time data synchronization between regions.
- D. It prevents any modifications to the cloned data.

Answer: A

Explanation:

Zero-Copy Cloning allows Snowflake users to create instant, writable clones of databases, schemas, or tables without physically copying data. Instead, Snowflake references the same underlying micropartitions, making the clone creation nearly instantaneous and storage-efficient. New storage is consumed only when updates apply to either the original or the clone, using copy-on-write mechanics. It does not perform physical copies, does not sync data in real-time, and does not freeze the clone from modification.

Question: 187

What is the main advantage of using Materialized Views in Snowflake?

- A. They always contain the latest data in real-time.
- B. They can be directly updated by DML statements.
- C. They physically store pre-computed results to improve query performance.
- D. They are primarily used for data masking.

Answer: C

Explanation:

Snowflake Materialized Views physically store the results of a defined query. This pre-computation significantly improves performance for repetitive or complex queries, especially those based on aggregations and joins. Snowflake automatically maintains materialized views asynchronously. They do not update in real-time, cannot be changed through DML, and are not intended for masking— dynamic masking policies serve that purpose.

Question: 188

Which of the following statements are true about Zero-Copy Cloning in Snowflake? (Select TWO)

- A. It requires significant additional storage.
- B. It is immediate and metadata-only operation.
- C. It duplicates the underlying data storage.
- D. It creates a writable copy of an object.

Answer: B,D

Explanation:

Zero-Copy Cloning is a metadata-only operation, creating instantaneous clones without copying underlying data. Clones are fully writable and operate independently. Snowflake only allocates storage when data diverges due to modifications. It does not require significant storage nor duplicate physical data.

Question: 189

What is the default Time Travel retention period for standard tables in Snowflake?

- A. 0 days
- B. 1 day

C. 7 days

D. 14 days

Answer: B

Explanation:

Standard permanent tables in Snowflake have a default 1-day Time Travel retention period. During this period, users may query historical data, restore dropped tables, or clone past states. Higher editions allow extending this period up to 90 days. Temporary and transient tables have 0-day retention by default.

Question: 190

How is the initial activation of the Snowflake Data Exchange feature, which is necessary for establishing private exchanges, typically accomplished?

- A. Any Snowflake account administrator can enable Data Exchange with a click.
- B. Data Exchange is automatically enabled for all accounts.
- C. Activation requires a SQL command executed by ACCOUNTADMIN.
- D. Activation requires a request to Snowflake Support.

Answer: D

Explanation:

Snowflake Data Exchange is not enabled by default. Accounts must open a support request to Snowflake Support, which then activates the feature. It cannot be enabled via UI or SQL. After activation, administrators may create private exchanges and listings.

Question: 191

What is a Reader Account in the context of Snowflake Data Sharing?

- A. A restricted account used for internal Snowflake operations.
- B. A standalone account provided by a data provider to a consumer who does not have a Snowflake account.
- C. An account primarily used for data replication.

D. An account that can only read data from external stages.

Answer: B

Explanation:

A Reader Account allows data providers to share data with recipients who do not have their own Snowflake subscriptions. Providers manage and pay for compute and storage used by the reader account. Reader accounts are limited to querying shared data and cannot alter Snowflake objects. They are not for replication, internal operations, or external stage reading.

Question: 192

Which layer of the Snowflake architecture performs query optimization and execution?

- A. Database Storage Layer
- B. Client Layer
- C. Cloud Services Layer
- D. Query Processing Layer

Answer: D

Explanation:

The Query Processing Layer executes SQL statements using virtual warehouses and massively parallel processing. It handles query optimization, execution, and scaling. The Cloud Services Layer parses and plans queries but does not execute them. Storage persists data, and the Client Layer is external to Snowflake architecture.

Question: 193

What is the primary responsibility of Role Hierarchy in Snowflake's access control model?

- A. To allow roles to inherit privileges from parent roles
- B. To define the order of SQL command execution
- C. To categorize different types of users
- D. To specify the size of virtual warehouses

Answer: A

Explanation:

Role hierarchy enables a parent role to inherit privileges assigned to child roles. This simplifies RBAC administration, reduces redundant grants, and ensures scalable security management. It has no relation to SQL sequencing, user categorization, or compute sizing.

Question: 194

Which of the following are date and time data types in Snowflake? (Choose any 3 options)

- A. TIMEDATE
- B. DATE
- C. Python
- D. TIMESTAMP
- E. DATETIME
- F. JavaScript

Answer: B,D,E

Explanation:

Snowflake supports several native data types for managing temporal information:

DATE – Stores dates only (year, month, day) without any time component.

TIMESTAMP – Represents date and time values, including several variants:

- **TIMESTAMP_NTZ** (no time zone)

- `TIMESTAMP_LTZ` (local time zone)
- `TIMESTAMP_TZ` (explicit time zone)

`DATETIME` – An alias for `TIMESTAMP_NTZ`, storing both date and time.

These data types enable powerful time-based analytics and comparisons.

`XTIMEDATE` is not a valid data type.

`X` Python and JavaScript are programming languages, not Snowflake data types.

Question: 195

What is the primary use of Snowflake Cortex LLM functions?

- A. Running machine learning models on structured data.
- B. Optimizing data loading performance.
- C. Performing traditional SQL joins.
- D. Analyzing text data for sentiment summarization and other AI/ML tasks.

Answer: D

Explanation:

Cortex LLM functions integrate large language model capabilities directly inside Snowflake. These functions enable:

- Sentiment analysis
- Summarization
- Document question answering
- Text classification
- Embedding generation for vector search

They operate directly on Snowflake data, require no external infrastructure, and support secure inplatform AI workloads.

Question: 196

Which command will create a named internal stage?

- A. CREATE @STAGE
- B. CREATE NAMED STAGE
- C. CREATE STAGE
- D. CREATE INTERNAL STAGE

Answer: C

Explanation:

CREATE STAGE is the correct Snowflake DDL command for creating a named internal stage. Stages hold files for loading/unloading. The command defaults to internal storage unless external parameters are provided.

Other options are not valid Snowflake syntax.

Question: 197

What is the primary purpose of the COPY INTO command in Snowflake for data loading?

- A. To replicate databases across regions.
- B. To create new virtual warehouses.
- C. To load data from staged files into a table.
- D. To manage user roles and permissions.

Answer: C

Explanation:

COPY INTO <table> bulk-loads data from internal or external stages into Snowflake tables. It supports transformations, validation, and multiple file formats.

It does not replicate databases, manage roles, or create warehouses.

Question: 198

Which of the following are valid components of Snowflake's core architecture? (Select THREE)

- A. On-premises Data Centers
- B. User Interface Layer
- C. Third-Party ETL Tools
- D. Database Storage Layer
- E. Query Processing Layer
- F. Cloud Services Layer

Answer: D,E,F

Explanation:

Snowflake's architecture consists of:

- Database Storage Layer – Stores optimized columnar data.
- Query Processing Layer – Executes workloads via virtual warehouses.
- Cloud Services Layer – Handles metadata, optimization, access control.

ETL tools and UI are external integrations. Snowflake is cloud-native and has no on-premises components.

Question: 199

When a virtual warehouse is scaled up (e.g., from X-Small to Small), what is the primary impact?

- A. It allows more concurrent users to connect.
- B. It can process larger and more complex queries faster.
- C. It gains more storage capacity.
- D. It reduces the cost of running queries.

Answer: B

Explanation:

Scaling up a warehouse increases compute resources (CPU, memory), improving performance on heavy workloads: large joins, transformations, and long-running analytical queries.

It does not increase storage, reduce cost, or guarantee higher concurrency (multi-cluster handles concurrency).

Question: 200

What is the SHOW GRANTS SQL command primarily used for?

- A. To view the history of data loads.
- B. To display all privileges granted to a role or user.
- C. To list all active users in the account.
- D. To show current warehouse credit usage.

Answer: B

Explanation:

SHOW GRANTS lists privileges assigned to roles, users, objects, or shares. It is essential for permissions auditing and RBAC management.

SHOW USERS lists users, not privileges.

Warehouse usage is in ACCOUNT_USAGE views.

Load history is retrieved via COPY_HISTORY or LOAD_HISTORY.

Question: 201

Which System-defined role is typically responsible for creating and managing users and roles within an account?

- A. USERADMIN
- B. SECURITYADMIN

C. ACCOUNTADMIN

D. SYSADMIN

Answer: B

Explanation:

SECURITYADMIN manages identity and access control, including creating, modifying, and dropping users and roles.

USERADMIN manages users but lacks full privilege management capability.

SYSADMIN manages object creation.

ACCOUNTADMIN holds full administrative privileges but delegates user/role management to SECURITYADMIN.

Question: 202

Which data type in Snowflake is best suited for storing semi-structured data like JSON or Avro without a predefined schema?

A. STRING

B. TEXT

C. VARCHAR

D. VARIANT

Answer: D

Explanation:

VARIANT supports nested, flexible structures and enables querying via JSON path notation. It is optimized for semi-structured formats (JSON, Avro, ORC, Parquet).

STRING, VARCHAR, and TEXT store plain text and cannot represent nested structures without conversion.

Question: 203

Which edition of Snowflake offers the most comprehensive security and compliance features, including HIPAA and PCI DSS compliance?

- A. Standard Edition
- B. Business Critical Edition
- C. Enterprise Edition
- D. Virtual Private Snowflake (VPS) Edition

Answer: B

Explanation:

Business Critical Edition includes advanced compliance certifications such as HIPAA, PCI DSS, HITRUST, and additional encryption and security controls.

Standard and Enterprise editions lack these compliance guarantees.

VPS provides even stronger isolation but builds on Business Critical features.

Question: 204

What kind of information can be found in Query History in Snowsight? (Select THREE)

- A. Virtual Warehouse size used
- B. Credit consumption
- C. Execution Status (e.g. Succeeded, Failed)
- D. Duration of the query
- E. Query ID

Answer: C,D,E

Explanation:

Query History in Snowsight includes:

- Query ID – A unique identifier for each executed query.
- Execution Status – Indicates whether the query succeeded, failed, or was canceled.
- Duration – Total time taken for the query to run.

While warehouse name is shown, warehouse size is not explicitly displayed. Credit consumption appears in Account Usage views, not directly in Query History.

Question: 205

Which Snowflake object is a logical container for schemas and other database objects?

- A. Schema
- B. View
- C. Table
- D. Database

Answer: D

Explanation:

A Database is the highest logical container for schemas and their objects in Snowflake.

Hierarchy:

Account → Database → Schema → Tables, Views, Stages, etc.

Schemas organize objects within a database; tables and views do not contain objects.

Question: 206

What is the purpose of auto-resume on a virtual warehouse?

- A. To restart the warehouse manually after suspension.
- B. To automatically start the warehouse when a query is submitted to it.
- C. To automatically scale up the warehouse size.
- D. To prevent the warehouse from suspending.

Answer: B

Explanation:

When auto-resume is enabled, Snowflake automatically starts a suspended warehouse when a query or task is

submitted. This ensures smooth operations without manual intervention.

It does not scale the warehouse, prevent suspension, or require manual restarting.

Question: 207

What is the highest level object in the Snowflake object hierarchy?

- A. Database
- B. Virtual Warehouse
- C. Schema
- D. Account

Answer: D

Explanation:

The Account is the top-level Snowflake container encompassing:

- All databases
- All schemas
- All compute resources (warehouses)
- All roles, users, and governance structures

All other objects exist within the Account context.

Question: 208

Which types of stages are supported in Snowflake for data loading and unloading? (Select TWO)

- A. Managed Stages
- B. Permanent Stages
- C. Temporary Stages
- D. External Stages
- E. Internal Stages

Answer: D,E

Explanation:

Snowflake supports two official stage types:

- Internal Stages – User stages, table stages, and named internal stages stored inside Snowflake.
- External Stages – Refer to cloud storage (S3, Azure Blob, GCS) using integration credentials.

Managed, permanent, and temporary stages are not official Snowflake classifications.

Question: 209

Which command is used to view the details of a file format object in Snowflake?

- A. LIST FILE FORMAT
- B. ALTER FILE FORMAT
- C. DESCRIBE FILE FORMAT
- D. SHOW FILE FORMATS

Answer: C

Explanation:

DESCRIBE FILE FORMAT <name> returns configuration settings such as FIELD_DELIMITER, TYPE, SKIP_HEADER, and COMPRESSION for the specified format.

SHOW FILE FORMATS only lists existing formats, not their internals.

LIST FILE FORMAT is invalid. ALTER modifies but does not display details.

Question: 210

When unloading data from a Snowflake table, what is the target destination for the COPY INTO @stage command?

- A. An internal or external stage
- B. Another Snowflake table
- C. A local file system

D. A data stream

Answer: A

Explanation:

COPY INTO @stage writes query results or table data into files stored in either internal or external stages.

Unloading cannot target a local filesystem directly, cannot load into another table (use INSERT instead), and does not write to streams.

Question: 211

What is the main characteristic of Snowflake's Standard Edition?

- A. Offers unlimited Time Travel history.
- B. Includes Business Critical features like HIPAA compliance.
- C. Supports Private Connectivity (AWS PrivateLink, Azure Private Link).
- D. Provides core Snowflake features for data warehousing and analytics.

Answer: D

Explanation:

Standard Edition provides all essential Snowflake capabilities:

- SQL data warehousing
- Secure data sharing
- Time Travel (1 day)
- Internal/external stages
- Basic governance features

It does not include HIPAA/PCI compliance (Business Critical), nor unlimited Time Travel, nor advanced private connectivity.

Question: 212

When should AUTO_SUSPEND be enabled for a virtual warehouse?

- A. To ensure the warehouse is always active for immediate queries.
- B. To save credits during periods of inactivity.
- C. When the warehouse is expected to run continuously.
- D. To increase the warehouse's performance.

Answer: B

Explanation:

AUTO_SUSPEND minimizes cost by suspending compute after a specified idle period.

It prevents unnecessary credit consumption while ensuring compute resumes automatically when needed.

Performance improvements depend on warehouse size, not suspension settings.

Question: 213

Which of the following is true about Snowflake clones?

- A. Clones can only be created from databases, not tables or schemas.
- B. Clones initially consume minimal/no storage, with storage only being used for changes made to the clone or the original.
- C. Clones consume the same storage as the original object.
- D. Clones are automatically synchronized with the original object.

Answer: B

Explanation:

Snowflake cloning is a metadata-only operation at creation. Both the clone and source share micropartitions until changes occur (copy-on-write).

They are independent and do not auto-sync.

They can be made at the database, schema, or table level.

Question: 214

What is the primary benefit of the separation of storage and compute in Snowflake?

- A. It allows independent scaling of resources.
- B. It reduces network latency.
- C. It simplifies data governance.
- D. It eliminates the need for data loading.

Answer: A

Explanation:

Snowflake's architecture separates storage and compute, enabling:

- Compute scaling (up/down or multi-cluster) without changing storage
- Storage expansion without affecting compute
- Cost optimization by paying for compute only when needed

This separation does not impact governance, latency, or data loading requirements.