DUMPSBOSS.

SnowPro Advanced: Architect Certification Exam

Snowflake ARA-C01

Version Demo

Total Demo Questions: 23

Total Premium Questions: 235

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Account and Security	67
Topic 2, Data Architecture	35
Topic 3, Data Engineering	61
Topic 4, Performance Optimization	38
Topic 5, Data Sharing	19
Topic 6, Business Continuity	15
Total	235

QUESTION NO: 1

When loading data into a table that captures the load time in a column with a default value of either CURRENT_TIME() or CURRENT_TIMESTAMP() what will occur?

A. All rows loaded using a specific COPY statement will have varying timestamps based on when the rows were inserted.

B. Any rows loaded using a specific COPY statement will have varying timestamps based on when the rows were read from the source.

C. Any rows loaded using a specific COPY statement will have varying timestamps based on when the rows were created in the source.

D. All rows loaded using a specific COPY statement will have the same timestamp value.

ANSWER: D

Explanation:

All rows loaded using a specific COPY statement will have the same timestamp value. Snowflake evaluates the current-time expression used as the column default in the context of the COPY statement, rather than separately according to the timing of each individual source row. Consequently, when COPY supplies no explicit value for that target column, Snowflake applies the default value consistently to every row inserted by that execution of the statement. This makes a default based on CURRENT_TIMESTAMP() useful for recording a common batch-load timestamp: the rows from one COPY operation can be identified as belonging to the same load event. CURRENT_TIME() follows the same statement-level behavior, but stores only the time portion appropriate to the target column's data type, whereas CURRENT_TIMESTAMP() supplies date, time, and timestamp-related information. The value reflects Snowflake's current session context, including the session time zone for timestamp display and interpretation. If a later COPY statement loads more data, its rows receive the current-time default evaluated for that later statement, producing a distinct load-time value for that separate batch. Snowflake documents current timestamp behavior at CURRENT_TIMESTAMP and column default-expression support in the CREATE TABLE reference.

QUESTION NO: 2

What does a Snowflake Architect need to consider when implementing a Snowflake Connector for Kafka?

A. Every Kafka message is in JSON or Avro format.

B. The default retention time for Kafka topics is 14 days.

C. The Kafka connector supports key pair authentication, OAUTH. and basic authentication (for example, username and password).

D. The Kafka connector will create one table and one pipe to ingest data for each topic. If the connector cannot create the table or the pipe it will result in an exception.

ANSWER: A

Explanation:

Every Kafka message is in JSON or Avro format is the key implementation consideration because the Snowflake Connector for Kafka is designed to ingest records serialized in supported structured formats. An architect must validate the format used by Kafka producers before designing the ingestion path, selecting Kafka Connect converters, and configuring the connector. JSON records can be processed using the connector's JSON handling, while Avro records require the appropriate Avro converter and schema-registry-related configuration where applicable. This decision affects how message values are translated into the Snowflake record representation, how schemas are managed, and whether semi-structured data can be loaded reliably into the target table. Confirming message serialization early is especially important when multiple producers publish to a topic, because the connector's ingestion configuration must be compatible with the records it consumes. Snowflake documents the supported Kafka record formats and connector architecture in its Kafka connector overview. Configuration details, including record converter settings used for JSON and Avro data, are available in the Snowflake Connector for Kafka installation documentation.

QUESTION NO: 3

An Architect needs to create a secure share and add an existing table to the share.

Which privileges are required by the role performing these actions? (Choose three.)

A. `CREATE SHARE` on the account

B. `USAGE` on the database and schema containing the table

C. `SELECT` on the table

D. `INSERT` on the table

E. `CREATE DATABASE` on the account

F. `OPERATE` on the virtual warehouse

ANSWER: A B C

Explanation:

`CREATE SHARE` is required at the account level to create a share object. The role that creates the share becomes its owner and can subsequently manage the share, including adding eligible objects.

The role must also have `USAGE` on the database and schema containing the table, and `SELECT` on the table. The database and schema privileges allow the object to be referenced, while `SELECT` authorizes the role to provide the table through the share. Snowflake requires these privileges to ensure that a provider can share only data to which its active role already has authorized access.

`INSERT` and `CREATE DATABASE` do not grant the ability to add an existing table to a share. See Snowflake's data sharing provider documentation and the access control privileges reference.

QUESTION NO: 4

A company is following the Data Mesh principles, including domain separation, and chose one Snowflake account for its data platform.

An Architect created two data domains to produce two data products. The Architect needs a third data domain that will use both of the data products to create an aggregate data product. The read access to the data products will be granted through a separate role.

Based on the Data Mesh principles, how should the third domain be configured to create the aggregate product if it has been granted the two read roles?

A. Use secondary roles for all users.

B. Create a hierarchy that grants both read roles to the third domain's role.

C. Request a technical ETL user with the sysadmin role.

D. Request that the two data domains share data using the Data Exchange.

ANSWER: B

Explanation:

Create a hierarchy that grants both read roles to the third domain's role. This lets the role responsible for the aggregate data product inherit the required SELECT privileges from each source domain while retaining a distinct domain-owned role for creating, owning, and governing the aggregate product. In Snowflake's RBAC model, privileges granted to a role are inherited by roles above it in the role hierarchy. The aggregate domain role can therefore use its inherited access to read both source products and create its own tables, views, or other published product objects under its domain boundary. This

design supports Data Mesh principles because each producing domain can continue to expose its product through a narrowly scoped read role, rather than giving the consuming domain broad administrative access or requiring a separate data-sharing mechanism within the same account. It also provides a clear auditable privilege path: source-domain read roles supply only consumption access, while the aggregate-domain role owns the newly created aggregate product and can grant access to its consumers independently. Snowflake recommends designing role hierarchies to reflect functional responsibilities and using inherited privileges to simplify controlled access management. See Snowflake's role hierarchy and privilege inheritance documentation and access control overview.

QUESTION NO: 5

Which security, governance, and data protection features require, at a MINIMUM, the Business Critical edition of Snowflake? (Choose two.)

A. Extended Time Travel (up to 90 days)

B. Customer-managed encryption keys through Tri-Secret Secure

C. Periodic rekeying of encrypted data

D. AWS, Azure, or Google Cloud private connectivity to Snowflake

E. Federated authentication and SSO

ANSWER: B D

Explanation:

Customer-managed encryption keys through Tri-Secret Secure requires the Business Critical edition (or higher) because it adds a customer-controlled key to Snowflake's encryption model. Snowflake maintains its own managed key material while the customer controls an additional key in their cloud key-management service. This enables the customer to revoke access to that key if needed, providing an additional level of control over access to encrypted account data. Snowflake documents Tri-Secret Secure as a Business Critical Edition feature and notes that it is available for accounts hosted on supported cloud platforms.

AWS, Azure, or Google Cloud private connectivity to Snowflake also requires, at minimum, Business Critical edition. This capability lets organizations access Snowflake through a private network path rather than the public internet, using the relevant cloud-provider private-connectivity service. Private connectivity supports architectures with network-isolation, regulatory, and controlled-ingress requirements, while still allowing clients and services in the customer's virtual network to reach Snowflake. Snowflake's editions documentation identifies private connectivity as a Business Critical capability, and its connectivity documentation describes the supported private endpoint approaches across cloud providers. See Snowflake editions and features and private connectivity documentation.

QUESTION NO: 6

Which Snowflake objects can be used in a data share? (Select TWO).

A. Standard view

B. Secure view

C. Stored procedure

D. External table

E. Stream

ANSWER: B D

Explanation:

Secure view and **External table** can be included in a Snowflake share. A secure view is designed for controlled data exposure: Snowflake restricts visibility into its definition and query details, allowing a provider to expose only the intended rows and columns to consumers while retaining the underlying source objects in the provider account. This makes secure views a key mechanism for publishing governed or transformed datasets without granting consumers direct access to the base tables.

An external table can also be shared. It represents data stored in external cloud storage and enables the provider to share access to that data through Snowflake metadata and authorization controls. The provider must ensure that the external table and its required supporting objects are configured appropriately so consumers can query the shared object. Snowflake's secure data sharing model lets consumers access shared objects without copying the data into their own account, while the provider continues to manage the shared database objects and permissions.

Snowflake documents the supported object types for shares and the requirements for sharing views and externally stored data in its Introduction to Secure Data Sharing and Share data as a provider documentation.

QUESTION NO: 7

Assuming all Snowflake accounts are using an Enterprise edition or higher, in which development and testing scenarios would be copying of data be required, and zero-copy cloning not be suitable? (Select TWO).

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

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

C. Data is in a production Snowflake account that needs to be provided to Developers in a separate development/testing Snowflake account in the same cloud region.

D. Developers create their own copies of a standard test database previously created for them in the development account, for their initial development and unit testing.

E. The release process requires pre-production testing of changes with data of production scale and complexity. For security reasons, pre-production also runs in the production account.

ANSWER: B C

Explanation:

Production and development run in different databases in the same account, and Developers need to see production-like data but with specific columns masked, requires a separate, physically transformed data set when the requirement is for developers to work only with persistently masked values. Zero-copy cloning creates a metadata-based clone that initially references the source table's existing micro-partitions; it does not itself create a newly materialized version in which sensitive column values have been replaced. A controlled copy process, such as creating a table from a query that applies the required transformation, produces the appropriate protected development data set.

Data is in a production Snowflake account that needs to be provided to Developers in a separate development/testing Snowflake account in the same cloud region, also requires data movement between accounts. Zero-copy cloning is an intra-account operation: the source object and its clone are created within the same Snowflake account. To make production data available in another account, an architect must use a cross-account mechanism such as replication, sharing where suitable for the access pattern, or an explicit unload/load or copy workflow. Replication creates a separate database copy in the target account, enabling an independently administered development or testing environment. See Snowflake's Zero-copy cloning documentation and replication overview.

QUESTION NO: 8

An Architect wants to stream website logs near real time to Snowflake using the Snowflake Connector for Kafka.

What characteristics should the Architect consider regarding the different ingestion methods? (Select TWO).

A. Snowpipe Streaming is the default ingestion method.

B. Snowpipe Streaming supports schema detection.

C. Snowpipe has lower latency than Snowpipe Streaming.

D. Snowpipe Streaming automatically flushes data every one second.

E. Snowflake can handle jumps or resetting offsets by default.

ANSWER: D E

Explanation:

“Snowpipe Streaming automatically flushes data every one second.” is correct because the Snowflake Connector for Kafka uses a short client-lag interval for Snowpipe Streaming ingestion. Its default maximum client lag is one second, which causes buffered records to be committed frequently and makes the data available with near-real-time latency. This direct, row-oriented streaming approach avoids the file-staging and file-loading cycle associated with traditional Snowpipe, making it appropriate for continuously arriving website-log events.

“Snowflake can handle jumps or resetting offsets by default.” is also correct. The connector coordinates Kafka offsets with Snowflake ingestion state, using offset tokens to support reliable processing and recovery. This offset-aware design lets the connector safely accommodate Kafka offset changes, including jumps and resets, while maintaining ingestion continuity. Architects should still design operational monitoring and recovery procedures around Kafka consumer behavior, but offset handling is built into the connector’s normal processing model rather than requiring a custom implementation for each reset event.

See the Snowflake Connector for Kafka overview and the Snowpipe Streaming overview for Snowflake’s ingestion and latency guidance.

QUESTION NO: 9

An Architect needs to meet a company requirement to ingest files from the company's AWS storage accounts into the company's Snowflake Google Cloud Platform (GCP) account. How can the ingestion of these files into the company's Snowflake account be initiated? (Select TWO).

A. Configure the client application to call the Snowpipe REST endpoint when new files have arrived in Amazon S3 storage.

B. Configure the client application to call the Snowpipe REST endpoint when new files have arrived in Amazon S3 Glacier storage.

C. Create an AWS Lambda function to call the Snowpipe REST endpoint when new files have arrived in Amazon S3 storage.

D. Configure AWS Simple Notification Service (SNS) to notify Snowpipe when new files have arrived in Amazon S3 storage.

E. Parquet

ANSWER: A C

Explanation:

“Configure the client application to call the Snowpipe REST endpoint when new files have arrived in Amazon S3 storage.” is correct because the Snowpipe REST API enables an application to explicitly submit file paths for ingestion. This is the appropriate initiation mechanism when files are stored in Amazon S3 but the target Snowflake account is hosted on Google Cloud Platform. The application can detect that an object has been written to S3 and make an authenticated REST request to insert the file metadata into the pipe’s queue.

“Create an AWS Lambda function to call the Snowpipe REST endpoint when new files have arrived in Amazon S3 storage.” is also correct. Lambda can be invoked in response to S3 object-created events and act as the client application that calls Snowpipe’s `insertFiles` REST endpoint. This creates an event-driven ingestion workflow without requiring the Snowflake account itself to receive native S3 auto-ingest notifications. The REST API request supplies the target pipe and one or more staged filenames, after which Snowpipe loads the data using its managed compute resources. Snowflake documents the REST endpoint and request format in the Snowpipe REST API overview, and documents supported cloud storage locations and external stages in Configuring an Amazon S3 storage integration.

QUESTION NO: 10

An Architect needs to improve the performance of reports that pull data from multiple Snowflake tables, join, and then aggregate the data. Users access the reports using several dashboards. There are performance issues on Monday mornings between 9:00am-11:00am when many users check the sales reports.

The size of the group has increased from 4 to 8 users. Waiting times to refresh the dashboards has increased significantly. Currently this workload is being served by a virtual warehouse with the following parameters:

AUTO-RESUME = TRUE AUTO_SUSPEND = 60 SIZE = Medium

What is the MOST cost-effective way to increase the availability of the reports?

A. Use materialized views and pre-calculate the data.

B. Increase the warehouse to size Large and set auto_suspend = 600.

C. Use a multi-cluster warehouse in maximized mode with 2 size Medium clusters.

D. Use a multi-cluster warehouse in auto-scale mode with 1 size Medium cluster, and set min_cluster_count = 1 and max_cluster_count = 4.

ANSWER: D

Explanation:

Use a multi-cluster warehouse in auto-scale mode with 1 size Medium cluster, and set min_cluster_count = 1 and max_cluster_count = 4. This configuration directly addresses the stated problem: dashboard users are submitting concurrent reporting workloads during a predictable, short peak period, and the primary symptom is increased queueing time. A multi-cluster warehouse adds clusters when demand causes queries to queue, allowing more queries to execute concurrently. This improves report availability without requiring the warehouse to run multiple clusters continuously.

With a minimum cluster count of one, the warehouse normally incurs the cost of only one Medium cluster. During the Monday morning surge, Snowflake can automatically start additional Medium clusters, up to four, as needed to absorb concurrent dashboard refreshes. When demand declines, it can reduce the number of running clusters again. This makes auto-scale an appropriate cost-conscious concurrency solution for variable demand. The existing auto-suspend setting also continues to limit compute consumption outside active periods. Snowflake documents that multi-cluster warehouses are designed to support concurrent users and queries, and that auto-scale dynamically starts and stops clusters based on workload demand. See Multi-cluster warehouses and Virtual warehouses.

QUESTION NO: 11

Which steps are recommended best practices for prioritizing cluster keys in Snowflake? (Choose two.)

A. Choose columns that are frequently used in join predicates.

B. Choose lower cardinality columns to support clustering keys and cost effectiveness.

C. Choose TIMESTAMP columns with nanoseconds for the highest number of unique rows.

D. Choose cluster columns that are most actively used in selective filters.

E. Choose cluster columns that are actively used in the GROUP BY clauses.

ANSWER: A D

Explanation:

Choose columns that are frequently used in join predicates. and **Choose cluster columns that are most actively used in selective filters.** are recommended because clustering is intended to improve micro-partition pruning and reduce the amount of data scanned for common, selective access patterns. When a frequently filtered column is part of the clustering key, Snowflake can maintain more useful clustering depth for that predicate, allowing queries to eliminate

irrelevant micro-partitions more effectively. This is particularly valuable for large tables where selective predicates routinely access only a small subset of rows.

Columns used in join predicates can also be appropriate clustering-key candidates, especially when joins repeatedly constrain access to related subsets of a large fact-style table. Co-locating similar key values within micro-partitions can reduce scanning for those recurring join patterns. Snowflake recommends selecting clustering columns according to the workload rather than treating clustering as a universal table-design requirement. The expected benefit should justify the compute resources used by automatic clustering, and query-profile evidence should be used to identify recurring filters and joins that scan substantial data. See Snowflake's clustering key documentation and micro-partitions and data clustering guidance.

QUESTION NO: 12

Company A has recently acquired company

B.

The Snowflake deployment for company B is located in the Azure West Europe region.

As part of the integration process, an Architect has been asked to consolidate company B's sales data into company A's Snowflake account which is located in the AWS us-east-1 region.

How can this requirement be met?

A. Replicate the sales data from company B's Snowflake account into company A's Snowflake account using cross-region data replication within Snowflake. Configure a direct share from company B's account to company A's account.

B. Export the sales data from company B's Snowflake account as CSV files, and transfer the files to company A's Snowflake account. Import the data using Snowflake's data loading capabilities.

C. Migrate company B's Snowflake deployment to the same region as company A's Snowflake deployment, ensuring data locality. Then perform a direct database-to-database merge of the sales data.

D. Build a custom data pipeline using Azure Data Factory or a similar tool to extract the sales data from company B's Snowflake account. Transform the data, then load it into company A's Snowflake account.

ANSWER: B

Explanation:

Exporting the sales data as CSV files, transferring those files, and loading them into Company A's account is a valid cross-cloud, cross-region consolidation method. Snowflake data is loaded from files placed in an internal stage or an external cloud-storage stage that the target account can access. Company B can use `COPY INTO <location>` to unload query results or table data to staged files, after which the files can be transferred to storage accessible to Company A. Company A can then define an appropriate file format and use `COPY INTO <table>` to ingest the data into its target sales tables.

This approach does not depend on both accounts being colocated on the same cloud platform or on establishing a replication topology between the accounts. It is therefore suitable when moving data from an Azure West Europe Snowflake deployment into an AWS us-east-1 Snowflake account. The architect should secure the transfer location, preserve required schema and file-format definitions, and validate row counts and data quality after loading. For recurring consolidation, the same pattern can be automated with scheduled unload, transfer, and load processes. Snowflake documents unloading data with `COPY INTO <location>` and loading staged files with `COPY INTO <table>` at COPY INTO <location> and COPY INTO <table>.

QUESTION NO: 13

Which statements describe a managed access schema in Snowflake? (Choose two.)

A. Object owners cannot grant privileges on objects in the schema.

B. The schema owner can manage privilege grants on objects in the schema.

C. Only ACCOUNTADMIN can create objects in the schema.

D. Object owners cannot alter or drop their own objects.

E. The schema automatically grants SELECT on all future tables to PUBLIC.

ANSWER: A B

Explanation:

In a managed access schema, object owners cannot independently grant privileges on the objects they own. Instead, privilege grants are centrally managed by the schema owner or by a role with the `MANAGE GRANTS` privilege. This model helps organizations apply consistent access-management practices when many teams create objects in the same schema.

Object ownership is still meaningful in a managed access schema. An owner can alter or drop an object, subject to normal ownership behavior, but does not control grants on that object. The schema owner can centrally grant privileges on objects in the schema, including objects created by other roles. See Snowflake's managed access schemas documentation and access control overview.

QUESTION NO: 14 - (SIMULATION)

When loading data from stage using COPY INTO, what options can you specify for the ON_ERROR clause?

ANSWER: See the explanation for the answer

Explanation:

Answer: The valid `ON_ERROR` options for `COPY INTO <table>` are:

`ABORT_STATEMENT` — stop the load when the first error occurs. This is the default.
`CONTINUE` — continue loading valid rows and skip rows containing errors.
`SKIP_FILE` — skip an entire file if any error is encountered in that file.
`SKIP_FILE_<num>` — skip a file when the number of errors reaches the specified number.
`SKIP_FILE_<num>%` — skip a file when the percentage of erroneous rows reaches the specified percentage.

```
COPY INTO target_table
FROM @my_stage
ON_ERROR = 'SKIP_FILE_5%';
```

Use `CONTINUE` for row-level tolerance, and a `SKIP_FILE...` value when a file should be rejected once its error threshold is met.

Reference: COPY INTO <table> — Snowflake Documentation

QUESTION NO: 15

A DevOps team has a requirement for recovery of staging tables used in a complex set of data pipelines. The staging tables are all located in the same staging schema. One of the requirements is to have online recovery of data on a rolling 7-day basis.

After setting up the DATA_RETENTION_TIME_IN_DAYS at the database level, certain tables remain unrecoverable past 1 day.

What would cause this to occur? (Choose two.)

A. The staging schema has not been setup for MANAGED ACCESS.

B. The DATA_RETENTION_TIME_IN_DAYS for the staging schema has been set to 1 day.

C. The tables exceed the 1 TB limit for data recovery.

D. The staging tables are of the TRANSIENT type.

E. The DevOps role should be granted ALLOW_RECOVERY privilege on the staging schema.

ANSWER: B D

Explanation:

The staging schema has a more specific setting than the database when its `DATA_RETENTION_TIME_IN_DAYS` value is explicitly set to 1. Snowflake resolves this parameter according to object hierarchy: an account-level value can be overridden by a database, a database value can be overridden by a schema, and a schema value can be overridden by an individual table. Therefore, tables inheriting a one-day value from the staging schema have only one day of Time Travel retention, despite the database being configured for seven days.

The staging tables being of the `TRANSIENT` type also imposes a one-day maximum Time Travel retention period. Transient tables are intended for non-permanent data and do not support Fail-safe; Snowflake limits their `DATA_RETENTION_TIME_IN_DAYS` setting to 0 or 1. A seven-day rolling online recovery requirement consequently requires permanent tables, together with a retention value configured at the applicable object level. Snowflake documents parameter inheritance and precedence in its parameter reference, and the one-day Time Travel limitation for transient objects in its temporary and transient tables documentation.

QUESTION NO: 16

An Architect is implementing a CI/CD process. When attempting to clone a table from a production to a development environment, the cloning operation fails.

What could be causing this to happen?

A. The table is transient.

B. The table has a masking policy.

C. The retention time for the table is set to zero.

D. Tables cannot be cloned from a higher environment to a lower environment.

ANSWER: B

Explanation:

The table has a masking policy. Snowflake identifies masking policies through policy references, and cloning behavior must preserve the required policy association and its dependencies. When a protected table is cloned as part of a CI/CD promotion or environment-refresh workflow, the masking policy and the objects it depends on must be resolvable in the destination context. If the policy is unavailable, inaccessible, or cannot be correctly associated during the clone operation, the table clone can fail. Architects should therefore deploy and validate the policy definitions, required privileges, and any referenced governance objects as part of the environment's deployment sequence before cloning protected tables. This is particularly important when production and development use separate databases or schemas, because object references and role permissions need to be designed deliberately for the target environment. Snowflake documents that cloning copies object metadata and that governance policies require special consideration when cloning databases and schemas. Review the cloning guidance and masking-policy administration guidance when building automated release processes: Snowflake object cloning documentation and Snowflake column-level security and masking policies documentation.

QUESTION NO: 17

The following statements have been executed successfully:

USE ROLE SYSADMIN;

CREATE OR REPLACE DATABASE DEV_TEST_DB;

CREATE OR REPLACE SCHEMA DEV_TEST_DB.SCHTEST WITH MANAGED ACCESS;

GRANT USAGE ON DATABASE DEV_TEST_DB TO ROLE DEV_PROJ_OWN;

GRANT USAGE ON SCHEMA DEV_TEST_DB.SCHTEST TO ROLE DEV_PROJ_OWN;

GRANT USAGE ON DATABASE DEV_TEST_DB TO ROLE ANALYST_PROJ;

GRANT USAGE ON SCHEMA DEV_TEST_DB.SCHTEST TO ROLE ANALYST_PROJ;

GRANT CREATE TABLE ON SCHEMA DEV_TEST_DB.SCHTEST TO ROLE DEV_PROJ_OWN;

USE ROLE DEV_PROJ_OWN;

CREATE OR REPLACE TABLE DEV_TEST_DB.SCHTEST.CURRENCY (

COUNTRY VARCHAR(255),

CURRENCY_NAME VARCHAR(255),

ISO_CURRENCY_CODE VARCHAR(15),

CURRENCY_CD NUMBER(38,0),

MINOR_UNIT VARCHAR(255),

WITHDRAWAL_DATE VARCHAR(255)

);

The role hierarchy is as follows (simplified from the diagram):

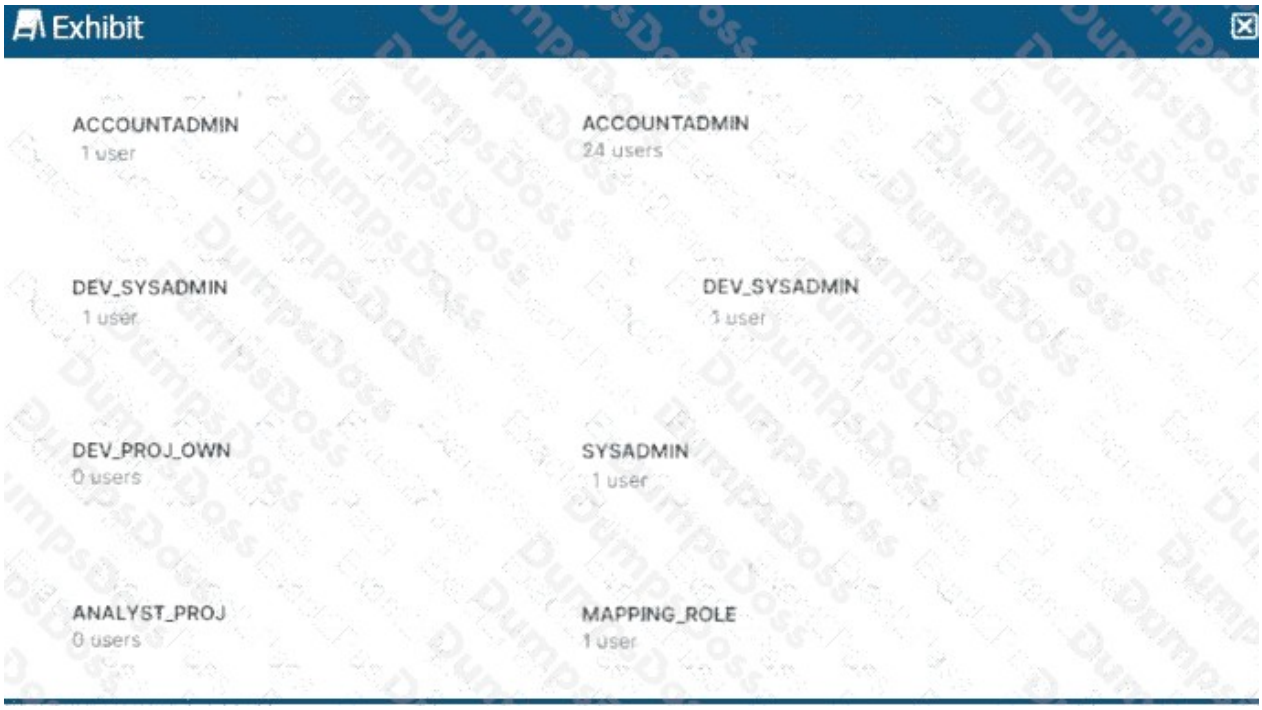
ACCOUNTADMIN └─ DEV_SYSADMIN └─ DEV_PROJ_OWN └─ ANALYST_PROJ

Separately:

ACCOUNTADMIN └─ SYSADMIN └─ MAPPING_ROLE

Which statements will return the records from the table

DEV_TEST_DB.SCHTEST.CURRENCY? (**Select TWO**)



A. USE ROLE DEV_PROJ_OWN;GRANT SELECT ON DEV_TEST_DB.SCHTEST.CURRENCY TO ROLE ANALYST_PROJ;USE ROLE ANALYST_PROJ;SELECT * FROM DEV_TEST_DB.SCHTEST.CURRENCY;

B. USE ROLE DEV_PROJ_OWN;SELECT * FROM DEV_TEST_DB.SCHTEST.CURRENCY;

C. USE ROLE SYSADMIN;SELECT * FROM DEV_TEST_DB.SCHTEST.CURRENCY;

D. USE ROLE MAPPING_ROLE;SELECT * FROM DEV_TEST_DB.SCHTEST.CURRENCY;

E. USE ROLE ACCOUNTADMIN;SELECT * FROM DEV_TEST_DB.SCHTEST.CURRENCY;

ANSWER: B E

Explanation:

“USE ROLE DEV_PROJ_OWN; SELECT * FROM DEV_TEST_DB.SCHTEST.CURRENCY;” returns rows because DEV_PROJ_OWN created the CURRENCY table and therefore owns it. Object ownership gives the owning role full control of the object, including the ability to query it. DEV_PROJ_OWN also has the required USAGE privileges on both the DEV_TEST_DB database and the SCHTEST schema, so the fully qualified table reference can be resolved.

“USE ROLE ACCOUNTADMIN; SELECT * FROM DEV_TEST_DB.SCHTEST.CURRENCY;” also returns rows. In the supplied hierarchy, privileges held by DEV_PROJ_OWN are inherited by DEV_SYSADMIN and then by ACCOUNTADMIN. Consequently, ACCOUNTADMIN inherits the ownership-derived privileges on CURRENCY as well as the required database and schema access. Snowflake’s role hierarchy is additive upward: a role inherits the privileges of roles granted to it. The fact that SCHTEST is a managed access schema does not remove the table owner’s own access; managed access centralizes the ability to grant object privileges to other roles with the schema owner or a role holding MANAGE GRANTS. See Snowflake’s role hierarchy and privilege inheritance documentation and its managed access schema documentation.

QUESTION NO: 18

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

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

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

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

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

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

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

ANSWER: A B C

Explanation:

Row access policies can protect data exposed through an external table, with support focused on the external table’s `VALUE` column. An external table may be created with a row access policy assigned to `VALUE`, allowing access to the semi-structured record content to be evaluated according to the policy at query time. Snowflake also supports adding a row access policy to the `VALUE` column after an external table already exists, using the appropriate `ALTER TABLE` syntax.

Virtual columns in an external table are derived from expressions over the underlying external data rather than being independently stored table columns. Consequently, a row access policy cannot be attached directly to a virtual column. To govern access through a row access policy for an external table, apply the policy to `VALUE`; policy logic can then use supported expressions and context functions to determine which rows are available to the querying user or role. These capabilities let architects apply centralized, policy-based row filtering to externally stored data while respecting the column-model limitations specific to external tables. Snowflake documents these external-table considerations in its row access policy overview and CREATE EXTERNAL TABLE reference.

QUESTION NO: 19

Which SQL alter command will MAXIMIZE memory and compute resources for a Snowpark stored procedure when executed on the snowpark_opt_wh warehouse?

A)

```
alter warehouse snowpark_opt_wh set max_concurrency_level = 1;
```

```
alter warehouse snowpark_opt_wh set max_concurrency_level = 2;
```

```
alter warehouse snowpark_opt_wh set max_concurrency_level = 8;
```

```
alter warehouse snowpark_opt_wh set max_concurrency_level = 16;
```

A. Option A

B. Option B

C. Option C

D. Option D

E. ALTER WAREHOUSE snowpark_opt_wh SET MAX_CONCURRENCY_LEVEL = 1;

ANSWER: E

Explanation:

`ALTER WAREHOUSE snowpark_opt_wh SET MAX_CONCURRENCY_LEVEL = 1;` is the configuration that maximizes the memory and compute capacity available to an individual Snowpark stored-procedure execution. `MAX_CONCURRENCY_LEVEL` controls the target number of concurrently executing statements on a warehouse. Setting it to `1` prevents the warehouse from scheduling multiple statements to share its available processing and memory resources, allowing the Snowpark workload to use the greatest possible share of the warehouse resources. This is especially relevant to memory-intensive Snowpark processing, such as data science, machine learning, or large in-memory transformations, where maximizing resources for one execution is more important than throughput for many simultaneous queries.

A Snowpark-optimized warehouse is designed to provide substantially more memory per node than a standard warehouse. Combining that warehouse type with a concurrency level of one is the appropriate configuration when a stored procedure needs maximum per-query resources. Snowflake documents that a lower concurrency level allocates more warehouse resources to each running query, and its Snowpark-optimized warehouse guidance specifically identifies these warehouses for memory-intensive Snowpark workloads. See the Snowflake documentation for MAX_CONCURRENCY_LEVEL and Snowpark-optimized warehouses.

QUESTION NO: 20

A user named USER_01 needs access to create a materialized view on a schema EDW. STG_SCHEMA. How can this access be provided?

A. GRANT CREATE MATERIALIZED VIEW ON SCHEMA EDW.STG_SCHEMA TO USER USER_01;

B. GRANT CREATE MATERIALIZED VIEW ON DATABASE EDW TO USER USERJD1;

C. GRANT ROLE NEW_ROLE TO USER USER_01;GRANT CREATE MATERIALIZED VIEW ON SCHEMA ECW.STG_SCHEKA TO NEW_ROLE;

D. GRANT CREATE MATERIALIZED VIEW ON SCHEMA EDW.STG_SCHEMA TO ROLE NEW_ROLE; GRANT ROLE NEW_ROLE TO USER USER_01;

ANSWER: D

Explanation:

`GRANT CREATE MATERIALIZED VIEW ON SCHEMA EDW.STG_SCHEMA TO ROLE NEW_ROLE; GRANT ROLE NEW_ROLE TO USER USER_01;` is correct because Snowflake's role-based access-control model grants object privileges to roles and then assigns those roles to users. The `CREATE MATERIALIZED VIEW` schema privilege authorizes the role to create materialized views in `EDW.STG_SCHEMA`. Assigning `NEW_ROLE` to `USER_01` makes that privilege available to the user when the role is active. In a production implementation, the role must already exist, or it should first be created with `CREATE ROLE NEW_ROLE`. The user also needs the role to be available through their active role hierarchy; if it is not their default role, they can activate it with `USE ROLE NEW_ROLE`. Snowflake documents `CREATE MATERIALIZED VIEW` as a schema-level privilege, and its access-control guidance establishes roles as the entities to which privileges are granted and which are subsequently granted to users. See the GRANT <privileges> documentation and the access control overview.

QUESTION NO: 21

Role A has the following permissions:

. USAGE on db1

. USAGE and CREATE VIEW on schemal in db1

. SELECT on tablel in schemal

Role B has the following permissions:

. USAGE on db2

. USAGE and CREATE VIEW on schema2 in db2

. SELECT on table2 in schema2

A user has Role A set as the primary role and Role B as a secondary role.

What command will fail for this user?

A. use database db1;use schema schemal;create view v1 as select * from db2.schema2.table2;

B. use database db2;use schema schema2;create view v2 as select * from db1.schema1.table1;

C. use database db2;use schema schema2;select * from db1.schemal.tablel union select * from table2;

D. use database db1;use schema schemal;select * from db2.schema2.table2;

ANSWER: B

Explanation:

`use database db2; use schema schema2; create view v2 as select * from db1.schema1.table1;` will fail because Snowflake evaluates object-creation authorization using the active primary role. Here, Role A is the primary role, and it does not have USAGE on `db2`, USAGE on `schema2`, or the CREATE VIEW privilege in `schema2`. Although Role B is enabled as a secondary role and its privileges can be used for many operations, secondary roles are not considered for authorization of CREATE statements. Consequently, Role B's privileges cannot authorize creation of `v2` in `db2.schema2`. The created object, if creation were authorized, would also be owned by the active primary role, which is why Snowflake applies this restriction. Snowflake documents that the primary role is used to authorize CREATE statements, while secondary roles are not considered for object creation. See the Snowflake documentation on secondary roles and the USE SECONDARY ROLES reference.

QUESTION NO: 22

An Architect is using an event table associated with a Sales database (sales_db) to track logging and tracing of procedures and functions. The event table is also used to refresh dynamic tables.

A stored procedure causing issues resides in the Marketing database (marketing_db). Both databases are in the same Snowflake account. The Marketing database is not associated with a specific event table.

How can the Architect investigate the issue?

A. Add a new event table and associate it with the Marketing database.

B. Add a new event table and associate it with the Snowflake account.

C. Query the event table SNOWFLAKE.TELEMETRY.EVENTS.

D. Query the event table MARKETING_DB.TELEMETRY.EVENTS.

ANSWER: C

Explanation:

Query the event table SNOWFLAKE.TELEMETRY.EVENTS. Snowflake event tables store telemetry emitted by supported features, including log messages, trace events, and metrics from handler code in stored procedures and functions. Event-table assignment can occur at the database level. When a database has no event table assigned, telemetry generated by objects in that database is directed to the account's default event table rather than to an event table assigned to a different database.

Because the failing procedure belongs to `marketing_db` and that database has no database-level event-table association, its emitted telemetry can be investigated through the account-level default event table, exposed as `SNOWFLAKE.TELEMETRY.EVENTS`. The Architect can filter this event data by timestamp, severity, record type, and attributes associated with the procedure execution to locate error logs and trace details. Appropriate privileges to query the shared `SNOWFLAKE` database and access telemetry data are also required. This approach supports immediate investigation without changing the telemetry configuration or creating additional objects. See Snowflake's event table setup documentation and the reference for telemetry events.

QUESTION NO: 23

Which Snowflake data modeling approach is designed for BI queries?

A. 3 NF

B. Star schema

C. Data Vault

D. Snowflake schema

ANSWER: B

Explanation:

Star schema is the appropriate modeling approach for BI queries. It organizes analytical data around a central fact table containing measurable business events, such as sales amounts or quantities, and surrounding dimension tables that provide descriptive context, such as customer, product, date, or location. This layout maps naturally to common BI workloads: users filter and group by descriptive attributes and aggregate measures from the fact table. Because the dimensions typically join directly to the fact table, the model is intuitive for report authors and semantic-layer tools, while also supporting efficient SQL patterns for dashboards, ad hoc analysis, and recurring aggregate queries.

Snowflake supports dimensional models and does not require data to be modeled in a particular way, but its guidance recognizes star schemas as a common approach for analytical and reporting use cases. The approach emphasizes understandable business-facing data structures and predictable fact-to-dimension joins, which is why it is widely used in BI-oriented warehouses. For Snowflake guidance on data modeling and dimensional-model concepts, see Snowflake documentation on creating tables and CTAS and Snowflake key concepts documentation.