

Snowflake

Exam Questions DEA-C01

SnowPro Advanced: Data Engineer Certification Exam



NEW QUESTION 1

A Data Engineer defines the following masking policy:

```
current_role() IN ('ADMIN') THEN val  
*****!
```

....

must be applied to the full_name column in the customer table:

```
TABLE customer(  
  
name VARCHAR,  
name VARCHAR,  
name VARCHAR AS CONCAT(first_name, ' ', last_name)
```

Which query will apply the masking policy on the full_name column?

- A. ALTER TABLE customer MODIFY COLUMN full_name Set MASKING POLICY name_policy;
- B. ALTER TABLE customer MODIFY COLUMN full_nam ADD MASKING POLICYname_poiiicy;
- C. ALTER TABLE customer MODIFY COLUMN first_nane SET MASKING POLICY name_policy; lasT_name SET MASKING POLICY name_pclicy;
- D. ALTER TABLE customer MODIFY COLUMN first_name ADD MASKING POLICY name_policy,

Answer: A

Explanation:

The query that will apply the masking policy on the full_name column is ALTER TABLE customer MODIFY COLUMN full_name SET MASKING POLICY name_policy;. This query will modify the full_name column and associate it with the name_policy masking policy, which will mask the first and last names of the customers with asterisks. The other options are incorrect because they do not follow the correct syntax for applying a masking policy on a column. Option B is incorrect because it uses ADD instead of SET, which is not a valid keyword for modifying a column. Option C is incorrect because it tries to apply the masking policy on two columns, first_name and last_name, which are not part of the table structure. Option D is incorrect because it uses commas instead of dots to separate the database, schema, and table names

NEW QUESTION 2

Which callback function is required within a JavaScript User-Defined Function (UDF) for it to execute successfully?

- A. initialize ()
- B. processRow ()
- C. handler
- D. finalize ()

Answer: B

Explanation:

The processRow () callback function is required within a JavaScript UDF for it to execute successfully. This function defines how each row of input data is processed and what output is returned. The other callback functions are optional and can be used for initialization, finalization, or error handling.

NEW QUESTION 3

A Data Engineer ran a stored procedure containing various transactions During the execution, the session abruptly disconnected preventing one transactionfrom committing or rolling hark.The transaction was left in a detached state and created a lock on resources ...must the Engineer take to immediately run a new transaction?

- A. Call the system function SYSTEM\$ABORT_TRANSACTION.
- B. Call the system function SYSTEM\$CANCEL_TRANSACTION.
- C. Set the LOCK_TIMEOUTto FALSE in the stored procedure
- D. Set the transaction abort on error to true in the stored procedure.

Answer: A

Explanation:

The system function SYSTEM\$ABORT_TRANSACTION can be used to abort a detached transaction that was left in an open state due to a session disconnect or termination. The function takes one argument: the transaction ID of the detached transaction. The function will abort the transaction and release any locks held by it. The other options are incorrect because they do not address the issue of a detached transaction. The system function SYSTEM\$CANCEL_TRANSACTION can be used to cancel a running transaction, but not a detached one. The LOCK_TIMEOUT parameter can be used to set a timeout period for acquiring locks on resources, but it does not affect existing locks. The TRANSACTION_ABORT_ON_ERROR parameter can be used to control whether a transaction should abort or continue when an error occurs, but it does not affect detached transactions.

NEW QUESTION 4

Which query will show a list of the 20 most recent executions of a specified task ktask, that have been scheduled within the last hour that have ended or are stillrunning's.

A)

```
select * from table(information_schema.task_history(scheduled_time_range_start  
=>dateadd('hour',-1,current_timestamp()), result_limit => 20,  
task_name=>'MYTASK'))
```

B)

```
select * from table(information_schema.task_history(scheduled_time_range_start
=>dateadd('hour',-1,current_timestamp()), result_limit => 20,
task_name=>'MYTASK')) where query_id IS NOT NULL;
```

C)

```
select * from table(information_schema.task_history(scheduled_time_range_start
=>dateadd('hour',-1,current_timestamp()), result_limit => 20,
task_name=>'MYTASK')) where STATE IN ('EXECUTING', 'SUCCEEDED', 'FAILED')
```

D)

```
select * from table(information_schema.task_history(scheduled_time_range_end
=>dateadd('hour',-1,current_timestamp()), result_limit => 10,
task_name=>'MYTASK')) where STATE IN ('EXECUTING', 'SUCCEEDED')
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B**NEW QUESTION 5**

What kind of Snowflake integration is required when defining an external function in Snowflake?

- A. API integration
- B. HTTP integration
- C. Notification integration
- D. Security integration

Answer: A**Explanation:**

An API integration is required when defining an external function in Snowflake. An API integration is a Snowflake object that defines how Snowflake communicates with an external service via HTTPS requests and responses. An API integration specifies parameters such as URL, authentication method, encryption settings, request headers, and timeout values. An API integration is used to create an external function object that invokes the external service from within SQL queries.

NEW QUESTION 6

Which stages support external tables?

- A. Internal stages only; within a single Snowflake account
- B. internal stages only from any Snowflake account in the organization
- C. External stages only from any region, and any cloud provider
- D. External stages only, only on the same region and cloud provider as the Snowflake account

Answer: C**Explanation:**

External stages only from any region, and any cloud provider support external tables. External tables are virtual tables that can query data from files stored in external stages without loading them into Snowflake tables. External stages are references to locations outside of Snowflake, such as Amazon S3 buckets, Azure Blob Storage containers, or Google Cloud Storage buckets. External stages can be created from any region and any cloud provider, as long as they have a valid URL and credentials. The other options are incorrect because internal stages do not support external tables. Internal stages are locations within Snowflake that can store files for loading or unloading data. Internal stages can be user stages, table stages, or named stages.

NEW QUESTION 7

What is a characteristic of the use of external tokenization?

- A. Secure data sharing can be used with external tokenization
- B. External tokenization cannot be used with database replication
- C. Pre-loading of unmasked data is supported with external tokenization
- D. External tokenization allows the preservation of analytical values after de-identification

Answer: D**Explanation:**

External tokenization is a feature in Snowflake that allows users to replace sensitive data values with tokens that are generated and managed by an external service. External tokenization allows the preservation of analytical values after de-identification, such as preserving the format, length, or range of the original values. This way, users can perform analytics on the tokenized data without compromising the security or privacy of the sensitive data.

NEW QUESTION 8

A Data Engineer wants to centralize grant management to maximize security. A user needs ownership on a table in a new schema. However, this user should not have the ability to make grant decisions. What is the correct way to do this?

- A. Grant ownership to the user on the table
- B. Revoke grant decisions from the user on the table
- C. Revoke grant decisions from the user on the schema.
- D. Add the with managed access parameter on the schema

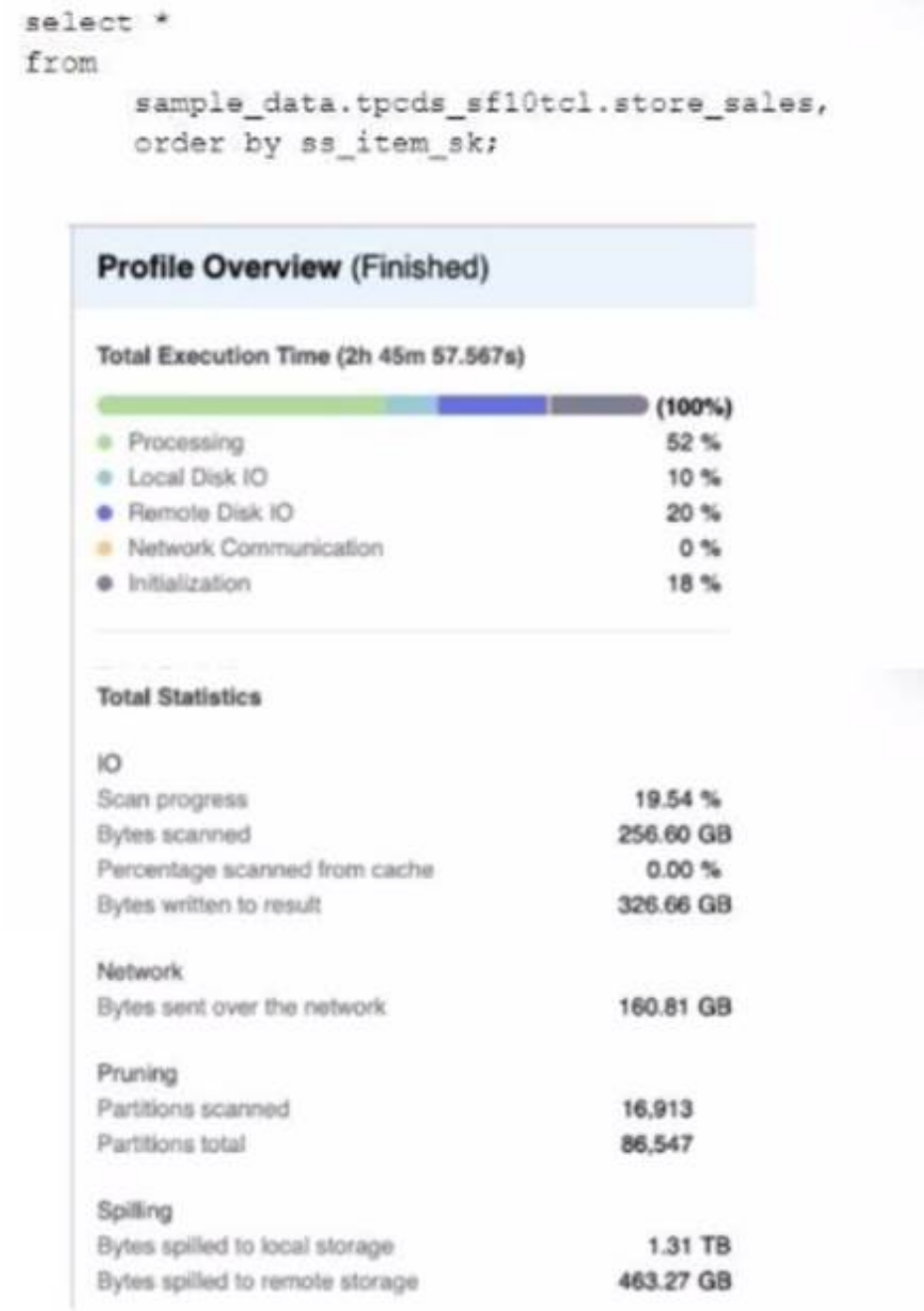
Answer: D

Explanation:

The with managed access parameter on the schema enables the schema owner to control the grant and revoke privileges on the objects within the schema. This way, the user who owns the table cannot make grant decisions, but only the schema owner can. This is the best way to centralize grant management and maximize security.

NEW QUESTION 9

A Data Engineer is evaluating the performance of a query in a development environment.



Based on the Query Profile what are some performance tuning options the Engineer can use? (Select TWO)

- A. Add a LIMIT to the ORDER BY If possible
- B. Use a multi-cluster virtual warehouse with the scaling policy set to standard
- C. Move the query to a larger virtual warehouse
- D. Create indexes to ensure sorted access to data
- E. Increase the max cluster count

Answer: AC

Explanation:

The performance tuning options that the Engineer can use based on the Query Profile are:

? Add a LIMIT to the ORDER BY If possible: This option will improve performance by reducing the amount of data that needs to be sorted and returned by the query. The ORDER BY clause requires sorting all rows in the input before returning them, which can be expensive and time-consuming. By adding a LIMIT clause, the query can return only a subset of rows that satisfy the order criteria, which can reduce sorting time and network transfer time.

? Create indexes to ensure sorted access to data: This option will improve performance by reducing the amount of data that needs to be scanned and filtered by the query. The query contains several predicates on different columns, such as o_orderdate, o_orderpriority, l_shipmode, etc. By creating indexes on these columns, the query can leverage sorted access to data and prune unnecessary micro-partitions or rows that do not match the predicates. This can reduce IO time and processing time.

The other options are not optimal because:

? Use a multi-cluster virtual warehouse with the scaling policy set to standard: This option will not improve performance, as the query is already using a multi-cluster virtual warehouse with the scaling policy set to standard. The Query Profile shows that the query is using a 2XL warehouse with 4 clusters and a standard scaling policy, which means that the warehouse can automatically scale up or down based on the load. Changing the warehouse size or the number of clusters will not affect the performance of this query, as it is already using the optimal resources.

? Increase the max cluster count: This option will not improve performance, as the query is not limited by the max cluster count. The max cluster count is a parameter that specifies the maximum number of clusters that a multi-cluster virtual warehouse can scale up to. The Query Profile shows that the query is using a 2XL warehouse with 4 clusters and a standard scaling policy, which means that the warehouse can automatically scale up or down based on the load. The default max cluster count for a 2XL warehouse is 10, which means that the warehouse can scale up to 10 clusters if needed. However, the query does not need more than 4 clusters, as it is not CPU-bound or memory-bound. Increasing the max cluster count will not affect the performance of this query, as it will not use more clusters than necessary.

NEW QUESTION 10

A CSV file around 1 TB in size is generated daily on an on-premise server. A corresponding table, internal stage, and file format have already been created in Snowflake to facilitate the data loading process.

How can the process of bringing the CSV file into Snowflake be automated using the LEAST amount of operational overhead?

- A. Create a task in Snowflake that executes once a day and runs a copy into statement that references the internal stage. The internal stage will read the files directly from the on-premise server and copy the newest file into the table from the on-premise server to the Snowflake table.
- B. On the on-premise server, schedule a SQL file to run using SnowSQL that executes a PUT to push a specific file to the internal stage. Create a task that executes once a day in Snowflake and runs a COPY INTO statement that references the internal stage. Schedule the task to start after the file lands in the internal stage.
- C. On the on-premise server, schedule a SQL file to run using SnowSQL that executes a PUT to push a specific file to the internal stage.
- D. Create a pipe that runs a copy into statement that references the internal stage. Snowpipe auto-ingest will automatically load the file from the internal stage when the new file lands in the internal stage.
- E. On the on-premise server, schedule a Python file that uses the Snowpark Python library. The Python script will read the CSV data into a DataFrame and generate an insert into statement that will directly load into the table. The script will bypass the need to move a file into an internal stage.

Answer: C

Explanation:

This option is the best way to automate the process of bringing the CSV file into Snowflake with the least amount of operational overhead. SnowSQL is a command-line tool that can be used to execute SQL statements and scripts on Snowflake. By scheduling a SQL file that executes a PUT command, the CSV file can be pushed from the on-premise server to the internal stage in Snowflake. Then, by creating a pipe that runs a COPY INTO statement that references the internal stage, Snowpipe can automatically load the file from the internal stage into the table when it detects a new file in the stage. This way, there is no need to manually start or monitor a virtual warehouse or task.

NEW QUESTION 10

A Data Engineer has developed a dashboard that will issue the same SQL select clause to Snowflake every 12 hours.

---will Snowflake use the persisted query results from the result cache provided that the underlying data has not changed?

- A. 12 hours
- B. 24 hours
- C. 14 days
- D. 31 days

Answer: C

Explanation:

Snowflake uses the result cache to store the results of queries that have been executed recently. The result cache is maintained at the account level and is shared across all sessions and users. The result cache is invalidated when any changes are made to the tables or views referenced by the query. Snowflake also has a retention policy for the result cache, which determines how long the results are kept in the cache before they are purged. The default retention period for the result cache is 24 hours, but it can be changed at the account, user, or session level. However, there is a maximum retention period of 14 days for the result cache, which cannot be exceeded. Therefore, if the underlying data has not changed, Snowflake will use the persisted query results from the result cache for up to 14 days.

NEW QUESTION 15

A company built a sales reporting system with Python, connecting to Snowflake using the Python Connector. Based on the user's selections, the system generates the SQL queries needed to fetch the data for the report. First, it gets the customers that meet the given query parameters (on average 1000 customer records for each report run) and then it loops the customer records sequentially. Inside that loop, it runs the generated SQL clause for the current customer to get the detailed data for that customer number from the sales data table.

When the Data Engineer tested the individual SQL clauses, they were fast enough (1 second to get the customers, 0.5 second to get the sales data for one customer) but the total runtime of the report is too long.

How can this situation be improved?

- A. Increase the size of the virtual warehouse.
- B. Increase the number of maximum clusters of the virtual warehouse.
- C. Define a clustering key for the sales data table.
- D. Rewrite the report to eliminate the use of the loop construct.

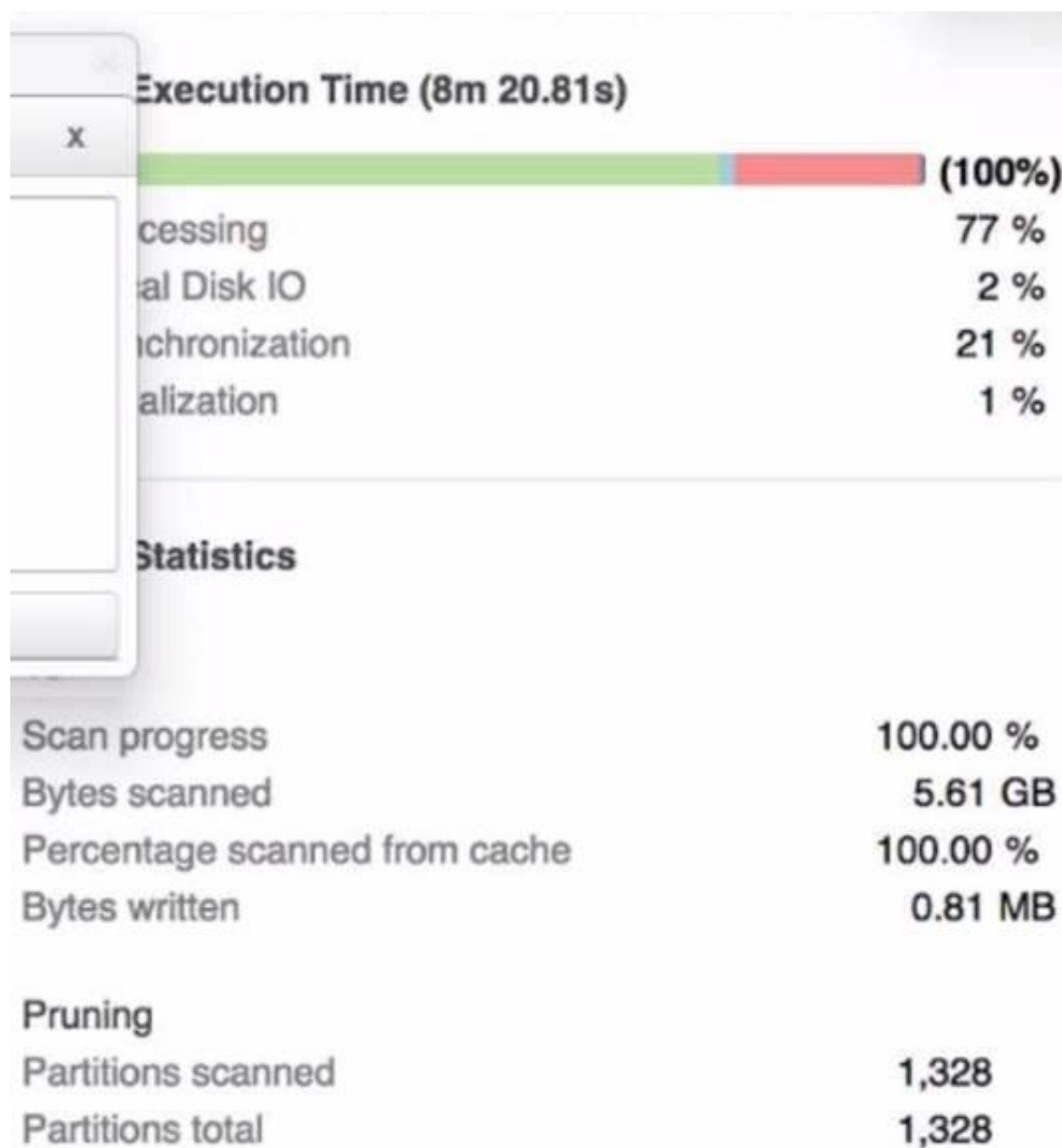
Answer: D

Explanation:

This option is the best way to improve the situation, as using a loop construct to run SQL queries for each customer is very inefficient and slow. Instead, the report should be rewritten to use a single SQL query that joins the customer and sales data tables and applies the query parameters as filters. This way, the report can leverage Snowflake's parallel processing and optimization capabilities and reduce the network overhead and latency.

NEW QUESTION 19

A Data Engineer is investigating a query that is taking a long time to return. The Query Profile shows the following:



What step should the Engineer take to increase the query performance?

- A. Add additional virtual warehouses.
- B. increase the size of the virtual warehouse.
- C. Rewrite the query using Common Table Expressions (CTEs)
- D. Change the order of the joins and start with smaller tables first

Answer: B

Explanation:

The step that the Engineer should take to increase the query performance is to increase the size of the virtual warehouse. The Query Profile shows that most of the time was spent on local disk IO, which indicates that the query was reading a lot of data from disk rather than from cache. This could be due to a large amount of data being scanned or a low cache hit ratio. Increasing the size of the virtual warehouse will increase the amount of memory and cache available for the query, which could reduce the disk IO time and improve the query performance. The other options are not likely to increase the query performance significantly. Option A, adding additional virtual warehouses, will not help unless they are used in a multi-cluster warehouse configuration or for concurrent queries. Option C, rewriting the query using Common Table Expressions (CTEs), will not affect the amount of data scanned or cached by the query. Option D, changing the order of the joins and starting with smaller tables first, will not reduce the disk IO time unless it also reduces the amount of data scanned or cached by the query.

NEW QUESTION 23

A table is loaded using Snowpipe and truncated afterwards Later, a Data Engineer finds that the table needs to be reloaded but the metadata of the pipe will not allow the same files to be loaded again.

How can this issue be solved using the LEAST amount of operational overhead?

- A. Wait until the metadata expires and then reload the file using Snowpipe
- B. Modify the file by adding a blank row to the bottom and re-stage the file
- C. Set the FORCE=TRUE option in the Snowpipe COPY INTO command
- D. Recreate the pipe by using the create or replace pipe command

Answer: C

Explanation:

The FORCE=TRUE option in the Snowpipe COPY INTO command allows Snowpipe to load files that have already been loaded before, regardless of the metadata. This is the easiest way to reload the same files without modifying them or recreating the pipe.

NEW QUESTION 28

A Data Engineer is working on a Snowflake deployment in AWS eu-west-1 (Ireland). The Engineer is planning to load data from staged files into target tables using the copy into command

Which sources are valid? (Select THREE)

- A. Internal stage on GCP us-central1 (Iowa)
- B. Internal stage on AWS eu-central-1 (Frankfurt)
- C. External stage on GCP us-central1 (Iowa)
- D. External stage in an Amazon S3 bucket on AWS eu-west-1 (Ireland)

- E. External stage in an Amazon S3 bucket on AWS eu-central 1 (Frankfurt)
- F. SSO attached to an Amazon EC2 instance on AWS eu-west-1 (Ireland)

Answer: CDE

Explanation:

The valid sources for loading data from staged files into target tables using the copy into command are:

? External stage on GCP us-central1 (Iowa): This is a valid source because Snowflake supports cross-cloud data loading from external stages on different cloud platforms and regions than the Snowflake deployment.

? External stage in an Amazon S3 bucket on AWS eu-west-1 (Ireland): This is a valid source because Snowflake supports data loading from external stages on the same cloud platform and region as the Snowflake deployment.

? External stage in an Amazon S3 bucket on AWS eu-central 1 (Frankfurt): This is a valid source because Snowflake supports cross-region data loading from external stages on different regions than the Snowflake deployment within the same cloud platform. The invalid sources are:

? Internal stage on GCP us-central1 (Iowa): This is an invalid source because internal stages are always located on the same cloud platform and region as the Snowflake deployment. Therefore, an internal stage on GCP us-central1 (Iowa) cannot be used for a Snowflake deployment on AWS eu-west-1 (Ireland).

? Internal stage on AWS eu-central-1 (Frankfurt): This is an invalid source because internal stages are always located on the same region as the Snowflake deployment. Therefore, an internal stage on AWS eu-central-1 (Frankfurt) cannot be used for a Snowflake deployment on AWS eu-west-1 (Ireland).

? SSO attached to an Amazon EC2 instance on AWS eu-west-1 (Ireland): This is an invalid source because SSO stands for Single Sign-On, which is a security integration feature in Snowflake, not a data staging option.

NEW QUESTION 33

How can the following relational data be transformed into semi-structured data using the LEAST amount of operational overhead?

```
create table provinces (province varchar, created_date date);
```

Row	PROVINCE	CREATED_DATE
2	Alberta	2020-01-19
1	Manitoba	2020-01-18

- A. Use the to_json function
- B. Use the PAESE_JSON function to produce a variant value
- C. Use the OBJECT_CONSTRUCT function to return a Snowflake object
- D. Use the TO_VARIANT function to convert each of the relational columns to VARIANT.

Answer: C

Explanation:

This option is the best way to transform relational data into semi-structured data using the least amount of operational overhead. The OBJECT_CONSTRUCT function takes a variable number of key-value pairs as arguments and returns a Snowflake object, which is a variant type that can store JSON data. The function can be used to convert each row of relational data into a JSON object with the column names as keys and the column values as values.

NEW QUESTION 37

A Data Engineer wants to check the status of a pipe named my_pipe. The pipe is inside a database named test and a schema named Extract (case-sensitive). Which query will provide the status of the pipe?

- A. SELECT FROM SYSTEM\$PIPE_STATUS ('test.'extract'.my_pipe');
- B. SELECT FROM SYSTEM\$PIPE_STATUS (,test,,Extracr,,ny_pipe, i l
- C. SELE2T * FROM SYSTEM\$PIPE_STATUS < ' tes
- D. "Extract", my_pipe');
- E. SELECT * FROM SYSTEM\$PIPE_STATUS ("tes
- F. 'extract' .my_pipe");

Answer: C

Explanation:

The query that will provide the status of the pipe is SELECT * FROM SYSTEM\$PIPE_STATUS('test."Extract".my_pipe');. The SYSTEM\$PIPE_STATUS function returns information about a pipe, such as its name, status, last received message timestamp, etc. The function takes one argument: the pipe name in a qualified form. The pipe name should include the database name, the schema name, and the pipe name, separated by dots. If any of these names are case-sensitive identifiers, they should be enclosed in double quotes. In this case, the schema name Extract is case-sensitive and should be quoted. The other options are incorrect because they do not follow the correct syntax for the pipe name argument. Option A and B use single quotes instead of double quotes for case-sensitive identifiers. Option D uses double quotes instead of single quotes for non-case-sensitive identifiers.

NEW QUESTION 41

Assuming that the session parameter USE_CACHED_RESULT is set to false, what are characteristics of Snowflake virtual warehouses in terms of the use of Snowpark?

- A. Creating a DataFrame from a table will start a virtual warehouse
- B. Creating a DataFrame from a staged file with the read () method will start a virtual warehouse
- C. Transforming a DataFrame with methods like replace () will start a virtual warehouse -
- D. Calling a Snowpark stored procedure to query the database with session, call () will start a virtual warehouse

Answer: A

Explanation:

Creating a DataFrame from a table will start a virtual warehouse because it requires reading data from Snowflake. The other options will not start a virtual warehouse because they either operate on local data or use an existing session to query Snowflake.

NEW QUESTION 46

Which output is provided by both the `SYSTEM$CLUSTERING_DEPTH` function and the `SYSTEM$CLUSTERING_INFORMATION` function?

- A. `average_depth`
- B. `notes`
- C. `average_overlaps`
- D. `total_partition_count`

Answer: A

Explanation:

The output that is provided by both the `SYSTEM$CLUSTERING_DEPTH` function and the `SYSTEM$CLUSTERING_INFORMATION` function is `average_depth`. This output indicates the average number of micro-partitions that contain data for a given column value or combination of column values. The other outputs are not common to both functions. The `notes` output is only provided by the `SYSTEM$CLUSTERING_INFORMATION` function and it contains additional information or recommendations about the clustering status of the table. The `average_overlaps` output is only provided by the `SYSTEM$CLUSTERING_DEPTH` function and it indicates the average number of micro-partitions that overlap with other micro-partitions for a given column value or combination of column values. The `total_partition_count` output is only provided by the `SYSTEM$CLUSTERING_INFORMATION` function and it indicates the total number of micro-partitions in the table.

NEW QUESTION 49

A Data Engineer needs to know the details regarding the micro-partition layout for a table named `invoice` using a built-in function. Which query will provide this information?

- A. `SELECT SYSTEM$CLUSTERING_INFORMATION('Invoice');`
- B. `SELECT $CLUSTERING_INFORMATION('Invoice');`
- C. `CALL SYSTEM$CLUSTERING_INFORMATION('Invoice');`
- D. `CALL $CLUSTERING_INFORMATION('Invoice');`

Answer: A

Explanation:

The query that will provide information about the micro-partition layout for a table named `invoice` using a built-in function is `SELECT SYSTEM$CLUSTERING_INFORMATION('Invoice');`. The `SYSTEM$CLUSTERING_INFORMATION` function returns information about the clustering status of a table, such as the clustering key, the clustering depth, the clustering ratio, the partition count, etc. The function takes one argument: the table name in a qualified or unqualified form. In this case, the table name is `invoice` and it is unqualified, which means that it will use the current database and schema as the context. The other options are incorrect because they do not use a valid built-in function for providing information about the micro-partition layout for a table. Option B is incorrect because it uses `$CLUSTERING_INFORMATION` instead of `SYSTEM$CLUSTERING_INFORMATION`, which is not a valid function name. Option C is incorrect because it uses `CALL` instead of `SELECT`, which is not a valid way to invoke a table function. Option D is incorrect because it uses `CALL` instead of `SELECT` and `$CLUSTERING_INFORMATION` instead of `SYSTEM$CLUSTERING_INFORMATION`, which are both invalid.

NEW QUESTION 50

Which Snowflake feature facilitates access to external API services such as geocoders, data transformation, machine Learning models and other custom code?

- A. Security integration
- B. External tables
- C. External functions
- D. Java User-Defined Functions (UDFs)

Answer: C

Explanation:

External functions are Snowflake functions that facilitate access to external API services such as geocoders, data transformation, machine learning models and other custom code. External functions allow users to invoke external services from within SQL queries and pass arguments and receive results as JSON values. External functions require creating an API integration object and an external function object in Snowflake, as well as deploying an external service endpoint that can communicate with Snowflake via HTTPS.

NEW QUESTION 52

A Data Engineer is writing a Python script using the Snowflake Connector for Python. The Engineer will use the `snowflake.connector.connect` function to connect to Snowflake. The requirements are:

*Raise an exception if the specified database schema or warehouse does not exist

*improve download performance

Which parameters of the `connect` function should be used? (Select TWO).

- A. `authenticator`
- B. `arrow_number_to_decimal`
- C. `client_prefetch_threads`
- D. `client_session_keep_alive`
- E. `validate_default_parameters`

Answer: CE

Explanation:

The parameters of the `connect` function that should be used are `client_prefetch_threads` and `validate_default_parameters`. The `client_prefetch_threads` parameter controls the number of threads used to download query results from Snowflake. Increasing this parameter can improve download performance by parallelizing the download process. The `validate_default_parameters` parameter controls whether an exception should be raised if the specified database, schema, or warehouse does not exist or is not authorized. Setting this parameter to `True` can help catch errors early and avoid unexpected results.

NEW QUESTION 55

.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

DEA-C01 Practice Exam Features:

- * DEA-C01 Questions and Answers Updated Frequently
- * DEA-C01 Practice Questions Verified by Expert Senior Certified Staff
- * DEA-C01 Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- * DEA-C01 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year

100% Actual & Verified — Instant Download, Please Click
[Order The DEA-C01 Practice Test Here](#)