



Snowflake

Exam Questions DEA-C01

SnowPro Advanced: Data Engineer Certification Exam

About ExamBible

Your Partner of IT Exam

Found in 1998

ExamBible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, ExamBible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

NEW QUESTION 1

Within a Snowflake account permissions have been defined with custom roles and role hierarchies. To set up column-level masking using a role in the hierarchy of the current user, what command would be used?

- A. CORRECT_ROLE
- B. IKVOKER_ROLE
- C. IS_ROLE_IN_SESSION
- D. IS_GRANTED_TO_INVOKER_ROLE

Answer: C

Explanation:

The IS_ROLE_IN_SESSION function is used to set up column-level masking using a role in the hierarchy of the current user. Column-level masking is a feature in Snowflake that allows users to apply dynamic data masking policies to specific columns based on the roles of the users who access them. The IS_ROLE_IN_SESSION function takes a role name as an argument and returns true if the role is in the current user's session, or false otherwise. The function can be used in a masking policy expression to determine whether to mask or unmask a column value based on the role of the user. For example:

```
CREATE OR REPLACE MASKING POLICY email_mask AS (val string) RETURNS string -
> CASE WHEN IS_ROLE_IN_SESSION('HR') THEN val ELSE REGEXP_REPLACE(val, '(.)@.', '\1****\2') END;
```

In this example, the IS_ROLE_IN_SESSION function is used to create a masking policy for an email column. The masking policy returns the original email value if the user has the HR role in their session, or returns a masked email value with asterisks if not.

NEW QUESTION 2

What is a characteristic of the use of binding variables in JavaScript stored procedures in Snowflake?

- A. All types of JavaScript variables can be bound
- B. All Snowflake first-class objects can be bound
- C. Only JavaScript variables of type number, string and sf Date can be bound
- D. Users are restricted from binding JavaScript variables because they create SQL injection attack vulnerabilities

Answer: C

Explanation:

A characteristic of the use of binding variables in JavaScript stored procedures in Snowflake is that only JavaScript variables of type number, string and sf Date can be bound. Binding variables are a way to pass values from JavaScript variables to SQL statements within a stored procedure. Binding variables can improve the security and performance of the stored procedure by preventing SQL injection attacks and reducing the parsing overhead. However, not all types of JavaScript variables can be bound. Only the primitive types number and string, and the Snowflake-specific type sf Date, can be bound. The other options are incorrect because they do not describe a characteristic of the use of binding variables in JavaScript stored procedures in Snowflake. Option A is incorrect because authenticator is not a type of JavaScript variable, but a parameter of the snowflake.connector.connect function. Option B is incorrect because arrow_number_to_decimal is not a type of JavaScript variable, but a parameter of the snowflake.connector.connect function. Option D is incorrect because users are not restricted from binding JavaScript variables, but encouraged to do so.

NEW QUESTION 3

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_name ADD MASKING POLICY name_policy;
- C. ALTER TABLE customer MODIFY COLUMN first_name SET MASKING POLICY name_policy; last_name SET MASKING POLICY name_policy;
- 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 4

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 5

A stream called TRANSACTIONS_STM is created on top of a transactions table in a continuous pipeline running in Snowflake. After a couple of months, the TRANSACTIONS table is renamed transactiok3_raw to comply with new naming standards. What will happen to the TRANSACTIONS_STM object?

- A. TRANSACTIONS_STM will keep working as expected
- B. TRANSACTIONS_STM will be stale and will need to be re-created
- C. TRANSACTIONS_STM will be automatically renamed TRANSACTIONS_RAW_STM.
- D. Reading from the traksactioks_3T>: stream will succeed for some time after the expected STALE_TIME.

Answer: B

Explanation:

A stream is a Snowflake object that records the history of changes made to a table. A stream is associated with a specific table at the time of creation, and it cannot be altered to point to a different table later. Therefore, if the source table is renamed, the stream will become stale and will need to be re-created with the new table name. The other options are not correct because:

? TRANSACTIONS_STM will not keep working as expected, as it will lose track of the changes made to the renamed table.

? TRANSACTIONS_STM will not be automatically renamed TRANSACTIONS_RAW_STM, as streams do not inherit the name changes of their source tables.

? Reading from the transactions_stm stream will not succeed for some time after the expected STALE_TIME, as streams do not have a STALE_TIME property.

NEW QUESTION 6

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

```
select *
from
  sample_data.tpcds_sf10tcl.store_sales,
  order by ss_item_sk;
```



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 7

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 8

What is a characteristic of the operations of streams in Snowflake?

- A. Whenever a stream is queried, the offset is automatically advanced.
- B. When a stream is used to update a target table the offset is advanced to the current time.
- C. Querying a stream returns all change records and table rows from the current offset to the current time.
- D. Each committed and uncommitted transaction on the source table automatically puts a change record in the stream.

Answer: C

Explanation:

A stream is a Snowflake object that records the history of changes made to a table. A stream has an offset, which is a point in time that marks the beginning of the change records to be returned by the stream. Querying a stream returns all change records and table rows from the current offset to the current time. The offset is not automatically advanced by querying the stream, but it can be manually advanced by using the ALTER STREAM command. When a stream is used to update a target table, the offset is advanced to the current time only if the ON UPDATE clause is specified in the stream definition. Each committed transaction on the source table automatically puts a change record in the stream, but uncommitted transactions do not.

NEW QUESTION 9

The following code is executed in a Snowflake environment with the default settings:

```
able customer;  
  
transaction;  
  
table customer  
integer,  
varchar  
  
into customer values ('1', 'John');  
  
ck;  
  
$1 from customer;
```

What will be the result of the select statement?

- A. SQL compilation error object 'CUSTOMER' does not exist or is not authorized.
- B. John
- C. 1
- D. 1John

Answer: C

NEW QUESTION 10

What is the purpose of the BUILD_FILE_URL function in Snowflake?

- A. It generates an encrypted URL for accessing a file in a stage.
- B. It generates a staged URL for accessing a file in a stage.
- C. It generates a permanent URL for accessing files in a stage.
- D. It generates a temporary URL for accessing a file in a stage.

Answer: B

Explanation:

The BUILD_FILE_URL function in Snowflake generates a temporary URL for accessing a file in a stage. The function takes two arguments: the stage name and the file path. The generated URL is valid for 24 hours and can be used to download or view the file contents. The other options are incorrect because they do not describe the purpose of the BUILD_FILE_URL function.

NEW QUESTION 10

A Data Engineer is building a set of reporting tables to analyze consumer requests by region for each of the Data Exchange offerings annually, as well as click-through rates for each listing

Which views are needed MINIMALLY as data sources?

- A. SNOWFLAKE- DATA_SHARING_USAGE - LISTING_EVENTS_BAILY
- B. SNOWFLAKE.DATA_SHARING_USAGE.LISTING_CONSOKE>TION_DAILY
- C. SNOWFLAK
- D. DATA_SHARING_USAG
- E. LISTING_TELEMETRY_DAILY
- F. SNOWFLAKE.ACCOUNT_USAGE.DATA _TRANSFER_HISTORY

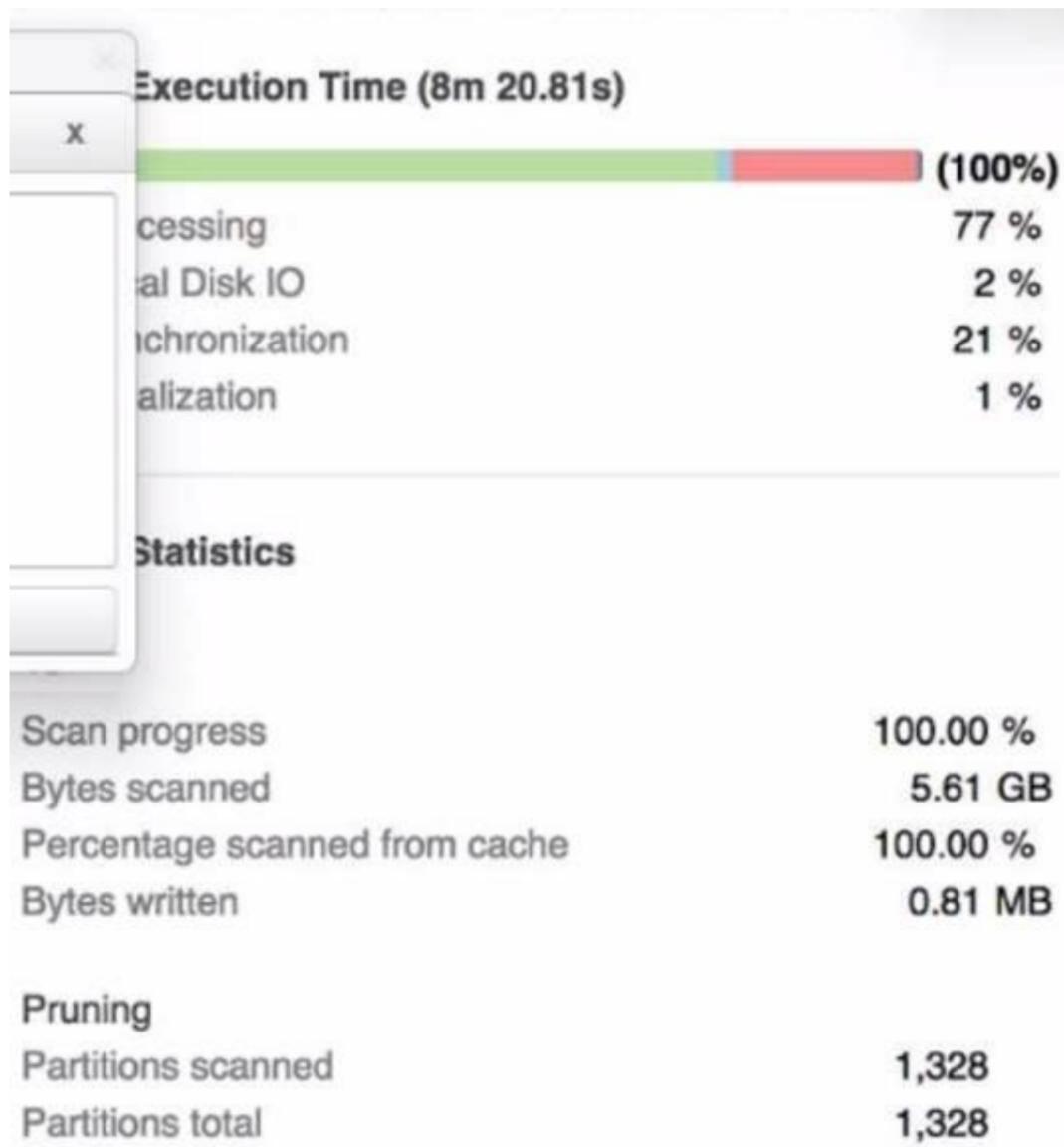
Answer: B

Explanation:

The SNOWFLAKE.DATA SHARING _USAGE.LISTING_CONSOKE>TION_DAILY view provides information about consumer requests by region for each of the Data Exchange offerings annually, as well as click- through rates for each listing. This view is the minimal data source needed for building the reporting tables. The other views are not relevant for this use case.

NEW QUESTION 15

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 20

A Data Engineer needs to ingest invoice data in PDF format into Snowflake so that the data can be queried and used in a forecasting solution. recommended way to ingest this data?

- A. Use Snowpipe to ingest the files that land in an external stage into a Snowflake table
- B. Use a COPY INTO command to ingest the PDF files in an external stage into a Snowflake table with a VARIANT column.
- C. Create an external table on the PDF files that are stored in a stage and parse the data into structured data
- D. Create a Java User-Defined Function (UDF) that leverages Java-based PDF parser libraries to parse PDF data into structured data

Answer: D

Explanation:

The recommended way to ingest invoice data in PDF format into Snowflake is to create a Java User-Defined Function (UDF) that leverages Java-based PDF parser libraries to parse PDF data into structured data. This option allows for more flexibility and control over how the PDF data is extracted and transformed. The other options are not suitable for ingesting PDF data into Snowflake. Option A and B are incorrect because Snowpipe and COPY INTO commands can only ingest files that are in supported file formats, such as CSV, JSON, XML, etc. PDF files are not supported by Snowflake and will cause errors or unexpected results. Option C is incorrect because external tables can only query files that are in supported file formats as well. PDF files cannot be parsed by external tables and will cause errors or unexpected results.

NEW QUESTION 25

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 27

Which functions will compute a 'fingerprint' over an entire table, query result, or window to quickly detect changes to table contents or query results? (Select TWO).

- A. `HASH(*)`
- B. `HASH_AGG(*)`
- C. `HASH_AGG(<expr>, <expr>)`
- D. `HASH_AGG_COMPARE(*)`
- E. `HASH_COMPARE(*)`

Answer: BC

Explanation:

The functions that will compute a 'fingerprint' over an entire table, query result, or window to quickly detect changes to table contents or query results are:

? `HASH_AGG(*)`: This function computes a hash value over all columns and rows in a table, query result, or window. The function returns a single value for each group defined by a `GROUP BY` clause, or a single value for the entire input if no `GROUP BY` clause is specified.

? `HASH_AGG(<expr>, <expr>)`: This function computes a hash value over two expressions in a table, query result, or window. The function returns a single value for each group defined by a `GROUP BY` clause, or a single value for the entire input if no `GROUP BY` clause is specified. The other functions are not correct because:

? `HASH(*)`: This function computes a hash value over all columns in a single row. The function returns one value per row, not one value per table, query result, or window.

? `HASH_AGG_COMPARE()`: This function compares two hash values computed by `HASH_AGG()` over two tables or query results and returns true if they are equal or false if they are different. The function does not compute a hash value itself, but rather compares two existing hash values.

? `HASH_COMPARE()`: This function compares two hash values computed by `HASH()` over two rows and returns true if they are equal or false if they are different. The function does not compute a hash value itself, but rather compares two existing hash values.

NEW QUESTION 32

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 36

At what isolation level are Snowflake streams?

- A. Snapshot
- B. Repeatable read
- C. Read committed
- D. Read uncommitted

Answer: B

Explanation:

The isolation level of Snowflake streams is repeatable read, which means that each transaction sees a consistent snapshot of data that does not change during its execution. Streams use time travel internally to provide this isolation level and ensure that queries on streams return consistent results regardless of concurrent transactions on their source tables.

NEW QUESTION 40

A Data Engineer wants to create a new development database (DEV) as a clone of the permanent production database (PROD) There is a requirement to disable Fail-safe for all tables.

Which command will meet these requirements?

- A. `CREATE DATABASE DEV CLONE PROD FAIL_SAFE=FALSE;`
- B. `CREATE DATABASE DEV CLONE PROD;`
- C. `CREATE TRANSIENT DATABASE DEV CLONE RPOD`
- D. `CREATE DATABASE DEV CLOSE PRODDATA_RETENTION_TIME_IN_DAYS =0L`

Answer: C

Explanation:

This option will meet the requirements of creating a new development database (DEV) as a clone of the permanent production database (PROD) and disabling Fail-safe for all tables. By using the CREATE TRANSIENT DATABASE command, the Data Engineer can create a transient database that does not have Fail-safe enabled by default. Fail-safe is a feature in Snowflake that provides additional protection against data loss by retaining historical data for seven days beyond the time travel retention period. Transient databases do not have Fail-safe enabled, which means that they do not incur additional storage costs for historical data beyond their time travel retention period. By using the CLONE option, the Data Engineer can create an exact copy of the PROD database, including its schemas, tables, views, and other objects.

NEW QUESTION 43

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 47

Which Snowflake objects does the Snowflake Kafka connector use? (Select THREE).

- A. Pipe
- B. Serverless task
- C. Internal user stage
- D. Internal table stage
- E. Internal named stage
- F. Storage integration

Answer: ADE

Explanation:

The Snowflake Kafka connector uses three Snowflake objects: pipe, internal table stage, and internal named stage. The pipe object is used to load data from an external stage into a Snowflake table using COPY statements. The internal table stage is used to store files that are loaded from Kafka topics into Snowflake using PUT commands. The internal named stage is used to store files that are rejected by the COPY statements due to errors or invalid data. The other options are not objects that are used by the Snowflake Kafka connector. Option B, serverless task, is an object that can execute SQL statements on a schedule without requiring a warehouse. Option C, internal user stage, is an object that can store files for a specific user in Snowflake using PUT commands. Option F, storage integration, is an object that can enable secure access to external cloud storage services without exposing credentials.

NEW QUESTION 52

.....

Relate Links

100% Pass Your DEA-C01 Exam with ExamBible Prep Materials

<https://www.exambible.com/DEA-C01-exam/>

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>