



Snowflake

Exam Questions DEA-C01

SnowPro Advanced: Data Engineer Certification Exam

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. INVOKER_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

A Data Engineer is building a pipeline to transform a 1 TB table by joining it with supplemental tables. The Engineer is applying filters and several aggregations leveraging Common Table Expressions (CTEs) using a size Medium virtual warehouse in a single query in Snowflake.

After checking the Query Profile, what is the recommended approach to MAXIMIZE performance of this query if the Profile shows data spillage?

- A. Enable clustering on the table
- B. Increase the warehouse size
- C. Rewrite the query to remove the CTEs.
- D. Switch to a multi-cluster virtual warehouse

Answer: B

Explanation:

The recommended approach to maximize performance of this query if the Profile shows data spillage is to increase the warehouse size. Data spillage occurs when the query requires more memory than the warehouse can provide and has to spill some intermediate results to disk. This can degrade the query performance by increasing the disk IO time. Increasing the warehouse size can increase the amount of memory available for the query and reduce or eliminate data spillage.

NEW QUESTION 3

A company is using Snowpipe to bring in millions of rows every day of Change Data Capture (CDC) into a Snowflake staging table on a real-time basis. The CDC needs to get processed and combined with other data in Snowflake and land in a final table as part of the full data pipeline.

How can a Data engineer MOST efficiently process the incoming CDC on an ongoing basis?

- A. Create a stream on the staging table and schedule a task that transforms data from the stream only when the stream has data.
- B. Transform the data during the data load with Snowpipe by modifying the related copy into statement to include transformation steps such as case statements and JOIN'S.
- C. Schedule a task that dynamically retrieves the last time the task was run from information_schema-task_history and use that timestamp to process the delta of the new rows since the last time the task was run.
- D. Use a create or replace table as statement that references the staging table and includes all the transformation SQL.
- E. Use a task to run the full create or replace table as statement on a scheduled basis.

Answer: A

Explanation:

The most efficient way to process the incoming CDC on an ongoing basis is to create a stream on the staging table and schedule a task that transforms data from the stream only when the stream has data. A stream is a Snowflake object that records changes made to a table, such as inserts, updates, or deletes. A stream can be queried like a table and can provide information about what rows have changed since the last time the stream was consumed. A task is a Snowflake object that can execute SQL statements on a schedule without requiring a warehouse. A task can be configured to run only when certain conditions are met, such as when a stream has data or when another task has completed successfully. By creating a stream on the staging table and scheduling a task that transforms data from the stream, the Data Engineer can ensure that only new or modified rows are processed and that no unnecessary computations are performed.

NEW QUESTION 4

Database XYZ has the data_retention_time_in_days parameter set to 7 days and table xyz.public.ABC has the data_retention_time_in_days set to 10 days.

A Developer accidentally dropped the database containing this single table 8 days ago and just discovered the mistake.

How can the table be recovered?

- A. undrop database xyz;
- B. create table abc_restore as select * from xyz.public.abc at ({offset => -60*60*24*8});
- C. create table abc_restore clone xyz.public.abc at (offset => -3*60*60*24*3);
- D. Create a Snowflake Support case to restore the database and table from a backup.

Answer: A

Explanation:

The table can be recovered by using the undrop database xyz; command. This command will restore the database that was dropped within the last 14 days, along with all its schemas and tables, including the customer table. The data_retention_time_in_days parameter does not affect this command, as it only applies to time

travel queries that reference historical data versions of tables or databases. The other options are not valid ways to recover the table. Option B is incorrect because creating a table as `select * from xyz.public.ABC at {offset => -6060248}` will not work, as this query will try to access a historical data version of the ABC table that does not exist anymore after dropping the database. Option C is incorrect because creating a table clone `xyz.public.ABC at {offset => -360024*3}` will not work, as this query will try to clone a historical data version of the ABC table that does not exist anymore after dropping the database. Option D is incorrect because creating a Snowflake Support case to restore the database and table from fail-safe will not work, as fail-safe is only available for disaster recovery scenarios and cannot be accessed by customers.

NEW QUESTION 5

A Data Engineer has created table t1 with datatype VARIANT: create or replace table t1 (cl variant);

The Engineer has loaded the following JSON data set. which has information about 4 laptop models into the table:

```
{
  "device_model": [
    {
      "manufacturer": "HP",
      "model": "HP 240 G8",
      "model_id": "hp 240 g8",
      "model_name": "240 G8"
    },
    {
      "manufacturer": "HP",
      "model": "HP EliteBook 1030 G1",
      "model_id": "hp elitebook 1030 g1",
      "model_name": "EliteBook 1030 G1"
    },
    {
      "manufacturer": "HP",
      "model": "HP ZBook 15 G2",
      "model_id": "hp zbook 15 g2",
      "model_name": "ZBook 15 G2"
    },
    {
      "manufacturer": "Lenovo",
      "model": "Lenovo B50-70",
      "model_id": "lenovo b50-70",
      "model_name": "B50-70"
    }
  ]
}
```

The Engineer now wants to query that data set so that results are shown as normal structured data. The result should be 4 rows and 4 columns without the double quotes surrounding the data elements in the JSON data.

The result should be similar to the use case where the data was selected from a normal

relational table z2 where t2 has string data type columns model id, model, manufacturer, and =iccisi_r.an=. and is queried with the SQL clause `select * from t2;`

Which select command will produce the correct results?

A)

```
select value:model_id::string
, value:model::string
, value:manufacturer::string
, value:model_name::string
from t1
, lateral flatten(input => c1);
```

B)

```
select value:model_id::string
, value:model::string
, value:manufacturer::string
, value:model_name::string
from t1
, lateral flatten(input => c1:device_model);
```

C)

```
select model_id::string
, model::string
, manufacturer::string
, model_name::string
from t1
, lateral flatten(input => c1:device_model);
```

D)

```
select value:model_id
, value:model
, value:manufacturer
, value:model_name
from t1
, lateral flatten(input => c1:device_model);
```

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

Answer: B**NEW QUESTION 6**

A new customer table is created by a data pipeline in a Snowflake schema where MANAGED ACCESS enabled.
.... Can gran access to the CUSTOMER table? (Select THREE.)

- A. The role that owns the schema
- B. The role that owns the database
- C. The role that owns the customer table
- D. The SYSADMIN role
- E. The SECURITYADMIN role
- F. The USERADMIN role with the manage grants privilege

Answer: ABE**Explanation:**

The roles that can grant access to the CUSTOMER table are the role that owns the schema, the role that owns the database, and the SECURITYADMIN role. These roles have the ownership or the manage grants privilege on the schema or the database level, which allows them to grant access to any object within them. The other options are incorrect because they do not have the necessary privilege to grant access to the CUSTOMER table. Option C is incorrect because the role that owns the customer table cannot grant access to itself or to other roles. Option D is incorrect because the SYSADMIN role does not have the manage grants privilege by default and cannot grant access to objects that it does not own. Option F is incorrect because the USERADMIN role with the manage grants privilege can only grant access to users and roles, not to tables.

NEW QUESTION 7

A Data Engineer ran a stored procedure containing various transactions During the execution, the session abruptly disconnected preventing one transaction from 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_TIMEOUT to 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 8

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 still running'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 9

A Data Engineer is working on a continuous data pipeline which receives data from Amazon Kinesis Firehose and loads the data into a staging table which will later be used in the data transformation process. The average file size is 300-500 MB.

The Engineer needs to ensure that Snowpipe is performant while minimizing costs. How can this be achieved?

- A. Increase the size of the virtual warehouse used by Snowpipe.
- B. Split the files before loading them and set the SIZE_LIMIT option to 250 MB.
- C. Change the file compression size and increase the frequency of the Snowpipe loads.
- D. Decrease the buffer size to trigger delivery of files sized between 100 to 250 MB in Kinesis Firehose.

Answer: B

Explanation:

This option is the best way to ensure that Snowpipe is performant while minimizing costs. By splitting the files before loading them, the Data Engineer can reduce the size of each file and increase the parallelism of loading. By setting the SIZE_LIMIT option to 250 MB, the Data Engineer can specify the maximum file size that can be loaded by Snowpipe, which can prevent performance degradation or errors due to large files. The other options are not optimal because:

? Increasing the size of the virtual warehouse used by Snowpipe will increase the performance but also increase the costs, as larger warehouses consume more credits per hour.

? Changing the file compression size and increasing the frequency of the Snowpipe loads will not have much impact on performance or costs, as Snowpipe already supports various compression formats and automatically loads files as soon as they are detected in the stage.

? Decreasing the buffer size to trigger delivery of files sized between 100 to 250 MB

in Kinesis Firehose will not affect Snowpipe performance or costs, as Snowpipe does not depend on Kinesis Firehose buffer size but rather on its own SIZE_LIMIT option.

NEW QUESTION 10

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 10

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 14

The JSON below is stored in a variant column named v in a table named jCustRaw:

```
{
  "id": "6282638561cf48544e2ef7e9",
  "company": "FLYBOYZ",
  "isActive": true,
  "name": "Dean Head",
  "teamMembers": [
    {
      "age": 29,
      "eyeColor": "green",
      "name": "Dominique Grimes",
      "registered": "2017-02-19T06:12:36 +06:00"
    },
    {
      "age": 39,
      "eyeColor": "green",
      "name": "Pearl Dunlap",
      "registered": "2018-05-12T09:21:42 +05:00"
    },
    {
      "age": 22,
      "eyeColor": "blue",
      "name": "Cardenas Warren",
      "registered": "2019-04-08T01:24:29 +05:00"
    }
  ]
}
```

Which query will return one row per team member (stored in the teamMembers array) along all of the attributes of each team member?

A)

```
select
  t2.name AS memberName
  ,t2.registered AS registeredDttm
  ,t2.age AS age
  ,t2.eyeColor AS eyeColor
from jCustRaw t1
  lateral flatten(v) t2
select
  Name
  ,t2.value:name::varchar AS memberName
  ,t2.value:registered::timestamp AS registeredDttm
  ,t2.value:age::number AS age
  ,t2.value:eyeColor::varchar AS eyeColor
from jCustRaw t1
  lateral flatten(input)
```

C)

```
select
  v:teamMembers.name::varchar AS memberName
  ,v:teamMembers.registered::timestamp AS registeredDttm
  ,v:teamMembers.age::number AS age
  ,v:teamMembers.eyeColor::varchar AS eyeColor
from jCustRaw;
```

D)

```
select
  v:teamMembers[0].name::varchar AS memberName
  ,v:teamMembers[0].registered::timestamp AS registeredDttm
  ,v:teamMembers[0].age::number AS age
  ,v:teamMembers[0].eyeColor::varchar AS eyeColor
from jCustRaw;
```

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

Answer: B

NEW QUESTION 19

What are characteristics of Snowpark Python packages? (Select THREE).

Third-party packages can be registered as a dependency to the Snowpark session using the session.import() method.

- A. Python packages can access any external endpoints
- B. Python packages can only be loaded in a local environment
- C. Third-party supported Python packages are locked down to prevent hitting
- D. The SQL command DESCRIBE FUNCTION will list the imported Python packages of the Python User-Defined Function (UDF).
- E. Querying information schema .packages will provide a list of supported Python packages and versions

Answer: ADE

Explanation:

The characteristics of Snowpark Python packages are:

? Third-party packages can be registered as a dependency to the Snowpark session using the session.import() method.

? The SQL command DESCRIBE FUNCTION will list the imported Python packages of the Python User-Defined Function (UDF).

? Querying information_schema.packages will provide a list of supported Python packages and versions.

These characteristics indicate how Snowpark Python packages can be imported, inspected, and verified in Snowflake. The other options are not characteristics of Snowpark Python packages. Option B is incorrect because Python packages can be loaded in both local and remote environments using Snowpark. Option C is incorrect because third-party supported Python packages are not locked down to prevent hitting external endpoints, but rather restricted by network policies and security settings.

NEW QUESTION 24

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 27

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 30

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' i:
- 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 31

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 32

Which system role is recommended for a custom role hierarchy to be ultimately assigned to?

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

Answer: B

Explanation:

The system role that is recommended for a custom role hierarchy to be ultimately assigned to is SECURITYADMIN. This role has the manage grants privilege on all objects in an account, which allows it to grant access privileges to other roles or revoke them as needed. This role can also create or modify custom roles and assign them to users or other roles. By assigning custom roles to SECURITYADMIN, the role hierarchy can be managed centrally and securely. The other options are not recommended system roles for a custom role hierarchy to be ultimately assigned to. Option A is incorrect because ACCOUNTADMIN is the most powerful role in an account, which has full access to all objects and operations. Assigning custom roles to ACCOUNTADMIN can pose a security risk and should be avoided. Option C is incorrect because SYSTEMADMIN is a role that has full access to all objects in the public schema of the account, but not to other schemas or databases. Assigning custom roles to SYSTEMADMIN can limit the scope and flexibility of the role hierarchy. Option D is incorrect because USERADMIN is a role that can manage users and roles in an account, but not grant access privileges to other objects. Assigning custom roles to USERADMIN can prevent the role hierarchy from controlling access to data and resources.

NEW QUESTION 36

When would a Data engineer use table with the flatten function instead of the lateral flatten combination?

- A. When TABLE with FLATTEN requires another source in the from clause to refer to
- B. When TABLE with FLATTEN requires no additional source in the from clause to refer to
- C. When the LATERAL FLATTEN combination requires no other source in the from clause to refer to
- D. When table with FLATTEN is acting like a sub-query executed for each returned row

Answer: A

Explanation:

The TABLE function with the FLATTEN function is used to flatten semi-structured data, such as JSON or XML, into a relational format. The TABLE function returns a table expression that can be used in the FROM clause of a query. The TABLE function with the FLATTEN function requires another source in the FROM clause to refer to, such as a table, view, or subquery that contains the semi-structured data. For example: SELECT t.value:city::string AS city, f.value AS population FROM cities t, TABLE(FLATTEN(input => t.value:population)) f; In this example, the TABLE function with the FLATTEN function refers to the cities table in the FROM clause, which contains JSON data in a variant column named value. The FLATTEN function flattens the population array within each JSON object and returns a table expression with two columns: key and value. The query then selects the city and population values from the table expression.

NEW QUESTION 40

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 42

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 46

A company has an extensive script in Scala that transforms data by leveraging DataFrames. A Data engineer needs to move these transformations to Snowpark. ...characteristics of data transformations in Snowpark should be considered to meet this requirement? (Select TWO)

- A. It is possible to join multiple tables using DataFrames.
- B. Snowpark operations are executed lazily on the server.
- C. User-Defined Functions (UDFs) are not pushed down to Snowflake
- D. Snowpark requires a separate cluster outside of Snowflake for computations
- E. Columns in different DataFrames with the same name should be referred to with squared brackets

Answer: AB

Explanation:

The characteristics of data transformations in Snowpark that should be considered to meet this requirement are:

? It is possible to join multiple tables using DataFrames.

? Snowpark operations are executed lazily on the server.

These characteristics indicate how Snowpark can perform data transformations using DataFrames, which are similar to the ones used in Scala. DataFrames are distributed collections of rows that can be manipulated using various operations, such as joins, filters, aggregations, etc. DataFrames can be created from different sources, such as tables, files, or SQL queries. Snowpark operations are executed lazily on the server, which means that they are not performed until an action is triggered, such as a write or a collect operation. This allows Snowpark to optimize the execution plan and reduce the amount of data transferred between the client and the server.

The other options are not characteristics of data transformations in Snowpark that should be considered to meet this requirement. Option C is incorrect because User-Defined Functions (UDFs) are pushed down to Snowflake and executed on the server. Option D is incorrect because Snowpark does not require a separate cluster outside of Snowflake for computations, but rather uses virtual warehouses within Snowflake. Option E is incorrect because columns in different DataFrames with the same name should be referred to with dot notation, not squared brackets.

NEW QUESTION 47

.....

Thank You for Trying Our Product

We offer two products:

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

2nd - Questons 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](#)