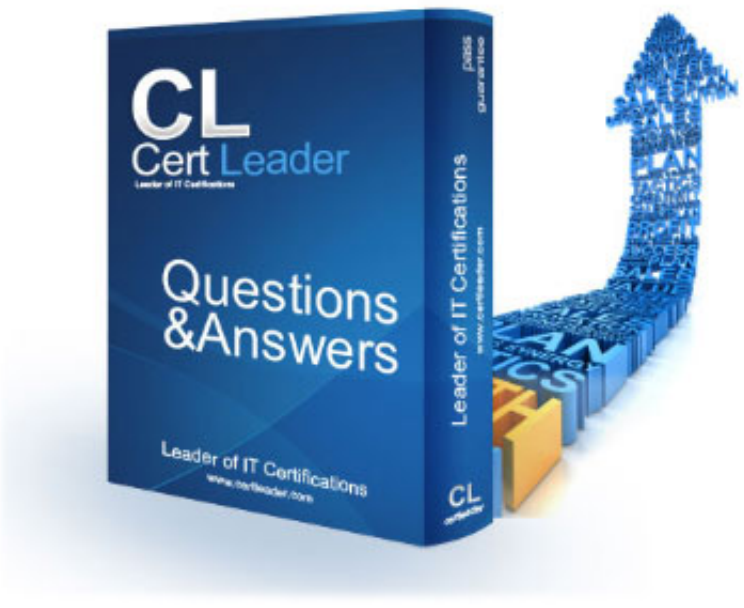
# DEA-C01 Dumps

# SnowPro Advanced: Data Engineer Certification Exam

## https://www.certleader.com/DEA-C01-dumps.html



**NEW QUESTION 1**
A Data Engineer needs to load JSON output from some software into Snowflake using Snowpipe.
Which recommendations apply to this scenario? (Select THREE)

A. Load large files (1 GB or larger)
B. Ensure that data files are 100-250 MB (or larger) in size compressed
C. Load a single huge array containing multiple records into a single table row
D. Verify each value of each unique element stores a single native data type (string or number)
E. Extract semi-structured data elements containing null values into relational columns before loading
F. Create data files that are less than 100 MB and stage them in cloud storage at a sequence greater than once each minute

**Answer:** BDF

**Explanation:**
The recommendations that apply to this scenario are:
? Ensure that data files are 100-250 MB (or larger) in size compressed: This recommendation will improve Snowpipe performance by reducing the number of files that need to be loaded and increasing the parallelism of loading. Smallerfiles can cause performance degradation or errors due to excessive metadata operations or network latency.
? Verify each value of each unique element stores a single native data type (string or number): This recommendation will improve Snowpipe performance by avoiding data type conversions or errors when loading JSON data into variant columns. Snowflake supports two native data types for JSON elements: string and number. If an element has mixed data types across different files or records, such as string and boolean, Snowflake will either convert them to string or raise an error, depending on the FILE_FORMAT option.
? Create data files that are less than 100 MB and stage them in cloud storage at a sequence greater than once each minute: This recommendation will minimize Snowpipe costs by reducing the number of notifications that need to be sent to Snowpipe for auto-ingestion. Snowpipe charges for notifications based on the number of files per notification and the frequency of notifications. By creating smaller files and staging them at a lower frequency, fewer notifications will be needed.

**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 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 processedand 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 andJOIN'S.
C. Schedule a task that dynamically retrieves the last time the task was run from information_schema-rask_hiSwOry and use that timestamp to process the delta of the new rows since the last time the task was run.
D. Use a create ok replace table as statement that references the staging table and includes all the transformation SQ
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**
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 5**
The following is returned fromSYSTEMCLUSTERING_INFORMATION () for a tablenamed orders with adate column named O_ORDERDATE:


```
{
  "cluster_by_keys" : "LINEAR(YEAR(O_ORDERDATE))",
  "total_partition_count" : 536,
  "total_constant_partition_count" : 493,
  "average_overlaps" : 0.1716,
  "average_depth" : 1.0914,
  "partition_depth_histogram" : {
    "00000" : 0,
    "00001" : 491,
    "00002" : 41,
    "00003" : 4,
    "00004" : 0,
    "00005" : 0,
    "00006" : 0,
    "00007" : 0,
    "00008" : 0,
    "00009" : 0,
    "00010" : 0,
    "00011" : 0,
    "00012" : 0,
    "00013" : 0,
    "00014" : 0,
    "00015" : 0,
    "00016" : 0
  }
}
```


What does the total_constant_partition_count value indicate about this table?

A. The table is clustered very well on_ORDERDATE, as there are 493 micro-partitions that could not be significantly improved by reclustering
B. The table is not clustered well on O_ORDERDATE, as there are 493 micro-partitions where the range of values in that column overlap with every other micro partition in the table.
C. The data inO_ORDERDATEdoes not change very often as there are 493 micro- partitionscontaining rows where that column has not been modified since the row was created
D. The data inO_ORDERDATEhas a very low cardinality as there are 493 micro-partitions where there is only a single distinct value in that column for all rows in the micro-partition

**Answer:** B

**Explanation:**
The total_constant_partition_count value indicates the number of micro- partitions where the clustering key column has a constant value across all rows in the micro-partition. However, this does not necessarily mean that the table is clustered well on that column, as there could be other micro-partitions where the range of values in that column overlap with each other. This is the case for the orders table, as the clustering depth is 1, which means that every micro-partition overlaps with every other micro-partition on O_ORDERDATE. This indicates that the table is not clustered well on O_ORDERDATE and could benefit from reclustering.

**NEW QUESTION 6**
A Data Engineer would like to define a file structure for loading and unloading data Where can the file structure be defined? (Select THREE)

A. copy command
B. MERGE command
C. FILE FORMAT Object
D. pipe object
E. stage object
F. INSERT command

**Answer:** ACE

**Explanation:**
The places where the file format can be defined are copy command, file format object, and stage object. These places allow specifying or referencing a file format that defines how data files are parsed and loaded into or unloaded from Snowflake tables. A file format can include various options, such as field delimiter, field enclosure, compression type, date format, etc. The other options are not places where the file format can be defined. Option B is incorrect because MERGE command is a SQL command that can merge data from one table into another based on a join condition, but it does not involve loading or unloading data files. Option D is incorrect because pipe object is a Snowflake object that can load data from an external stage into a Snowflake table using COPY statements, but it does not define or reference a file format. Option F is incorrect because INSERT command is a SQL command that can insert data into a Snowflake table from literal values or subqueries, but it does not involve loading or unloading data files.

**NEW QUESTION 7**
A Data Engineer enables a result cache at the session level with the following command: ALTER SESSION SET USE CACHED RESULT = TRUE;
The Engineer then runs the following select query twice without delay:

```
SELECT *
FROM SNOWFLAKE_SAMPLE_DATA.TPCH_SF1.CUSTOMER
SAMPLE(10) SEED (99);
```

The underlying table does not change between executions What are the results of both runs?

A. The first and second run returned the same results because sample is deterministic
B. The first and second run returned the same results, because the specific SEEDvalue was provided.
C. The first and second run returned different results because the query is evaluated each time it is run.
D. The first and second run returned differentresults because the query uses *instead of an explicit column list

**Answer:** B

**Explanation:**
The result cache is enabled at the session level, which means that repeated queries will return cached results if there is no change in the underlying data or session parameters. However, in this case, the result cache is not relevant because the query uses a specific SEED value for sampling, which makes it deterministic. Therefore, both runs will return the same results regardless of caching.

**NEW QUESTION 8**
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 _STMwill keep working as expected
B. TRANSACTIONS _STMwill be stale and will need to be re-created
C. TRANSACTIONS _STMwill be automatically renamedTRANSACTIONS _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 9**
Given the table sales which has a clustering key of column CLOSED_DATE which table function will return the average clustering depth for the SALES_REPRESENTATIVEcolumn for the North American region?
A)

```
select system$clustering_information('Sales', 'sales_representative', 'region = ''North America''');
```

```
select system$clustering_depth('Sales', 'sales_representative', 'region = ''North America''');
```

C)

```
select system$clustering_depth('Sales', 'sales_representative') where region = 'North America';
```

D)

```
select system$clustering_information('Sales', 'sales_representative') where region = 'North America';
```

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

**Answer:** B

**Explanation:**

The table function SYSTEM$CLUSTERING_DEPTH returns the average clustering depth for a specified column or set of columns in a table. The function takes two arguments: the table name and the column name(s). In this case, the table name is sales and the column name is SALES_REPRESENTATIVE. The function also supports a WHERE clause to filter the rows for which the clustering depth is calculated. In this case, the WHERE clause is REGION = 'North America'. Therefore, the function call in Option B will return the desired result.

**NEW QUESTION 10**
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 10**
Assuming a Data Engineer has all appropriate privileges and context which statements would be used to assess whether the User-Defined Function (UDF), MTBATA3ASZ. SALES .REVENUE_BY_REGION, exists and is secure? (Select TWO)

A. SHOW DS2R FUNCTIONS LIKE 'REVEX'^BYJIESION' IN SCHEMA SALES;
B. SELECT IS_SECURE FROM SNOWFLAK
C. INFCRXATION_SCKZM
D. FUNCTIONS WHERE FUNCTI0N_3CHEMA = 'SALES' AND FUNCTI CN_NAXE =•ftEVEXUE_BY_RKXQH4;
E. SELECT IS_SEC"JRE FROM INFOR>LVTICN_SCHEM
F. FUNCTIONS WHERE FUNCTION_SCHEMA = 'SALES1 AND FUNGTZON_NAME = ' REVENUE_BY_REGION';
G. SHOW EXTERNAL FUNCTIONS LIKE 'REVENUE_BY_REGION'IB SCHEMA SALES;
H. SHOW SECURE FUNCTIONS LIKE 'REVENUE 3Y REGION' IN SCHEMA SALES;

**Answer:** AB

**Explanation:**
The statements that would be used to assess whether the UDF, MTBATA3ASZ. SALES .REVENUE_BY_REGION, exists and is secure are:
? SHOW DS2R FUNCTIONS LIKE 'REVEX'^BYJIESION' IN SCHEMA SALES;:
This statement will show information about the UDF, including its name, schema, database, arguments, return type, language, and security option. If the UDF does not exist, the statement will return an empty result set.
? SELECT IS_SECURE FROM SNOWFLAKE. INFCRXATION_SCKZMA.
FUNCTIONS WHERE FUNCTI0N_3CHEMA = 'SALES' AND FUNCTI CN_NAXE
= •ftEVEXUE_BY_RKXQH4;: This statement will query the SNOWFLAKE.INFORMATION_SCHEMA.FUNCTIONS view, which contains metadata about the UDFs in the current database. The statement will return the IS_SECURE column, which indicates whether the UDF is secure or not. If the UDF does not exist, the statement will return an empty result set. The other statements are not correct because:
? SELECT IS_SEC"JRE FROM INFOR>LVTICN_SCHEMA. FUNCTIONS WHERE
FUNCTION_SCHEMA = 'SALES1 AND FUNGTZON_NAME = '
REVENUE_BY_REGION';: This statement will query the INFORMATION_SCHEMA.FUNCTIONS view, which contains metadata about the UDFs in the current schema. However, the statement has a typo in the schema name ('SALES1' instead of 'SALES'), which will cause it to fail or return incorrect results.
? SHOW EXTERNAL FUNCTIONS LIKE 'REVENUE_BY_REGION' IB SCHEMA
SALES;: This statement will show information about external functions, not UDFs. External functions are Snowflake functions that invoke external services via HTTPS requests and responses. The statement will not return any results for the UDF.
? SHOW SECURE FUNCTIONS LIKE 'REVENUE 3Y REGION' IN SCHEMA
SALES;: This statement is invalid because there is no such thing as secure functions in Snowflake. Secure functions are a feature of some other databases, such as PostgreSQL, but not Snowflake. The statement will cause a syntax error.

**NEW QUESTION 12**
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 offeringsannually, 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 17**
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 18**
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 19**
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 querywill 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 I
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 24**
Which output is provided by both theSYSTEM$CLUSTERING_DEPTHfunction and theSYSTEM$CLUSTERING_INFORMATIONfunction?

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 combinationof 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 26**
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 28**
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 31**
A Data Engineer executes a complex query and wants to make use of Snowflake s query results caching capabilities to reuse the results.
Which conditions must be met? (Select THREE).

A. The results must be reused within 72 hours.
B. The query must be executed using the same virtual warehouse.
C. The USED_CACHED_RESULT parameter must be included in the query.
D. The table structure contributing to the query result cannot have changed
E. The new query must have the same syntax as the previously executed query.
F. The micro-partitions cannot have changed due to changes to other data in the table

**Answer:** ADE

**Explanation:**
Snowflake’s query results caching capabilities allow users to reuse the results of previously executed queries without re-executing them. For this to happen, the following conditions must be met:
? The results must be reused within 24 hours (not 72 hours), which is the default
time-to-live (TTL) for cached results.

? The query must be executed using any virtual warehouse (not necessarily the same one), as long as it is in the same region and account as the original query.
? The USED_CACHED_RESULT parameter does not need to be included in the query, as it is enabled by default at the account level. However, it can be disabled or overridden at the session or statement level.
? The table structure contributing to the query result cannot have changed, such as adding or dropping columns, changing data types, or altering constraints.
? The new query must have the same syntax as the previously executed query, including whitespace and case sensitivity.
? The micro-partitions cannot have changed due to changes to other data in the table, such as inserting, updating, deleting, or merging rows.

**NEW QUESTION 36**
.....

# Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

## 100% Pass Your DEA-C01 Exam with Our Prep Materials Via below:

https://www.certleader.com/DEA-C01-dumps.html