ITTEST

QUESTION & ANSWER

Guías de estudio precisos, Alta tasa de paso!

Ittest ofrece información actualizada de forma gratuita en un año!

Exam : **DEA-C01**

Title : SnowPro Advanced Data Engineer Certification

Version : DEMO

1.Streams cannot be created to query change data on which of the following objects? [Select All that Apply]

A. Standard tables, including shared tables.

B. Views, including secure views

C. Directory tables

D. Query Log Tables

E. External tables

Answer: B,D,E

Explanation:

Streams capture DML changes (INSERT, UPDATE, DELETE) made to the underlying table. You can't create streams on objects that do not directly store data or do not support DML operations, such as views and external tables.

Query log tables in Snowflake are system-generated views that give you insights into the queries executed. You cannot create streams on these tables because they are system-managed and don't support direct DML operations.

2.Tasks may optionally use table streams to provide a convenient way to continuously process new or changed data. A task can transform new or changed rows that a stream surfaces. Each time a task is scheduled to run, it can verify whether a stream contains change data for a table and either consume the change data or skip the current run if no change data exists.

Which System Function can be used by Data engineer to verify whether a stream contains changed data for a table?

A. SYSTEM$STREAM_HAS_CHANGE_DATA

B. SYSTEM$STREAM_CDC_DATA

C. SYSTEM$STREAM_HAS_DATA

D. SYSTEM$STREAM_DELTA_DATA

Answer: A

Explanation:

The SYSTEM$STREAM_HAS_CHANGE_DATA function is used to check if there is any change data available in a specified stream. If the stream has data, the function will return TRUE; otherwise, it will return FALSE. This is useful when combined with tasks to decide whether or not to execute a transformation or processing based on the presence of new change data in the stream.

3.The Above example indicates that the SF_DATA table is not well-clustered for which of following valid reasons?

A. Zero (0) constant micro-partitions out of 1156 total micro-partitions.

B. High average of overlapping micro-partitions.

C. High average of overlap depth across micro-partitions.

D. Most of the micro-partitions are grouped at the lower-end of the histogram, with the majority of micro-partitions having an overlap depth between 64 and 128.

E. ALL of the above

Answer: E

4.Mark a Data Engineer, looking to implement streams on local views & want to use change tracking

metadata for one of its Data Loading use case. Please select the incorrect understanding points of Mark with respect to usage of Streams on Views?

A. For streams on views, change tracking must be enabled explicitly for the view and underlying tables to add the hidden columns to these tables.

B. The CDC records returned when querying a stream rely on a combination of the offset stored in the stream and the change tracking metadata stored in the table.

C. Views with GROUP BY & LIMIT Clause are supported by Snowflake.

D. As an alternative to streams, Snowflake supports querying change tracking metadata for views using the CHANGES clause for SELECT statements.

E. Enabling change tracking adds a pair of hidden columns to the table and begins storing change tracking metadata. The values in these hidden CDC data columns provide the input for the stream metadata columns. The columns consume a small amount of storage.

Answer: C

Explanation:

A stream object records data manipulation language (DML) changes made to tables, including inserts, updates, and deletes, as well as metadata about each change, so that actions can be taken using the changed data. This process is referred to as change data capture (CDC). An individual table stream tracks the changes made to rows in a source table. A table stream (also referred to as simply a "stream") makes a "change table" available of what changed, at the row level, between two transactional points of time in a table. This allows querying and consuming a sequence of change records in a transactional fashion.

Streams can be created to query change data on the following objects:

· Standard tables, including shared tables.

· Views, including secure views

· Directory tables

· External tables

When created, a stream logically takes an initial snapshot of every row in the source object (e.g. table, external table, or the underlying tables for a view) by initializing a point in time (called an off-set) as the current transactional version of the object. The change tracking system utilized by the stream then records information about the DML changes after this snapshot was taken. Change records provide the state of a row before and after the change. Change information mirrors the column structure of the tracked source object and includes additional metadata columns that describe each change event.

Note that a stream itself does not contain any table data. A stream only stores an offset for the source object and returns CDC records by leveraging the versioning history for the source object. When the first stream for a table is created, a pair of hidden columns are added to the source table and begin storing change tracking metadata. These columns consume a small amount of storage. The CDC records returned when querying a stream rely on a combination of the offset stored in the stream and the change tracking metadata stored in the table. Note that for streams on views, change tracking must be enabled explicitly for the view and underlying tables to add the hidden columns to these tables.

Streams on views support both local views and views shared using Snowflake Secure Data Sharing, including secure views. Currently, streams cannot track changes in materialized views.

Views with the following operations are not yet supported:

· GROUP BY clauses

· QUALIFY clauses

· Subqueries not in the FROM clause

· Correlated subqueries

· LIMIT clauses

Change Tracking:

Change tracking must be enabled in the underlying tables.

Prior to creating a stream on a view, you must enable change tracking on the underlying tables for the view.

Set the CHANGE_TRACKING parameter when creating a view (using CREATE VIEW) or later (using ALTER VIEW).

As an alternative to streams, Snowflake supports querying change tracking metadata for tables or views using the CHANGES clause for SELECT statements. The CHANGES clause enables querying change tracking metadata between two points in time without having to create a stream with an explicit transactional offset.

5.To advance the offset of a stream to the current table version without consuming the change data in a DML operation, which of the following operations can be done by Data Engineer? [Select 2]

A. using the CREATE OR REPLACE STREAM syntax, Recreate the STREAM

B. Insert the current change data into a temporary table. In the INSERT statement, query the stream but include a WHERE clause that filters out all of the change data (e.g. WHERE 0 = 1).

C. A stream advances the offset only when it is used in a DML transaction, so none of the options works without consuming the change data of table.

D. Delete the offset using STREAM properties SYSTEM$RESET_OFFSET(<stream_id>)

Answer: A, B

Explanation:

When created, a stream logically takes an initial snapshot of every row in the source object (e.g. table, external table, or the underlying tables for a view) by initializing a point in time (called an off-set) as the current transactional version of the object. The change tracking system utilized by the stream then records information about the DML changes after this snapshot was taken. Change records provide the state of a row before and after the change. Change information mirrors the column structure of the tracked source object and includes additional metadata columns that describe each change event.

Note that a stream itself does not contain any table data. A stream only stores an offset for the source object and returns CDC records by leveraging the versioning history for the source object.

A new table version is created whenever a transaction that includes one or more DML statements is committed to the table.

In the transaction history for a table, a stream offset is located between two table versions. Querying a stream returns the changes caused by transactions committed after the offset and at or before the current time.

Multiple queries can independently consume the same change data from a stream without changing the offset. A stream advances the offset only when it is used in a DML transaction. This behavior applies to both explicit and autocommit transactions. (By default, when a DML statement is executed, an autocommit transaction is implicitly started and the transaction is committed at the completion of the statement. This behavior is controlled with the AUTOCOMMIT parameter.) Querying a stream alone does not advance its offset, even within an explicit transaction; the stream contents must be consumed in a DML statement.

To advance the offset of a stream to the current table version without consuming the change data in a DML operation, complete either of the following actions:

· Recreate the stream (using the CREATE OR REPLACE STREAM syntax).

Insert the current change data into a temporary table. In the INSERT statement, query the stream but include a WHERE clause that filters out all of the change data (e.g. WHERE 0 = 1).