DATABRICKS-CERTIFIED-PR OFESSIONAL-DATA-ENGINEER Q&As

Databricks Certified Professional Data Engineer Exam

Pass Databricks DATABRICKS-CERTIFIED-PROFESSIONAL-DATA-ENGINEER Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/databricks-certified-professional-data-engineer.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Databricks Official Exam Center

QUESTION 1

An hourly batch job is configured to ingest data files from a cloud object storage container where each batch represent all records produced by the source system in a given hour. The batch job to process these records into the Lakehouse is sufficiently delayed to ensure no late-arriving data is missed. Theuser_idfield represents a unique key for the data, which has the following schema:

user_id BIGINT, username STRING, user_utc STRING, user_region STRING, last_login BIGINT, auto_pay BOOLEAN, last_updated BIGINT

New records are all ingested into a table namedaccount_historywhich maintains a full record of all data in the same schema as the source. The next table in the system is namedaccount_currentand is implemented as a Type 1 table representing the most recent value for each uniqueuser_id.

Assuming there are millions of user accounts and tens of thousands of records processed hourly, which implementation can be used to efficiently update the describedaccount_currenttable as part of each hourly batch job?

A. Use Auto Loader to subscribe to new files in the account history directory; configure a Structured Streaminq trigger once job to batch update newly detected files into the account current table.

B. Overwrite the account current table with each batch using the results of a query against the account history table grouping by user id and filtering for the max value of last updated.

C. Filter records in account history using the last updated field and the most recent hour processed, as well as the max last iogin by user id write a merge statement to update or insert the most recent value for each user id.

D. Use Delta Lake version history to get the difference between the latest version of account history and one version prior, then write these records to account current.

E. Filter records in account history using the last updated field and the most recent hour processed, making sure to deduplicate on username; write a merge statement to update or insert the most recent value for each username.

Correct Answer: C

Explanation: This is the correct answer because it efficiently updates the account current table with only the most recent value for each user id. The code filters records in account history using the last updated field and the most recent hour processed, which means it will only process the latest batch of data. It also filters by the max last login by user id, which means it will only keep the most recent record for each user id within that batch. Then, it writes a merge statement to update or insert the most recent value for each user id into account current, which means it will perform an upsert operation based on the user id column. Verified References: [Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "Upsert into a table using merge" section.

QUESTION 2

Which of the following is true of Delta Lake and the Lakehouse?

A. Because Parquet compresses data row by row. strings will only be compressed when a character is repeated multiple times.

B. Delta Lake automatically collects statistics on the first 32 columns of each table which are leveraged in data skipping based on query filters.

C. Views in the Lakehouse maintain a valid cache of the most recent versions of source tables at all times.

D. Primary and foreign key constraints can be leveraged to ensure duplicate values are never entered into a dimension table.

E. Z-order can only be applied to numeric values stored in Delta Lake tables

Correct Answer: A

Explanation: This is the correct answer because it is true of Delta Lake and the Lakehouse. Delta Lake uses Parquet as the underlying storage format for data files. Parquet is a columnar format that compresses data by column rather than by row. This means that Parquet can achieve high compression ratios for columns that have low cardinality or high repetition of values, such as integers, booleans, or dates. However, for columns that have high cardinality or low repetition of values, such as strings, Parquet cannot compress data very well. Therefore, strings will only be compressed when a character is repeated multiple times within a row. Verified References:[Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "Delta Lake core features - Schema enforcement and evolution" section.

QUESTION 3

Which statement regarding stream-static joins and static Delta tables is correct?

A. Each microbatch of a stream-static join will use the most recent version of the static Delta table as of each microbatch.

B. Each microbatch of a stream-static join will use the most recent version of the static Delta table as of the job\\'s initialization.

C. The checkpoint directory will be used to track state information for the unique keys present in the join.

D. Stream-static joins cannot use static Delta tables because of consistency issues.

E. The checkpoint directory will be used to track updates to the static Delta table.

Correct Answer: A

Explanation: This is the correct answer because stream-static joins are supported by Structured Streaming when one of the tables is a static Delta table. A static Delta table is a Delta table that is not updated by any concurrent writes, such as appends or merges, during the execution of a streaming query. In this case, each microbatch of a stream-static join will use the most recent version of the static Delta table as of each microbatch, which means it will reflect any changes made to the static Delta table before the start of each microbatch. Verified References:[Databricks Certified Data Engineer Professional], under "Structured Streaming" section; Databricks Documentation, under "Stream and static joins" section.

QUESTION 4

The data engineering team maintains a table of aggregate statistics through batch nightly updates. This includes total sales for the previous day alongside totals and averages for a variety of time periods including the 7 previous days, year-todate, and quarter-to-date. This table is namedstore_saies_summaryand the schema is as follows:

```
store_id INT, total_sales_qtd FLOAT, avg_daily_sales_qtd FLOAT,  total_sales_ytd FLOAT,
avg_daily_sales_ytd FLOAT, previous_day_sales FLOAT, total_sales_7d FLOAT, avg_daily_sales_7d
FLOAT, updated TIMESTAMP
```

The tabledaily_store_salescontains all the information needed to update store_sales_summary. The schema for this table is:

store_id INT, sales_date DATE, total_sales FLOAT

Ifdaily_store_salesis implemented as a Type 1 table and thetotal_salescolumn might be adjusted after manual data auditing, which approach is the safest to generate accurate reports in thestore_sales_summarytable?

A. Implement the appropriate aggregate logic as a batch read against the daily_store_sales table and overwrite the store_sales_summary table with each Update.

B. Implement the appropriate aggregate logic as a batch read against the daily_store_sales table and append new rows nightly to the store_sales_summary table.

C. Implement the appropriate aggregate logic as a batch read against the daily_store_sales table and use upsert logic to update results in the store_sales_summary table.

D. Implement the appropriate aggregate logic as a Structured Streaming read against the daily_store_sales table and use upsert logic to update results in the store_sales_summary table.

E. Use Structured Streaming to subscribe to the change data feed for daily_store_sales and apply changes to the aggregates in the store_sales_summary table with each update.

Correct Answer: E

Explanation: The daily_store_sales table contains all the information needed to update store_sales_summary. The schema of the table is: store_id INT, sales_date DATE, total_sales FLOAT The daily_store_sales table is implemented as a Type 1 table, which means that old values are overwritten by new values and no history is maintained. The total_sales column might be adjusted after manual data auditing, which means that the data in the table may change over time. The safest approach to generate accurate reports in the store_sales_summary table is to use Structured Streaming to subscribe to the change data feed for daily_store_sales and apply changes to the aggregates in the store_sales_summary table with each update. Structured Streaming is a scalable and fault-tolerant stream processing engine built on Spark SQL. Structured Streaming allows processing data streams as if they were tables or DataFrames, using familiar operations such as select, filter, groupBy, or join. Structured Streaming also supports output modes that specify how to write the results of a streaming query to a sink, such as append, update, or complete. Structured Streaming can handle both streaming and batch data sources in a unified manner. The change data feed is a feature of Delta Lake that provides structured streaming sources that can subscribe to changes made to a Delta Lake table. The change data feed captures both data changes and schema changes as ordered events that can be processed by downstream applications or services. The change data feed can be configured with different options, such as starting from a specific version or timestamp, filtering by operation type or partition values, or excluding no-op changes. By using Structured Streaming to subscribe to the change data feed for daily_store_sales, one can capture and process any changes made to the total_sales column due to manual data auditing. By applying these changes to the aggregates in the store_sales_summary table with each update, one can ensure that the reports are always consistent and accurate with the latest data. Verified References: [Databricks Certified Data Engineer Professional], under "Spark Core" section; Databricks Documentation, under "Structured Streaming" section; Databricks Documentation, under "Delta Change Data Feed" section.

QUESTION 5

A table is registered with the following code:

```
CREATE TABLE recent_orders AS (
  SELECT a.user_id, a.email, b.order_id, b.order_date
  FROM
    (SELECT user_id, email
    FROM users) a
    INNER JOIN
    (SELECT user_id, order_id, order_date
    FROM orders
    WHERE order_date >= (current_date() - 7)) b
    ON a.user_id = b.user_id
)
```

Bothusersandordersare Delta Lake tables. Which statement describes the results of queryingrecent_orders?

A. All logic will execute at query time and return the result of joining the valid versions of the source tables at the time the query finishes.

B. All logic will execute when the table is definedand store the result of joiningtables to the DBFS; this stored data will be returned when the table is queried.

C. Results will be computed and cached when the table is defined; these cached results will incrementally update as new records are inserted into source tables.

D. All logic will execute at query time and return the result of joining the valid versions of the source tables at the time the query began.

E. The versions of each source table will be stored in the table transaction log; query results will be saved to DBFS with each query.

Correct Answer: D

Explanation: This is the correct answer because Delta Lake supports time travel, which allows users to query data as of a specific version or timestamp. The code uses the VERSION AS OF syntax to specify the version of each source table to be used in the join. The result of querying recent_orders will be the same as joining those versions of the source tables at query time. The query will use snapshot isolation, which means it will use a consistent snapshot of the table at the time the query began, regardless of any concurrent updates or deletes. Verified References: [Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "Query an older snapshot of a table (time travel)" section.