Vendor: Databricks

Exam Code:DATABRICKS-CERTIFIED-PROFESSIONAL-DATA-ENGINEER

Exam Name: Databricks Certified Professional Data Engineer Exam

Version: Demo

QUESTION 1

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 2

A table in the Lakehouse namedcustomer_churn_paramsis used in churn prediction by the machine learning team. The table contains information about customers derived from a number of upstream sources. Currently, the data engineering

team populates this table nightly by overwriting the table with the current valid values derived from upstream data sources.

The churn prediction model used by the ML team is fairly stable in production. The team is only interested in making predictions on records that have changed in the past 24 hours.

Which approach would simplify the identification of these changed records?

A. Apply the churn model to all rows in the customer_churn_params table, but implement logic to perform an upsert into the predictions table that ignores rows where predictions have not changed.

B. Convert the batch job to a Structured Streaming job using the complete output mode; configure a Structured Streaming job to read from the customer_churn_params table and incrementally predict against the churn model.

C. Calculate the difference between the previous model predictions and the current customer_churn_params on a key identifying unique customers before making new predictions; only make predictions on those customers not in the previous predictions.

D. Modify the overwrite logic to include a field populated by calling spark.sql.functions.current_timestamp() as data are being written; use this field to identify records written on a particular date.

E. Replace the current overwrite logic with a merge statement to modify only those records that have changed; write logic to make predictions on the changed records identified by the change data feed.

Correct Answer: B

Explanation: This is the correct answer because the JSON posted to the Databricks REST API endpoint 2.0/jobs/create defines a new job with an existing cluster id and a notebook task, but also specifies a new cluster spec with some

configurations. According to the documentation, if both an existing cluster id and a new cluster spec are provided, then a new cluster will be created for each run of the job with those configurations, and then terminated after completion.

Therefore, the logic defined in the referenced notebook will be executed three times on new clusters with those configurations. Verified References:

[Databricks Certified Data Engineer Professional], under "Monitoring and Logging" section; Databricks Documentation, under "JobsClusterSpecNewCluster" section.

QUESTION 3

Which distribution does Databricks support for installing custom Python code packages?

A. sbt

B. CRAN

C. CRAM

D. nom

E. Wheels

F. jars

Correct Answer: D

QUESTION 4

A Delta table of weather records is partitioned by date and has the below schema:

date DATE, device_id INT, temp FLOAT, latitude FLOAT, longitude FLOAT

To find all the records from within the Arctic Circle, you execute a query with the below filter:

latitude > 66.3

Which statement describes how the Delta engine identifies which files to load?

A. All records are cached to an operational database and then the filter is applied

B. The Parquet file footers are scanned for min and max statistics for the latitude column

C. All records are cached to attached storage and then the filter is applied

D. The Delta log is scanned for min and max statistics for the latitude column

E. The Hive metastore is scanned for min and max statistics for the latitude column

Correct Answer: D

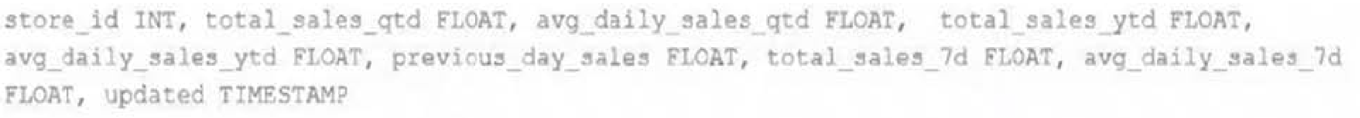
Explanation: This is the correct answer because Delta Lake uses a transaction log to store metadata about each table, including min and max statistics for each column in each data file. The Delta engine can use this information to quickly

identify which files to load based on a filter condition, without scanning the entire table or the file footers. This is called data skipping and it can improve query performance significantly. Verified References:

[Databricks Certified Data Engineer Professional], under "Delta Lake" section; [Databricks Documentation], under "Optimizations - Data Skipping" section.

QUESTION 5

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 6

The data architect has mandated that all tables in the Lakehouse should be configured as external Delta Lake tables.

Which approach will ensure that this requirement is met?

A. Whenever a database is being created, make sure that the location keyword is used

B. When configuring an external data warehouse for all table storage. leverage Databricks for all ELT.

C. Whenever a table is being created, make sure that the location keyword is used.

D. When tables are created, make sure that the external keyword is used in the create table statement.

E. When the workspace is being configured, make sure that external cloud object storage has been mounted.

Correct Answer: C

Explanation: This is the correct answer because it ensures that this requirement is met. The requirement is that all tables in the Lakehouse should be configured as external Delta Lake tables. An external table is a table that is stored outside of the default warehouse directory and whose metadata is not managed by Databricks. An external table can be created by using the location keyword to specify the path to an existing directory in a cloud storage system, such as DBFS or S3. By creating external tables, the data engineering team can avoid losing data if they drop or overwrite the table, as well as leverage existing data without moving or copying it. Verified References: [Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "Create an external table" section.

QUESTION 7

A Databricks job has been configured with 3 tasks, each of which is a Databricks notebook. Task A does not depend on other tasks. Tasks B and C run in parallel, with each having a serial dependency on task A.

If tasks A and B complete successfully but task C fails during a scheduled run, which statement describes the resulting state?

A. All logic expressed in the notebook associated with tasks A and B will have been successfully completed; some operations in task C may have completed successfully.

B. All logic expressed in the notebook associated with tasks A and B will have been successfully completed; any changes made in task C will be rolled back due to task failure.

C. All logic expressed in the notebook associated with task A will have been successfully completed; tasks B and C will not commit any changes because of stage failure.

D. Because all tasks are managed as a dependency graph, no changes will be committed to the Lakehouse until ail tasks have successfully been completed.

E. Unless all tasks complete successfully, no changes will be committed to the Lakehouse; because task C failed, all commits will be rolled back automatically.

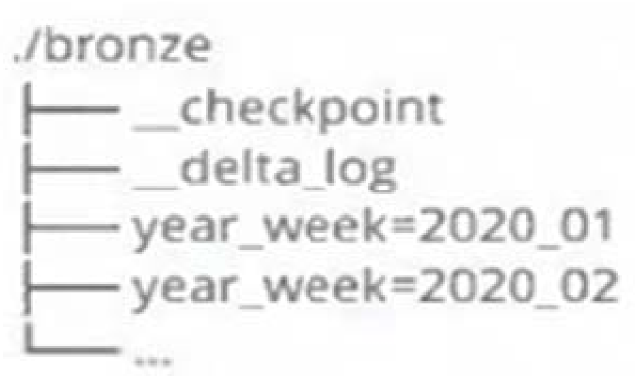
Correct Answer: A

Explanation: The query uses the CREATE TABLE USING DELTA syntax to create a Delta Lake table from an existing Parquet file stored in DBFS. The query also uses the LOCATION keyword to specify the path to the Parquet file as /mnt/finance_eda_bucket/tx_sales.parquet. By using the LOCATION keyword, the query creates an external table, which is a table that is stored outside of the default warehouse directory and whose metadata is not managed by Databricks. An external table can be created from an existing directory in a cloud storage system, such as DBFS or S3, that contains data files in a supported format, such as Parquet or CSV. The resulting state after running the second command is that an external table will be created in the storage container mounted to /mnt/finance_eda_bucket with the new name prod.sales_by_store. The command will not change any data or move any files in the storage container; it will only update the table reference in the metastore and create a new Delta transaction log for the renamed table. Verified References: [Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "ALTER TABLE RENAME TO" section; Databricks Documentation, under "Create an external table" section.

QUESTION 8

A data architect has designed a system in which two Structured Streaming jobs will concurrently write to a single bronze Delta table. Each job is subscribing to a different topic from an Apache Kafka source, but they will write data with the same schema. To keep the directory structure simple, a data engineer has decided to nest a checkpoint directory to be shared by both streams.

The proposed directory structure is displayed below:



```
./bronze
├── _checkpoint
├── _delta_log
├── year_week=2020_01
├── year_week=2020_02
└── ...
```

Which statement describes whether this checkpoint directory structure is valid for the given scenario and why?

A. No; Delta Lake manages streaming checkpoints in the transaction log.

B. Yes; both of the streams can share a single checkpoint directory.

C. No; only one stream can write to a Delta Lake table.

D. Yes; Delta Lake supports infinite concurrent writers.

E. No; each of the streams needs to have its own checkpoint directory.

Correct Answer: E

Explanation: This is the correct answer because checkpointing is a critical feature of Structured Streaming that provides fault tolerance and recovery in case of failures. Checkpointing stores the current state and progress of a streaming query in a reliable storage system, such as DBFS or S3. Each streaming query must have its own checkpoint directory that is unique and exclusive to that query. If two streaming queries share the same checkpoint directory, they will interfere with each other and cause unexpected errors or data loss. Verified References: [Databricks Certified Data Engineer Professional], under "Structured Streaming" section; Databricks Documentation, under "Checkpointing" section.

QUESTION 9

What statement is true regarding the retention of job run history?

A. It is retained until you export or delete job run logs

B. It is retained for 30 days, during which time you can deliver job run logs to DBFS or S3

C. t is retained for 60 days, during which you can export notebook run results to HTML

D. It is retained for 60 days, after which logs are archived

E. It is retained for 90 days or until the run-id is re-used through custom run configuration

Correct Answer: B

Explanation: This is the correct answer because it is true regarding the retention of job run history. Job run history is the information about each run of a job, such as the start time, end time, status, logs, and output. Job run history is retained for 30 days by default, during which time you can view it in the Jobs UI or access it through the Jobs API. You can also deliver job run logs to DBFS or S3 using the Log Delivery feature, which allows you to specify a destination path and a delivery frequency for each job. By delivering job run logs to DBFS or S3, you can preserve them beyond the 30-day retention period and use them for further analysis or troubleshooting. Verified References: [Databricks Certified Data Engineer Professional], under "Databricks Jobs" section;Databricks Documentation, under "Job run history" section; Databricks Documentation, under "Log Delivery" section.

QUESTION 10

The DevOps team has configured a production workload as a collection of notebooks scheduled to run daily using the Jobs UI. A new data engineering hire is onboarding to the team and has requested access to one of these notebooks to review the production logic.

What are the maximum notebook permissions that can be granted to the user without allowing accidental changes to production code or data?

A. Can Manage

B. Can Edit

C. No permissions

D. Can Read

E. Can Run

Correct Answer: D

Explanation: This is the correct answer because it is the maximum notebook permissions that can be granted to the user without allowing accidental changes to production code or data. Notebook permissions are used to control access to notebooks in Databricks workspaces. There are four types of notebook permissions: Can Manage, Can Edit, Can Run, and Can Read. Can Manage allows full control over the notebook, including editing, running, deleting, exporting, and changing permissions. Can Edit allows modifying and running the notebook, but not changing permissions or deleting it. Can Run allows executing commands in an existing cluster attached to the notebook, but not modifying or exporting it. Can Read allows viewing the notebook content, but not running or modifying it. In this case, granting Can Read permission to the user will allow them to review the production logic in the notebook without allowing them to makeany changes to it or run any commands that may affect production data. Verified References: [Databricks Certified Data Engineer Professional], under "Databricks Workspace" section; Databricks Documentation, under "Notebook permissions" section.

QUESTION 11

To reduce storage and compute costs, the data engineering team has been tasked with curating a series of aggregate tables leveraged by business intelligence dashboards, customer-facing applications, production machine learning models, and ad hoc analytical queries.

The data engineering team has been made aware of new requirements from a customer- facing application, which is the only downstream workload they manage entirely. As a result, an aggregate table used by numerous teams across the organization will need to have a number of fields renamed, and additional fields will also be added.

Which of the solutions addresses the situation while minimally interrupting other teams in the organization without increasing the number of tables that need to be managed?

A. Send all users notice that the schema for the table will be changing; include in the communication the logic necessary to revert the new table schema to match historic queries.

B. Configure a new table with all the requisite fields and new names and use this as the source for the customer-facing application; create a view that maintains the original data schema and table name by aliasing select fields from the new table.

C. Create a new table with the required schema and new fields and use Delta Lake\\'s deep clone functionality to sync up changes committed to one table to the corresponding table.

D. Replace the current table definition with a logical view defined with the query logic currently writing the aggregate table; create a new table to power the customer-facing application.

E. Add a table comment warning all users that the table schema and field names will be changing on a given date; overwrite the table in place to the specifications of the customer- facing application.

Correct Answer: B

Explanation: This is the correct answer because it addresses the situation while minimally interrupting other teams in the organization without increasing the number of tables that need to be managed. The situation is that an aggregate table used by numerous teams across the organization will need to have a number of fields renamed, and additional fields will also be added, due to new requirements from a customer-facing application. By configuring a new table with all the requisite fields and new names and using this as the source for the customer-facing application, the data engineering team can meet the new requirements without affecting other teams that rely on the existing table schema and name. By creating a view that maintains the original data schema and table name by aliasing select fields from the new table, the data engineering team can also avoid duplicating data or creating additional tables that need to be managed. Verified References: [Databricks Certified Data Engineer Professional], under "Lakehouse" section; Databricks Documentation, under "CREATE VIEW" section.

QUESTION 12

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.