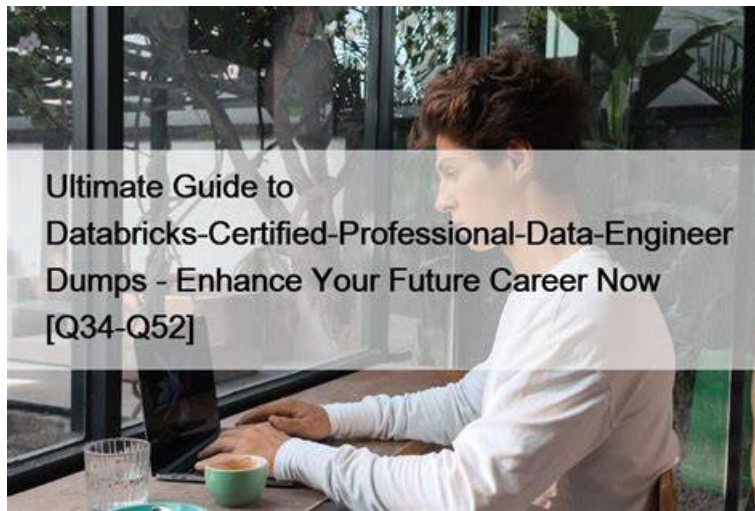


Databricks Databricks-Certified-Professional-Data-Engineer Dump File, Reliable Databricks-Certified-Professional-Data-Engineer Test Sims



The Databricks Certified Professional Data Engineer Exam (Databricks-Certified-Professional-Data-Engineer) questions are being offered in three easy-to-use and different formats. These formats are Databricks Dumps PDF, desktop-based Databricks Databricks-Certified-Professional-Data-Engineer practice test software, and web-based Databricks-Certified-Professional-Data-Engineer practice exam. All these three Databricks-Certified-Professional-Data-Engineer Exam Dumps formats contain real, valid, and updated Databricks-Certified-Professional-Data-Engineer exam questions that surely repeat in the upcoming Databricks-Certified-Professional-Data-Engineer exam and you can easily pass the Databricks Databricks-Certified-Professional-Data-Engineer exam on the first attempt.

Databricks Databricks-Certified-Professional-Data-Engineer Exam Syllabus Topics:

Section	Weight	Objectives
Data Processing with Spark	25-30%	- Spark DataFrames and Spark SQL • 1. Spark SQL queries and functions • 2. DataFrame operations and transformations • 3. Window functions - Python and SQL for data engineering • 1. Spark APIs in Python • 2. Performance optimization techniques • 3. Built-in and user-defined functions
Pipeline Development and Orchestration	10-15%	- Databricks workflows • 1. Jobs and job scheduling • 2. Task dependencies and orchestration • 3. Monitoring and alerting

Delta Lake	20-25%	- Delta Lake operations • 1. Delta Live Tables • 2. Merge, update, delete operations • 3. Schema evolution and enforcement - Delta Lake fundamentals • 1. Time travel and data versioning • 2. ACID transactions • 3. Optimize and Z-order
Data Ingestion	15-20%	- Batch ingestion methods • 1. Spark APIs for ingestion • 2. Integration with external systems • 3. DBR autoloader - Streaming ingestion • 1. Kafka integration • 2. Structured streaming fundamentals
Data Warehouse and Lakehouse Architecture	15-20%	- Lakehouse architecture principles • 1. Data governance fundamentals • 2. Differences between data lake, data warehouse, and lakehouse • 3. Bronze, silver, gold data layers

>> **Databricks Databricks-Certified-Professional-Data-Engineer Dump File** <<

Reliable Databricks-Certified-Professional-Data-Engineer Test Sims - Valid Databricks-Certified-Professional-Data-Engineer Study Notes

Our Databricks-Certified-Professional-Data-Engineer exam questions are valuable and useful and if you buy our Databricks-Certified-Professional-Data-Engineer study materials will provide first-rate service to you to make you satisfied. We provide not only the free download and try out of the Databricks-Certified-Professional-Data-Engineer Practice Guide but also the immediate download after your purchase successfully. To see whether our Databricks-Certified-Professional-Data-Engineer training dumps are worthy to buy, you can have a try on our product right now.

Databricks Certified Professional Data Engineer Exam Sample Questions (Q112-Q117):

NEW QUESTION # 112

A data engineering team uses Databricks Lakehouse Monitoring to track the percent_null metric for a critical column in their Delta table.

The profile metrics table (prod_catalog.prod_schema.customer_data_profile_metrics) stores hourly percent_null values.

The team wants to:

Trigger an alert when the daily average of percent_null exceeds 5% for three consecutive days.

Ensure that notifications are not spammed during sustained issues.

Options:

- A. SELECT SUM(CASE WHEN percent_null > 5 THEN 1 ELSE 0 END) AS violation_days FROM prod_catalog.prod_schema.customer_data_profile_metrics WHERE window.end >= CURRENT_TIMESTAMP - INTERVAL '3' DAY Alert Condition: violation_days >= 3 Notification Frequency: Just once
- B. SELECT percent_null FROM prod_catalog.prod_schema.customer_data_profile_metrics WHERE window.end >= CURRENT_TIMESTAMP - INTERVAL '1' DAY Alert Condition: percent_null > 5

Notification Frequency: At most every 24 hours

- C. `SELECT AVG(percent_null) AS daily_avg`
`FROM prod_catalog.prod_schema.customer_data_profile_metrics`
`WHERE window.end >= CURRENT_TIMESTAMP - INTERVAL '3' DAY`
Alert Condition: `daily_avg > 5`
Notification Frequency: Each time alert is evaluated
- D. `WITH daily_avg AS (`
`SELECT DATE_TRUNC('DAY', window.end) AS day,`
`AVG(percent_null) AS avg_null`
`FROM prod_catalog.prod_schema.customer_data_profile_metrics`
`GROUP BY DATE_TRUNC('DAY', window.end)`
`)`
`SELECT day, avg_null`
`FROM daily_avg`
`ORDER BY day DESC`
`LIMIT 3`
Alert Condition: `ALL avg_null > 5` for the latest 3 rows
Notification Frequency: Just once

Answer: D

Explanation:

The key requirement is to detect when the daily average of percent_null is greater than 5% for three consecutive days.

Option A only checks the last 24 hours, not consecutive days. It would trigger too frequently and cause spam.

Option C calculates an average across all records in the last 3 days, but this could be skewed by one high or low day - it does not ensure consecutive daily violations.

Option D simply counts days where the threshold was exceeded, but it does not guarantee that those days were consecutive. This could incorrectly trigger on non-adjacent violations.

Option B is correct:

It aggregates hourly values into daily averages.

It checks that the last 3 consecutive days all had averages above 5%.

It avoids redundant alerts by using Notification Frequency: Just once.

This matches Databricks Lakehouse Monitoring best practices, where SQL alerts should be designed to aggregate metrics to the correct granularity (daily here) and ensure consecutive threshold violations before triggering.

Reference (Databricks Lakehouse Monitoring, SQL Alerts Best Practices):

Use `DATE_TRUNC` to compute metrics at the correct time granularity.

To detect consecutive-day issues, filter the last N daily aggregates and check conditions across all rows.

Always configure alerts with controlled notification frequency to prevent alert fatigue.

NEW QUESTION # 113

A Delta Lake table was created with the below query:

Consider the following query:

```
DROP TABLE prod.sales_by_store -
```

If this statement is executed by a workspace admin, which result will occur?

- A. An error will occur because Delta Lake prevents the deletion of production data.
- B. The table will be removed from the catalog but the data will remain in storage.
- C. The table will be removed from the catalog and the data will be deleted.
- D. Data will be marked as deleted but still recoverable with Time Travel.
- E. Nothing will occur until a COMMIT command is executed.

Answer: C

Explanation:

When a table is dropped in Delta Lake, the table is removed from the catalog and the data is deleted. This is because Delta Lake is a transactional storage layer that provides ACID guarantees. When a table is dropped, the transaction log is updated to reflect the deletion of the table and the data is deleted from the underlying storage. References:

* <https://docs.databricks.com/delta/quick-start.html#drop-a-table>

* <https://docs.databricks.com/delta/delta-batch.html#drop-table>

NEW QUESTION # 114

The view `updates` represents an incremental batch of all newly ingested data to be inserted or updated in the `customers` table. The following logic is used to process these records.

```
MERGE INTO customers
USING (
  SELECT updates.customer_id as merge_key, updates.*
  FROM updates

  UNION ALL

  SELECT NULL as merge_key, updates.*
  FROM updates JOIN customers
  ON updates.customer_id = customers.customer_id
  WHERE customers.current = true AND updates.address <> customers.address
) staged_updates
ON customers.customer_id = merge_key
WHEN MATCHED AND customers.current = true AND customers.address <> staged_updates.address THEN
  UPDATE SET current = false, end_date = staged_updates.effective_date
WHEN NOT MATCHED THEN
  INSERT(customer_id, address, current, effective_date, end_date)
  VALUES(staged_updates.customer_id, staged_updates.address, true, staged_updates.effective_date,
null)
```

Which statement describes this implementation?

- A. The `customers` table is implemented as a Type 0 table; all writes are append only with no changes to existing values.
- B. The `customers` table is implemented as a Type 1 table; old values are overwritten by new values and no history is maintained.
- C. The `customers` table is implemented as a Type 2 table; old values are overwritten and new customers are appended.
- D. The `customers` table is implemented as a Type 3 table; old values are maintained as a new column alongside the current value.
- E. The `customers` table is implemented as a Type 2 table; old values are maintained but marked as no longer current and new values are inserted.

Answer: E

Explanation:

Explanation

The logic uses the `MERGE INTO` command to merge new records from the view `updates` into the table `customers`. The `MERGE INTO` command takes two arguments: a target table and a source table or view. The command also specifies a condition to match records between the target and the source, and a set of actions to perform when there is a match or not. In this case, the condition is to match records by `customer_id`, which is the primary key of the `customers` table. The actions are to update the existing record in the target with the new values from the source, and set the `current_flag` to false to indicate that the record is no longer current; and to insert a new record in the target with the new values from the source, and set the `current_flag` to true to indicate that the record is current. This means that old values are maintained but marked as no longer current and new values are inserted, which is the definition of a Type 2 table. Verified References: [Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "Merge Into (Delta Lake on Databricks)" section.

NEW QUESTION # 115

The following code has been migrated to a Databricks notebook from a legacy workload:

```
%sh
git clone https://github.com/foo/data_loader;
python ./data_loader/run.py;
mv ./output /dbfs/mnt/new_data
```

The code executes successfully and provides the logically correct results, however, it takes over 20 minutes to extract and load around 1 GB of data.

Which statement is a possible explanation for this behavior?

- A. `%sh` triggers a cluster restart to collect and install Git. Most of the latency is related to cluster startup time.
- B. `%sh` does not distribute file moving operations; the final line of code should be updated to use `%fs` instead.
- C. `%sh` executes shell code on the driver node. The code does not take advantage of the worker nodes or Databricks

optimized Spark.

- D. Python will always execute slower than Scala on Databricks. The run.py script should be refactored to Scala.
- E. Instead of cloning, the code should use %sh pip install so that the Python code can get executed in parallel across all nodes in a cluster.

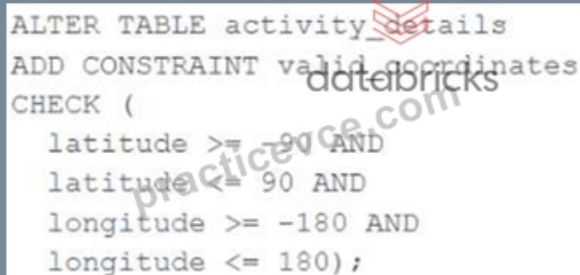
Answer: C

Explanation:

<https://www.databricks.com/blog/2020/08/31/introducing-the-databricks-web-terminal.html> The code is using %sh to execute shell code on the driver node. This means that the code is not taking advantage of the worker nodes or Databricks optimized Spark. This is why the code is taking longer to execute. A better approach would be to use Databricks libraries and APIs to read and write data from Git and DBFS, and to leverage the parallelism and performance of Spark. For example, you can use the Databricks Connect feature to run your Python code on a remote Databricks cluster, or you can use the Spark Git Connector to read data from Git repositories as Spark DataFrames.

NEW QUESTION # 116

A CHECK constraint has been successfully added to the Delta table named activity_details using the following logic:



```
ALTER TABLE activity_details
ADD CONSTRAINT valid_coordinates
CHECK (
  latitude >= -90 AND
  latitude <= 90 AND
  longitude >= -180 AND
  longitude <= 180);
```

A batch job is attempting to insert new records to the table, including a record where latitude = 45.50 and longitude = 212.67. Which statement describes the outcome of this batch insert?

- A. The write will fail completely because of the constraint violation and no records will be inserted into the target table.
- B. The write will fail when the violating record is reached; any records previously processed will be recorded to the target table.
- C. The write will include all records in the target table; any violations will be indicated in the boolean column named valid_coordinates.
- D. The write will insert all records except those that violate the table constraints; the violating records will be reported in a warning log.
- E. The write will insert all records except those that violate the table constraints; the violating records will be recorded to a quarantine table.

Answer: A

Explanation:

The CHECK constraint is used to ensure that the data inserted into the table meets the specified conditions. In this case, the CHECK constraint is used to ensure that the latitude and longitude values are within the specified range. If the data does not meet the specified conditions, the write operation will fail completely and no records will be inserted into the target table. This is because Delta Lake supports ACID transactions, which means that either all the data is written or none of it is written. Therefore, the batch insert will fail when it encounters a record that violates the constraint, and the target table will not be updated. References:

* Constraints : <https://docs.delta.io/latest/delta-constraints.html>

* ACID Transactions : <https://docs.delta.io/latest/delta-intro.html#acid-transactions>

NEW QUESTION # 117

.....

The Databricks Certified Professional Data Engineer Exam (Databricks-Certified-Professional-Data-Engineer) exam questions are the real, valid, and updated Databricks-Certified-Professional-Data-Engineer Exam Questions that are specifically designed for quick and complete Databricks-Certified-Professional-Data-Engineer exam preparation. With PracticeVCE Databricks Certified Professional Data Engineer Exam (Databricks-Certified-Professional-Data-Engineer) practice test questions you can start Databricks Databricks-Certified-Professional-Data-Engineer exam preparation immediately.

Reliable Databricks-Certified-Professional-Data-Engineer Test Sims: <https://www.practicevce.com/Databricks/Databricks-Certified-Professional-Data-Engineer-practice-exam-dumps.html>

- [illegible]