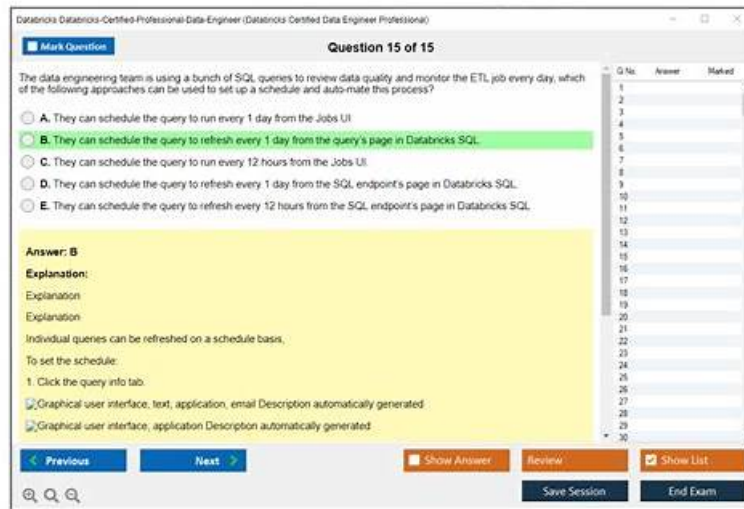


# Databricks-Certified-Professional-Data-Engineer Actual Test Pdf & Databricks-Certified-Professional-Data-Engineer Exam Outline



DOWNLOAD the newest DumpExam Databricks-Certified-Professional-Data-Engineer PDF dumps from Cloud Storage for free: <https://drive.google.com/open?id=1NkkHw7LxIJL5QveOabulcodjs8-WXoaY>

DumpExam deeply hope our Databricks-Certified-Professional-Data-Engineer study materials can bring benefits and profits for our customers. So we have been persisting in updating our Databricks-Certified-Professional-Data-Engineer test torrent and trying our best to provide customers with the latest study materials. More importantly, the updating system we provide is free for all customers. If you decide to buy our Databricks-Certified-Professional-Data-Engineer Study Materials, we can guarantee that you will have the opportunity to use the updating system for free.

Databricks Certified Professional Data Engineer certification exam is a valuable certification for data professionals who want to demonstrate their skills in building and managing complex data solutions on the Databricks platform. Databricks-Certified-Professional-Data-Engineer Exam is designed to test the knowledge and skills required to design, implement, and manage data engineering workflows in a collaborative environment. Candidates who pass the exam will have a deep understanding of Databricks technologies and will be able to build and manage scalable and reliable data solutions.

Databricks Certified Professional Data Engineer Exam is an excellent way for data engineers to demonstrate their expertise in working with the Databricks platform. It provides employers with a way to validate the skills of potential candidates and ensures that data engineers have the knowledge and skills needed to work with Databricks effectively.

>>> **Databricks-Certified-Professional-Data-Engineer Actual Test Pdf** <<<

## Explore the Databricks Databricks-Certified-Professional-Data-Engineer Online Practice Test Engine

Great concentrative progress has been made by our company, who aims at further cooperation with our candidates in the way of using our Databricks-Certified-Professional-Data-Engineer exam engine as their study tool. Owing to the devotion of our professional research team and responsible working staff, our Databricks-Certified-Professional-Data-Engineer Training Materials have received wide recognition and now, with more people joining in the Databricks-Certified-Professional-Data-Engineer exam army, we has become the top-raking Databricks-Certified-Professional-Data-Engineer learning guide provider in the international market.

## Databricks Certified Professional Data Engineer Exam Sample Questions (Q105-Q110):

NEW QUESTION # 105

If E1 and E2 are two events, how do you represent the conditional probability given that E2 occurs given that E1 has occurred?

- A.  $P(E2)/P(E1)$
- B.  $P(E2)/(P(E1)+E2)$
- C.  $P(E1+E2)/P(E1)$
- D.  $P(E1)/P(E2)$

**Answer: A**

#### NEW QUESTION # 106

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 and the data will be deleted.
- C. The table will be removed from the catalog but the data will remain in storage.
- D. Data will be marked as deleted but still recoverable with Time Travel.
- E. Nothing will occur until a COMMIT command is executed.

**Answer: B**

Explanation:

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 # 107

A Structured Streaming job deployed to production has been experiencing delays during peak hours of the day.

At present, during normal execution, each microbatch of data is processed in less than 3 seconds. During peak hours of the day, execution time for each microbatch becomes very inconsistent, sometimes exceeding 30 seconds. The streaming write is currently configured with a trigger interval of 10 seconds.

Holding all other variables constant and assuming records need to be processed in less than 10 seconds, which adjustment will meet the requirement?

- A. Decrease the trigger interval to 5 seconds; triggering batches more frequently may prevent records from backing up and large batches from causing spill.
- B. Increase the trigger interval to 30 seconds; setting the trigger interval near the maximum execution time observed for each batch is always best practice to ensure no records are dropped.
- C. Use the trigger once option and configure a Databricks job to execute the query every 10 seconds; this ensures all backlogged records are processed with each batch.
- D. The trigger interval cannot be modified without modifying the checkpoint directory; to maintain the current stream state, increase the number of shuffle partitions to maximize parallelism.
- E. Decrease the trigger interval to 5 seconds; triggering batches more frequently allows idle executors to begin processing the next batch while longer running tasks from previous batches finish.

**Answer: A**

Explanation:

Explanation

The adjustment that will meet the requirement of processing records in less than 10 seconds is to decrease the trigger interval to 5 seconds. This is because triggering batches more frequently may prevent records from backing up and large batches from causing spill. Spill is a phenomenon where the data in memory exceeds the available capacity and has to be written to disk, which can slow down the processing and increase the execution time<sup>1</sup>. By reducing the trigger interval, the streaming query can process smaller

batches of data more quickly and avoid spill. This can also improve the latency and throughput of the streaming job<sup>2</sup>.

The other options are not correct, because:

Option A is incorrect because triggering batches more frequently does not allow idle executors to begin processing the next batch while longer running tasks from previous batches finish. In fact, the opposite is true. Triggering batches more frequently may cause concurrent batches to compete for the same resources and cause contention and backpressure<sup>2</sup>. This can degrade the performance and stability of the streaming job.

Option B is incorrect because increasing the trigger interval to 30 seconds is not a good practice to ensure no records are dropped. Increasing the trigger interval means that the streaming query will process larger batches of data less frequently, which can increase the risk of spill, memory pressure, and timeouts<sup>12</sup>. This can also increase the latency and reduce the throughput of the streaming job.

Option C is incorrect because the trigger interval can be modified without modifying the checkpoint directory. The checkpoint directory stores the metadata and state of the streaming query, such as the offsets, schema, and configuration<sup>3</sup>. Changing the trigger interval does not affect the state of the streaming query, and does not require a new checkpoint directory. However, changing the number of shuffle partitions may affect the state of the streaming query, and may require a new checkpoint directory<sup>4</sup>.

Option D is incorrect because using the trigger once option and configuring a Databricks job to execute the query every 10 seconds does not ensure that all backlogged records are processed with each batch. The trigger once option means that the streaming query will process all the available data in the source and then stop<sup>5</sup>. However, this does not guarantee that the query will finish processing within 10 seconds, especially if there are a lot of records in the source. Moreover, configuring a Databricks job to execute the query every 10 seconds may cause overlapping or missed batches, depending on the execution time of the query.

References: Memory Management Overview, Structured Streaming Performance Tuning Guide, Checkpointing, Recovery Semantics after Changes in a Streaming Query, Triggers

### NEW QUESTION # 108

Which statement regarding spark configuration on the Databricks platform is true?

- A. The Databricks REST API can be used to modify the Spark configuration properties for an interactive cluster without interrupting jobs.
- B. Spark configuration set within a notebook will affect all SparkSessions attached to the same interactive cluster
- **C. Spark configuration properties set for an interactive cluster with the Clusters UI will impact all notebooks attached to that cluster.**
- D. When the same spark configuration property is set for an interactive to the same interactive cluster.

**Answer: C**

Explanation:

When Spark configuration properties are set for an interactive cluster using the Clusters UI in Databricks, those configurations are applied at the cluster level. This means that all notebooks attached to that cluster will inherit and be affected by these configurations. This approach ensures consistency across all executions within that cluster, as the Spark configuration properties dictate aspects such as memory allocation, number of executors, and other vital execution parameters. This centralized configuration management helps maintain standardized execution environments across different notebooks, aiding in debugging and performance optimization.

Reference:

Databricks documentation on configuring clusters: <https://docs.databricks.com/clusters/configure.html>

### NEW QUESTION # 109

When using the complete mode to write stream data, how does it impact the target table?

- A. Stream must complete to write the data
- B. Target table cannot be updated while stream is pending
- C. Entire stream waits for complete data to write
- **D. Target table is overwritten for each batch**
- E. Delta commits transaction once the stream is stopped

**Answer: D**

Explanation:

Explanation

The answer is Target table is overwritten for each batch

Complete mode - The whole Result Table will be outputted to the sink after every trigger. This is supported for aggregation queries

• • • • •

**Databricks-Certified-Professional-Data-Engineer Exam Outline:** <https://www.dumpexam.com/Databricks-Certified-Professional-Data-Engineer-valid-torrent.html>

- BTW, DOWNLOAD part of DumpExam Databricks-Certified-Professional-Data-Engineer dumps from Cloud Storage:  
<https://drive.google.com/open?id=1NkkHw7LxIJL5OveOabulcodis8-WXoaY>

