

PRACTICE QUESTIONS

Databricks Certified Data Engineer Professional

SHARED BY [u/kind_ibis_39](#) on aiexamcerts

EDITION Free Set 1

QUESTIONS 10

SHARED October 15, 2026

EXAM Databricks Certified Data Engineer Professional

Shared by u/kind_ibis_39 on aiexamcerts

10 practice questions for the Databricks Certified Data Engineer Professional exam, each followed by its answer and a short explanation. More free sets and the full practice exam:

www.aiexamcerts.com/databricks/databricks-certified-data-engineer-professional. Spotted a mistake? Email support@aiexamcerts.com with the exam name and the question number.

1. A streaming job must process the backlog currently in a source directory in several batches, then stop on its own, and keep using the same checkpoint on later runs. Which trigger should be set?
 - A `trigger(processingTime='1 second')`
 - B `trigger(continuous='1 second')`
 - C `trigger(processingTime='0 seconds')` with a new checkpoint on each run
 - D `trigger(availableNow=True)`

Answer: D

`availableNow` processes all data available when the query starts, in one or more micro-batches, and then stops while keeping checkpoint progress so the next run resumes correctly. A processing-time trigger keeps the query running indefinitely, so it does not stop by itself. Creating a new checkpoint on each run would lose the record of which files were already processed.

2. To read a Delta table as it existed at a given point in time without changing it, append the clause `TIMESTAMP _____` to the table reference.

Answer: AS OF

The clause is `TIMESTAMP AS OF`, and the same pattern works with `VERSION AS OF` and a version number. `RESTORE TABLE` is a different statement that rewrites the current table back to an earlier state, so it is not the read-only option. Writing `TIMESTAMP` without `AS OF` is a syntax error.

3. In a Databricks Workflows job, which mechanism lets one task pass a small value to a downstream task?
 - A Notebook widgets defined in the downstream notebook only
 - B Environment variables set in the cluster policy
 - C `dbutils.jobs.taskValues.set` in the upstream task and `taskValues.get` downstream
 - D A change to `spark.sql.shuffle.partitions`

Answer: C

`dbutils.jobs.taskValues.set` stores a small value that downstream tasks read with `dbutils.jobs.taskValues.get`, which is the supported way to share values between tasks. Notebook widgets only supply parameters to the notebook that defines them when it runs. Cluster policies and shuffle partition settings do not carry values between tasks.

4. The Databricks CLI command `databricks bundle run` only validates the YAML configuration and does not start any job.

- A True
- B False

Answer: B (False)

`databricks bundle` validate checks the configuration, `databricks bundle deploy` uploads resources, and `databricks bundle run` starts a job or pipeline defined in the bundle. Treating `run` as a `validate-only` command confuses it with `validate`. Running the bundle is the step that triggers execution of the deployed resource.

5. Which two statements about granting read access to a table in Unity Catalog are correct? (Choose two.)

- A `GRANT SELECT ON TABLE catalog.schema.sales` TO a group gives that group read access to the table
- B `GRANT USAGE ON` a cluster policy to a group to give it table access
- C Granting `CAN ATTACH TO` on a cluster to the group gives it table access
- D A `SELECT` privilege granted on the schema is inherited by the tables in it
- E Tagging the table with the group name gives that group read access

Answer: A, D

`GRANT SELECT` on a table, or on the containing schema for inheritance, gives the group read access. A cluster policy governs which compute settings users may choose and does not grant table privileges. Cluster permissions control who can use compute, and a tag is only metadata, so neither gives anyone table access.

6. In the medallion architecture, which layer holds cleaned, deduplicated and conformed data that is ready to join across sources?

- A Bronze
- B Silver
- C Gold
- D Platinum

Answer: B

Silver tables apply cleansing, deduplication, type fixes and conformance across sources, so they are reliable building blocks for joins. Bronze keeps raw data as it was ingested, close to the source format. Gold holds business-level aggregates for reporting, and Platinum is not one of the standard medallion layers.

7. A modeler wants the optimizer and documentation tools to know that every order's `customer_id` refers to a row in the customer dimension. Which Unity Catalog feature expresses this relationship?
- A A Delta Sharing share that lists the customer dimension
 - B A row filter
 - C A primary key and foreign key constraint
 - D A column mask

Answer: C

Primary and foreign key constraints declare the relationship in Unity Catalog, which documents the model and can help query planning, although Databricks does not enforce them on writes. A row filter restricts which rows a user sees, and a column mask hides or transforms values. A Delta Sharing share exposes tables to other organizations and does not describe joins.

8. In Spark, a stage that writes shuffle data to local disk because executor memory is full is said to _____.

Answer: spill (also accepted: spill to disk)

This behavior is called spilling, and it slows a stage because intermediate data goes to disk. Caching stores data for reuse, and broadcasting sends a small table to every executor. Neither describes a shortage of execution memory.

9. A streaming query's throughput has dropped, and the team wants per-batch input rates and processing durations. Where should they look first?
- A The Structured Streaming progress metrics and the Streaming tab
 - B The driver's stderr log from the first micro-batch after the restart
 - C The Unity Catalog grants page
 - D The cluster's autoscaling event history

Answer: A

The streaming query progress reports `numInputRows`, input rate and batch durations for each micro-batch, which localizes the throughput drop. The driver's stderr log and the autoscaling history do not break down input rate and duration per micro-batch, and the grants page records only privileges.

10. A team wants unit tests for PySpark transformations that run in CI without a live cluster. Which approach fits?
- A Run the job in production and compare the output to yesterday's by hand
 - B Test only through ad hoc queries in the SQL editor
 - C Run pytest with a local SparkSession and small in-memory DataFrames
 - D Attach the tests to an all-purpose cluster and run them by hand each week

Answer: C

Pytest with a local SparkSession and small DataFrames lets CI verify transformation logic quickly and repeatably, before any deployment. Testing in production exposes bad output to consumers. Ad hoc queries are not repeatable in CI, and a weekly manual run on a live cluster does not give fast feedback on each change.