



DATABRICKS-CERTIFIED- PR OFESIONAL-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.passapply.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



VCE & PDF

PassApply.com

[https://www.passapply.com/databricks-certified-professional-data-engineer.](https://www.passapply.com/databricks-certified-professional-data-engineer)
2024 Latest passapply DATABRICKS-CERTIFIED-PROFESSIONAL-DATA-ENGINEER PDF and VCE dumps Download

- ⚙️ **Instant Download** After Purchase
- ⚙️ **100% Money Back** Guarantee
- ⚙️ **365 Days** Free Update
- ⚙️ **800,000+** Satisfied Customers





QUESTION 1

A junior data engineer has been asked to develop a streaming data pipeline with a grouped aggregation using `DataFrameDf`. The pipeline needs to calculate the average humidity and average temperature for each non-overlapping five-minute interval. Events are recorded once per minute per device.

Streaming `DataFrameDf` has the following schema:

```
"device_id INT, event_time TIMESTAMP, temp FLOAT, humidity FLOAT"
```

Code block:

```
df.withWatermark("event_time", "10 minutes")
  .groupBy(
    _____
    "device_id"
  )
  .agg(
    avg("temp").alias("avg_temp"),
    avg("humidity").alias("avg_humidity")
  )
  .writeStream
  .format("delta")
  .saveAsTable("sensor_avg")
```

Choose the response that correctly fills in the blank within the code block to complete this task.

- A. `to_interval("event_time", "5 minutes").alias("time")`
- B. `window("event_time", "5 minutes").alias("time")`
- C. `"event_time"`
- D. `window("event_time", "10 minutes").alias("time")`
- E. `lag("event_time", "10 minutes").alias("time")`

Correct Answer: B

Explanation: This is the correct answer because the window function is used to group streaming data by time intervals. The window function takes two arguments: a time column and a window duration. The window duration specifies how long each window is, and must be a multiple of 1 second. In this case, the window duration is "5 minutes", which means each window will cover a non-overlapping five-minute interval. The window function also returns a struct column with



two fields: start and end, which represent the start and end time of each window. The alias function is used to rename the struct column as "time". Verified References: [Databricks Certified Data Engineer Professional], under "Structured Streaming" section; Databricks Documentation, under "WINDOW" section.

QUESTION 2

The data governance team is reviewing code used for deleting records for compliance with GDPR. They note the following logic is used to delete records from the Delta Lake table named users.

```
DELETE FROM users
WHERE user_id IN
(SELECT user_id FROM delete_requests)
```

Assuming that user_id is a unique identifying key and that delete_requests contains all users that have requested deletion, which statement describes whether successfully executing the above logic guarantees that the records to be deleted are no longer accessible and why?

- A. Yes; Delta Lake ACID guarantees provide assurance that the delete command succeeded fully and permanently purged these records.
- B. No; the Delta cache may return records from previous versions of the table until the cluster is restarted.
- C. Yes; the Delta cache immediately updates to reflect the latest data files recorded to disk.
- D. No; the Delta Lake delete command only provides ACID guarantees when combined with the merge into command.
- E. No; files containing deleted records may still be accessible with time travel until a vacuum command is used to remove invalidated data files.

Correct Answer: E

Explanation: The code uses the DELETE FROM command to delete records from the users table that match a condition based on a join with another table called delete_requests, which contains all users that have requested deletion. The DELETE FROM command deletes records from a Delta Lake table by creating a new version of the table that does not contain the deleted records. However, this does not guarantee that the records to be deleted are no longer accessible, because Delta Lake supports time travel, which allows querying previous versions of the table using a timestamp or version number. Therefore, files containing deleted records may still be accessible with time travel until a vacuum command is used to remove invalidated data files from physical storage. Verified References: [Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "Delete from a table" section; Databricks Documentation, under "Remove files no longer referenced by a Delta table" section.

QUESTION 3

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 4

The security team is exploring whether or not the Databricks secrets module can be leveraged for connecting to an external database.

After testing the code with all Python variables being defined with strings, they upload the password to the secrets module and configure the correct permissions for the currently active user. They then modify their code to the following (leaving all other variables unchanged).

```
password = dbutils.secrets.get(scope="db_creds", key="jdbc_password")

print(password)

df = (spark
      .read
      .format("jdbc")
      .option("url", connection)
      .option("dbtable", tablename)
      .option("user", username)
      .option("password", password)
      )
```

Which statement describes what will happen when the above code is executed?

- A. The connection to the external table will fail; the string "redacted" will be printed.
- B. An interactive input box will appear in the notebook; if the right password is provided, the connection will succeed and the encoded password will be saved to DBFS.



- C. An interactive input box will appear in the notebook; if the right password is provided, the connection will succeed and the password will be printed in plain text.
- D. The connection to the external table will succeed; the string value of password will be printed in plain text.
- E. The connection to the external table will succeed; the string "redacted" will be printed.

Correct Answer: E

Explanation: This is the correct answer because the code is using the `dbutils.secrets.get` method to retrieve the password from the secrets module and store it in a variable. The secrets module allows users to securely store and access sensitive information such as passwords, tokens, or API keys. The connection to the external table will succeed because the password variable will contain the actual password value. However, when printing the password variable, the string "redacted" will be displayed instead of the plain text password, as a security measure to prevent exposing sensitive information in notebooks. Verified References: [Databricks Certified Data Engineer Professional], under "Security and Governance" section; Databricks Documentation, under "Secrets" section.

QUESTION 5

A Delta Lake table was created with the below query:

```
CREATE TABLE prod.sales_by_stor  
USING DELTA  
LOCATION "/mnt/prod/sales_by_store"
```

Realizing that the original query had a typographical error, the below code was executed:

```
ALTER TABLE prod.sales_by_stor RENAME TO prod.sales_by_store
```

Which result will occur after running the second command?

- A. The table reference in the metastore is updated and no data is changed.
- B. The table name change is recorded in the Delta transaction log.
- C. All related files and metadata are dropped and recreated in a single ACID transaction.
- D. The table reference in the metastore is updated and all data files are moved.
- E. A new Delta transaction log is created for the renamed table.

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 result that will occur after running the second command is that the table reference in the metastore is updated and no data is changed. The metastore is a service that stores metadata about tables, such as their schema, location, properties, and partitions. The metastore allows users to access



tables using SQL commands or Spark APIs without knowing their physical location or format. When renaming an external table using the ALTER TABLE RENAME TO command, only the table reference in the metastore is updated with the new name; no data files or directories are moved or changed in the storage system. The table will still point to the same location and use the same format as before. However, if renaming a managed table, which is a table whose metadata and data are both managed by Databricks, both the table reference in the metastore and the data files in the default warehouse directory are moved and renamed accordingly. Verified References: [Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "ALTER TABLE RENAME TO" section; Databricks Documentation, under "Metastore" section; Databricks Documentation, under "Managed and external tables" section.

[DATABRICKS-CERTIFIED-PROFESSIONAL-DATA-ENGINEER VCE Dumps](#)

[DATABRICKS-CERTIFIED-PROFESSIONAL-DATA-ENGINEER Practice Test](#)

[DATABRICKS-CERTIFIED-PROFESSIONAL-DATA-ENGINEER Study Guide](#)