



VCE & PDF

GeekCert.com

<https://www.geekcert.com/databricks-certified-professional-data-engineer.html>  
2024 Latest geekcert DATABRICKS-CERTIFIED-PROFESSIONAL-DATA-ENGINEER PDF and VCE dumps Download

# DATABRICKS-CERTIFIED- PR OFESIONAL-DATA-ENGINEER<sup>Q&As</sup>

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.geekcert.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

GeekCert.com

<https://www.geekcert.com/databricks-certified-professional-data-engineer.html>  
2024 Latest geekcert 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

Which Python variable contains a list of directories to be searched when trying to locate required modules?

- A. `importlib.resource path`
- B. `sys.path`
- C. `os.path`
- D. `pypi.path`
- E. `pylib.source`

Correct Answer: B

### QUESTION 2

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 3

A junior data engineer has configured a workload that posts the following JSON to the Databricks REST API endpoint `2.0/jobs/create`.

```
{
  "name": "Ingest new data",
  "existing_cluster_id": "6015-954420-peace720",
  "notebook_task": {
    "notebook_path": "/Prod/ingest.py"
  }
}
```

Assuming that all configurations and referenced resources are available, which statement describes the result of executing this workload three times?

- A. Three new jobs named "Ingest new data" will be defined in the workspace, and they will each run once daily.
- B. The logic defined in the referenced notebook will be executed three times on new clusters with the configurations of the provided cluster ID.
- C. Three new jobs named "Ingest new data" will be defined in the workspace, but no jobs will be executed.
- D. One new job named "Ingest new data" will be defined in the workspace, but it will not be executed.
- E. The logic defined in the referenced notebook will be executed three times on the referenced existing all purpose cluster.

Correct Answer: C

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 a name, an existing cluster id, and a notebook task. However, it does not specify any schedule or trigger for the job execution. Therefore, three new jobs with the same name and configuration will be created in the workspace, but none of them will be executed until they are manually triggered or scheduled. Verified References: [Databricks Certified Data Engineer Professional], under "Monitoring and Logging" section; [Databricks Documentation], under "Jobs API - Create" section.

### QUESTION 4



The Databricks workspace administrator has configured interactive clusters for each of the data engineering groups. To control costs, clusters are set to terminate after 30 minutes of inactivity. Each user should be able to execute workloads against their assigned clusters at any time of the day.

Assuming users have been added to a workspace but not granted any permissions, which of the following describes the minimal permissions a user would need to start and attach to an already configured cluster.

- A. "Can Manage" privileges on the required cluster
- B. Workspace Admin privileges, cluster creation allowed. "Can Attach To" privileges on the required cluster
- C. Cluster creation allowed. "Can Attach To" privileges on the required cluster
- D. "Can Restart" privileges on the required cluster
- E. Cluster creation allowed. "Can Restart" privileges on the required cluster

Correct Answer: C

Explanation: This is the minimal permission a user would need to start and attach to an already configured cluster. Cluster creation allowed means that the user can create new clusters or start existing clusters that are stopped. "Can Attach To" privileges on the required cluster means that the user can attach notebooks or libraries to that cluster and run commands on it. Verified References: Databricks Certified Data Engineer Professional, under "Security and Governance" section; Databricks Documentation, under "Cluster permissions" section.

## QUESTION 5

Review the following error traceback:



```
-----  
AnalysisException                                Traceback (most recent call last)  
<command-3293767849433948> in <module>  
----> 1 display(df.select(3*"heartrate"))  
  
/databricks/spark/python/pyspark/sql/dataframe.py in select(self, *cols)  
    1690         [Row(name='Alice', age=12), Row(name='Bob', age=15)]  
    1691         """  
-> 1692         jdf = self._jdf.select(self._jcols(*cols))  
    1693         return DataFrame(jdf, self.sql_ctx)  
    1694  
  
/databricks/spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py in __call__(self, *args)  
    1302  
    1303         answer = self.gateway_client.send_command(command)  
-> 1304         return_value = get_return_value(  
    1305             answer, self.gateway_client, self.target_id, self.name)  
    1306  
  
/databricks/spark/python/pyspark/sql/utils.py in deco(*a, **kw)  
    121         # Hide where the exception came from that shows a non-Pythonic  
    122         # JVM exception message.  
--> 123         raise converted from None  
    124     else:  
    125         raise  
  
AnalysisException: cannot resolve 'heartrateheartrateheartrate' given input columns:  
[spark_catalog.database.table.device_id, spark_catalog.database.table.heartrate,  
spark_catalog.database.table.mrn, spark_catalog.database.table.time];  
'Project ['heartrateheartrateheartrate]  
+- SubqueryAlias spark_catalog.database.table  
   +- Relation[device_id#75L,heartrate#76,mrn#77L,time#78] parquet
```

Which statement describes the error being raised?

- A. The code executed was PySpark but was executed in a Scala notebook.
- B. There is no column in the table named heartrateheartrateheartrate
- C. There is a type error because a column object cannot be multiplied.
- D. There is a type error because a DataFrame object cannot be multiplied.
- E. There is a syntax error because the heartrate column is not correctly identified as a column.

Correct Answer: E

Explanation: The error is a Py4JJavaError, which means that an exception was thrown in Java code called by Python code using Py4J. Py4J is a library that enables Python programs to dynamically access Java objects in a Java Virtual Machine (JVM). PySpark uses Py4J to communicate with Spark's JVM-based engine. The error message shows that the exception was thrown by org.apache.spark.sql.AnalysisException, which means that an error occurred during the analysis phase of Spark SQL query processing. The error message also shows that the cause of the exception was "cannot resolve 'heartrateheartrateheartrate' given input columns". This means that Spark could not find a column named



heartrateheartrateheartrate in the input DataFrame or Dataset. The reason for this error is that there is a syntax error in the code that caused this exception. The code

is:

```
df.withColumn("heartrate", heartrate * 3)
```

The code tries to create a new column called heartrate by multiplying an existing column called heartrate by 3. However, the code does not correctly identify the heartrate column as a column object, but rather as a plain Python variable. This

causes PySpark to concatenate the variable name with itself three times, resulting in heartrateheartrateheartrate, which is not a valid column name. To fix this error, the code should use one of the following ways to identify the heartrate

column as a column object:

```
df.withColumn("heartrate", df["heartrate"] * 3) df.withColumn("heartrate", df.heartrate * 3) df.withColumn("heartrate", col("heartrate") * 3)
```

Verified References: [Databricks Certified Data Engineer Professional], under "Spark Core" section; Py4J Documentation, under "What is Py4J?"; Databricks Documentation, under "Query plans - Analysis phase"; Databricks Documentation, under "Accessing columns".

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

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

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