DATABRICKS-CERTIFIED-PR OFESSIONAL-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.pass2lead.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

QUESTION 1

The DevOps team has configured a production workload as a collection of notebooks scheduled to run daily using the Jobs UI. A new data engineering hire is onboarding to the team and has requested access to one of these notebooks to review the production logic.

What are the maximum notebook permissions that can be granted to the user without allowing accidental changes to production code or data?

A. Can Manage

B. Can Edit

C. No permissions

D. Can Read

E. Can Run

Correct Answer: D

Explanation: This is the correct answer because it is the maximum notebook permissions that can be granted to the user without allowing accidental changes to production code or data. Notebook permissions are used to control access to notebooks in Databricks workspaces. There are four types of notebook permissions: Can Manage, Can Edit, Can Run, and Can Read. Can Manage allows full control over the notebook, including editing, running, deleting, exporting, and changing permissions. Can Edit allows modifying and running the notebook, but not changing permissions or deleting it. Can Run allows executing commands in an existing cluster attached to the notebook, but not modifying or exporting it. Can Read allows viewing the notebook content, but not running or modifying it. In this case, granting Can Read permission to the user will allow them to review the production logic in the notebook without allowing them to makeany changes to it or run any commands that may affect production data. Verified References: [Databricks Certified Data Engineer Professional], under "Databricks Workspace" section; Databricks Documentation, under "Notebook permissions" section.

QUESTION 2

A table in the Lakehouse namedcustomer_churn_paramsis used in churn prediction by the machine learning team. The table contains information about customers derived from a number of upstream sources. Currently, the data engineering

team populates this table nightly by overwriting the table with the current valid values derived from upstream data sources.

The churn prediction model used by the ML team is fairly stable in production. The team is only interested in making predictions on records that have changed in the past 24 hours.

Which approach would simplify the identification of these changed records?

A. Apply the churn model to all rows in the customer_churn_params table, but implement logic to perform an upsert into the predictions table that ignores rows where predictions have not changed.

B. Convert the batch job to a Structured Streaming job using the complete output mode; configure a Structured Streaming job to read from the customer_churn_params table and incrementally predict against the churn model.

C. Calculate the difference between the previous model predictions and the current customer_churn_params on a key identifying unique customers before making new predictions; only make predictions on those customers not in the previous predictions.

D. Modify the overwrite logic to include a field populated by calling spark.sql.functions.current_timestamp() as data are being written; use this field to identify records written on a particular date.

E. Replace the current overwrite logic with a merge statement to modify only those records that have changed; write logic to make predictions on the changed records identified by the change data feed.

Correct Answer: B

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 an existing cluster id and a notebook task, but also specifies a new cluster spec with some

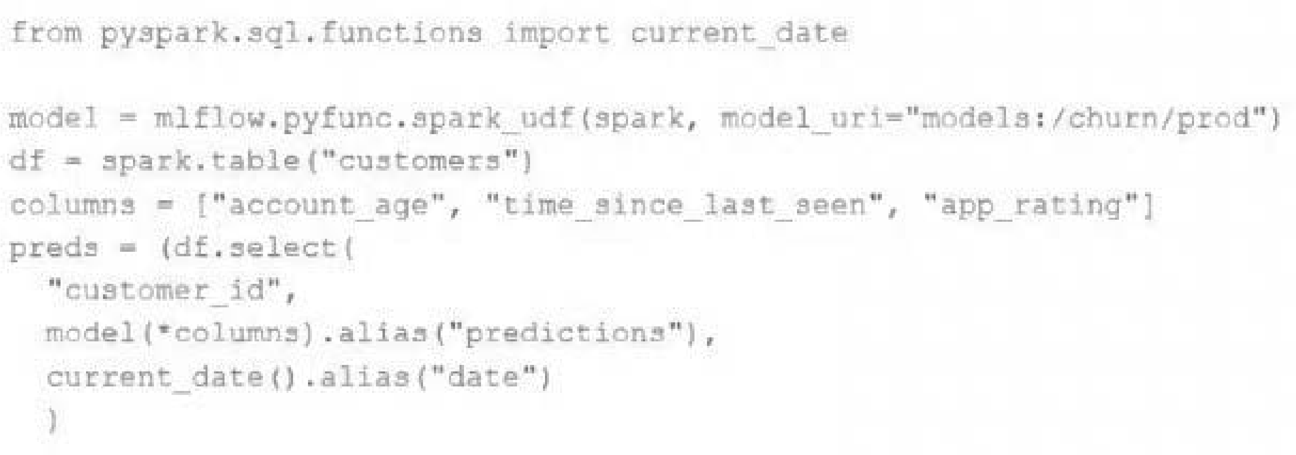
configurations. According to the documentation, if both an existing cluster id and a new cluster spec are provided, then a new cluster will be created for each run of the job with those configurations, and then terminated after completion.

Therefore, the logic defined in the referenced notebook will be executed three times on new clusters with those configurations. Verified References:

[Databricks Certified Data Engineer Professional], under "Monitoring and Logging" section; Databricks Documentation, under "JobsClusterSpecNewCluster" section.

QUESTION 3

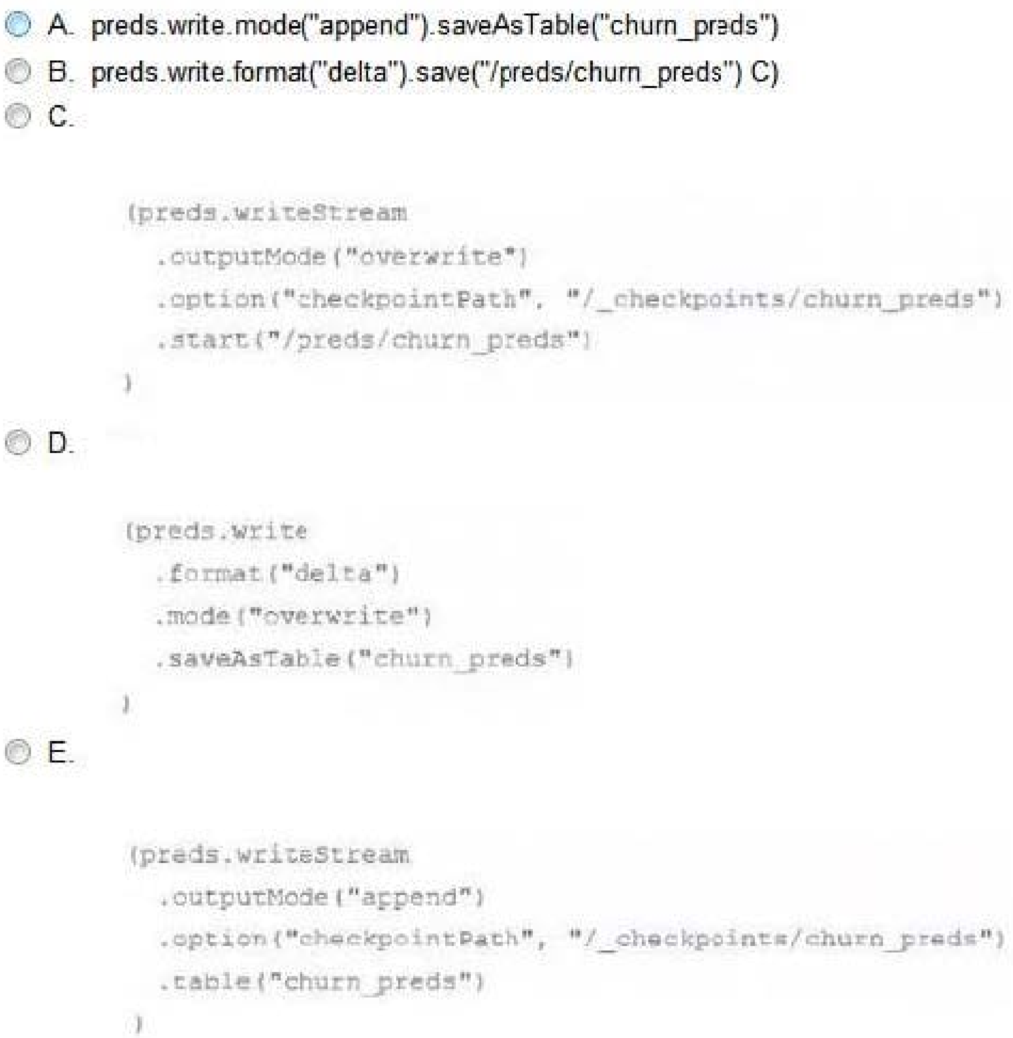
The data science team has created and logged a production model using MLflow. The following code correctly imports and applies the production model to output the predictions as a new DataFrame namedpredswith the schema "customer_id LONG, predictions DOUBLE, date DATE".



```
from pyspark.sql.functions import current_date

model = mlflow.pyfunc.spark_udf(spark, model_uri="models:/churn/prod")
df = spark.table("customers")
columns = ["account_age", "time_since_last_seen", "app_rating"]
preds = (df.select(
  "customer_id",
  model(*columns).alias("predictions"),
  current_date().alias("date")
  )
```

The data science team would like predictions saved to a Delta Lake table with the ability to compare all predictions across time. Churn predictions will be made at most once per day. Which code block accomplishes this task while minimizing potential compute costs?



○ A. preds.write.mode("append").saveAsTable("churn_preds")

○ B. preds.write.format("delta").save("/preds/churn_preds") C)

○ C.

```
(preds.writeStream
  .outputMode("overwrite")
  .option("checkpointPath", "/_checkpoints/churn_preds")
  .start("/preds/churn_preds")
)
```

○ D.

```
(preds.write
  .format("delta")
  .mode("overwrite")
  .saveAsTable("churn_preds")
)
```

○ E.

```
(preds.writeStream
  .outputMode("append")
  .option("checkpointPath", "/_checkpoints/churn_preds")
  .table("churn_preds")
)
```

A. Option

B. Option

C. Option

D. Option

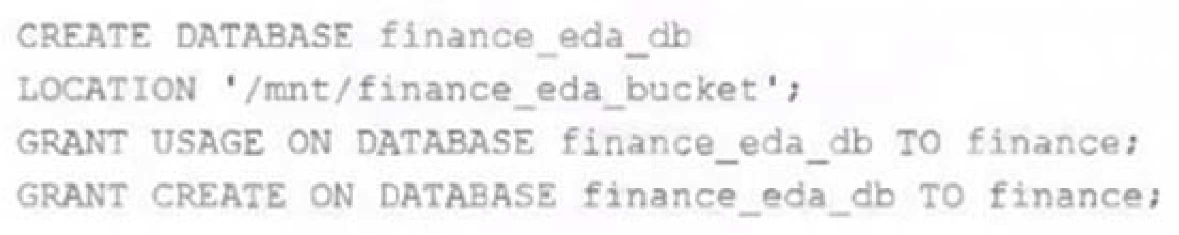
E. Option

Correct Answer: C

Explanation: This is the correct answer because it will save the predictions to a Delta Lake table with the ability to compare all predictions across time. The code uses the mergeInto method to perform an upsert operation, which means it will insert new records or update existing records based on the customer_id and date columns. This way, the table will always contain the latest predictions for each customer and date, and also keep the history of previous predictions. The code also uses a new job cluster to run the job, which will minimize the compute costs as it will be created and terminated for each run. Verified References: [Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "Upsert into a table using merge" section.

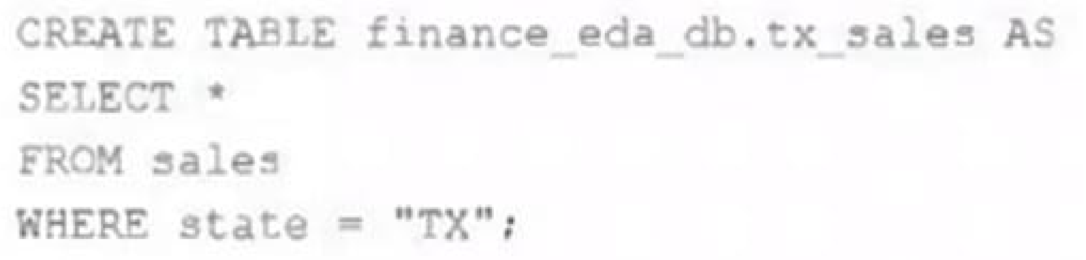
QUESTION 4

An external object storage container has been mounted to the location/mnt/finance_eda_bucket. The following logic was executed to create a database for the finance team:



```
CREATE DATABASE finance_eda_db
LOCATION '/mnt/finance_eda_bucket';
GRANT USAGE ON DATABASE finance_eda_db TO finance;
GRANT CREATE ON DATABASE finance_eda_db TO finance;
```

After the database was successfully created and permissions configured, a member of the finance team runs the following code:



```
CREATE TABLE finance_eda_db.tx_sales AS
SELECT *
FROM sales
WHERE state = "TX";
```

If all users on the finance team are members of thefinancegroup, which statement describes how thetx_salestable will be created?

A. A logical table will persist the query plan to the Hive Metastore in the Databricks control plane.

B. An external table will be created in the storage container mounted to /mnt/finance eda bucket.

C. A logical table will persist the physical plan to the Hive Metastore in the Databricks control plane.

D. An managed table will be created in the storage container mounted to /mnt/finance eda bucket.

E. A managed table will be created in the DBFS root storage container.

Correct Answer: B

Explanation: The code uses the CREATE TABLE USING DELTA command to create a Delta Lake table from an existing Parquet file stored in an external object storage container mounted to /mnt/finance_eda_bucket. The code 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 code 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. Verified References: [Databricks Certified Data Engineer Professional], under "Delta Lake" section; Databricks Documentation, under "Create an external table" section.

QUESTION 5

Which statement describes integration testing?

A. Validates interactions between subsystems of your application

B. Requires an automated testing framework

C. Requires manual intervention

D. Validates an application use case

E. Validates behavior of individual elements of your application

Correct Answer: A

Explanation: This is the correct answer because it describes integration testing. Integration testing is a type of testing that validates interactions between subsystems of your application, such as modules, components, or services. Integration testing ensures that the subsystems work together as expected and produce the correct outputs or results. Integration testing can be done at different levels of granularity, such as component integration testing, system integration testing, or end-to-end testing. Integration testing can help detect errors or bugs that may not be found by unit testing, which only validates behavior of individual elements of your application. Verified References: [Databricks Certified Data Engineer Professional], under "Testing" section; Databricks Documentation, under "Integration testing" section.