DATABRICKS-CERTIFIED-ASSOCIAT

Q&As

Databricks Certified Associate Developer for Apache Spark 3.0

SATISFACTION GUARANTEED
100%
SATISFACTION GUARANTEED

QUESTION 1

Which of the following describes characteristics of the Dataset API?

A. The Dataset API does not support unstructured data.

B. In Python, the Dataset API mainly resembles Pandas\\' DataFrame API.

C. In Python, the Dataset API\\'s schema is constructed via type hints.

D. The Dataset API is available in Scala, but it is not available in Python.

E. The Dataset API does not provide compile-time type safety.

Correct Answer: D

QUESTION 2

Which of the following describes a valid concern about partitioning?

A. A shuffle operation returns 200 partitions if not explicitly set.

B. Decreasing the number of partitions reduces the overall runtime of narrow transformations if there are more executors available than partitions.

C. No data is exchanged between executors when coalesce() is run.

D. Short partition processing times are indicative of low skew.

E. The coalesce() method should be used to increase the number of partitions.

Correct Answer: A

QUESTION 3

The code block displayed below contains an error. The code block should produce a DataFrame with color

as the only column and three rows with color values of red, blue, and green, respectively.

Find the error.

Code block:

1.spark.createDataFrame([("red",), ("blue",), ("green",)], "color")

2.Instead of calling spark.createDataFrame, just DataFrame should be called.

A. The commas in the tuples with the colors should be eliminated.

B. The colors red, blue, and green should be expressed as a simple Python list, and not a list of tuples.

C. Instead of color, a data type should be specified.

D. The "color" expression needs to be wrapped in brackets, so it reads ["color"].

Correct Answer: D

QUESTION 4

The code block displayed below contains multiple errors. The code block should remove column transactionDate from DataFrame transactionsDf and add a column transactionTimestamp in which

dates that are expressed as strings in column transactionDate of DataFrame transactionsDf are converted into unix timestamps. Find the errors.

Sample of DataFrame transactionsDf:

```
1.+-------------+---------+-----+-------+---------+----+----------------+

2.|transactionId|predError|value|storeId|productId| f| transactionDate|
3.+-------------+---------+-----+-------+---------+----+----------------+

4.| 1| 3| 4| 25| 1|null|2020-04-26 15:35|

5.| 2| 6| 7| 2| 2|null|2020-04-13 22:01|

6.| 3| 3| null| 25| 3|null|2020-04-02 10:53|

7.+-------------+---------+-----+-------+---------+----+----------------+
```

Code block:

```
1.transactionsDf = transactionsDf.drop("transactionDate")

2.transactionsDf["transactionTimestamp"] = unix_timestamp("transactionDate", "yyyy-MM- dd")
```

A. Column transactionDate should be dropped after transactionTimestamp has been written. The string indicating the date format should be adjusted. The withColumn operator should be used instead of the existing column assignment. Operator to_unixtime() should be used instead of unix_timestamp().

B. Column transactionDate should be dropped after transactionTimestamp has been written. The withColumn operator should be used instead of the existing column assignment. Column transactionDate should be wrapped in a col() operator.

C. Column transactionDate should be wrapped in a col() operator.

D. The string indicating the date format should be adjusted. The withColumnReplaced operator should be used instead of the drop and assign pattern in the code block to replace column transactionDate with the new column transactionTimestamp.

E. Column transactionDate should be dropped after transactionTimestamp has been written. The string indicating the date format should be adjusted. The withColumn operator should be used instead of the existing column assignment.

Correct Answer: E

This requires a lot of thinking to get right. For solving it, you may take advantage of the digital notepad that is provided to you during the test. You have probably seen that the code block includes multiple errors. In the test, you are usually confronted with a code block that only contains a single error. However, since you are practicing here, this challenging multi-error will make it easier for you to deal with single-error questions in the real exam. You can clearly see that column transactionDate should be dropped only after transactionTimestamp has been written. This is because to generate column transactionTimestamp, Spark needs to read the values from column transactionDate. Values in column transactionDate in the original transactionsDf DataFrame look like 2020- 04-26 15:35. So, to convert those correctly, you would have to pass yyyy-MM-dd HH:mm. In other words: The string indicating the date format should be adjusted. While you might be tempted to change unix_timestamp() to to_unixtime() (in line with the from_unixtime() operator), this function does not exist in Spark. unix_timestamp() is the correct operator to use here. Also, there is no DataFrame.withColumnReplaced() operator. A similar operator that exists is DataFrame.withColumnRenamed(). Whether you use col() or not is irrelevant with unix_timestamp() - the command is fine with both. Finally, you cannot assign a column like transactionsDf["columnName"] = ... in Spark. This is Pandas syntax (Pandas is a popular Python package for data analysis), but it is not supported in Spark. So, you need to use Spark\\'s DataFrame.withColumn() syntax instead. More info: pyspark.sql.functions.unix_timestamp -- PySpark 3.1.2 documentation Static notebook | Dynamic notebook: See test 3, 28 (Databricks import instructions)

QUESTION 5

Which is the highest level in Spark\\'s execution hierarchy?

A. Task

B. Executor

C. Slot

D. Job

E. Stage

Correct Answer: D