

DATABRICKS-CERTIFIED-ASSOCIATE-DEVELOPER-FOR-APACHE-SPARK

Q&As

Databricks Certified Associate Developer for Apache Spark 3.0

Pass Databricks DATABRICKS-CERTIFIED-ASSOCIATE-DEVELOPER-FOR-APACHE-SPARK Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass2lead.com/databricks-certified-associate-developer-for-apache-spark.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Databricks Official Exam Center

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



QUESTION 1

Which of the following code blocks reduces a DataFrame from 12 to 6 partitions and performs a full shuffle?

- A. `DataFrame.repartition(12)`
- B. `DataFrame.coalesce(6).shuffle()`
- C. `DataFrame.coalesce(6)`
- D. `DataFrame.coalesce(6, shuffle=True)`
- E. `DataFrame.repartition(6)`

Correct Answer: E

QUESTION 2

Which of the following code blocks returns DataFrame transactionsDf sorted in descending order by column predError, showing missing values last?

- A. `transactionsDf.sort(asc_nulls_last("predError"))`
- B. `transactionsDf.orderBy("predError").desc_nulls_last()`
- C. `transactionsDf.sort("predError", ascending=False)`
- D. `transactionsDf.desc_nulls_last("predError")`
- E. `transactionsDf.orderBy("predError").asc_nulls_last()`

Correct Answer: C

`transactionsDf.sort("predError", ascending=False)` Correct! When using `DataFrame.sort()` and setting `ascending=False`, the DataFrame will be sorted by the specified column in descending order, putting all missing values last. An alternative, although not listed as an answer here, would be `transactionsDf.sort(desc_nulls_last("predError"))`.
`transactionsDf.sort(asc_nulls_last("predError"))` Incorrect. While this is valid syntax, the DataFrame will be sorted on column `predError` in ascending order and not in descending order, putting missing values last.
`transactionsDf.desc_nulls_last("predError")` Wrong, this is invalid syntax. There is no method `DataFrame.desc_nulls_last()` in the Spark API. There is a Spark function `desc_nulls_last()` however (link see below).
`transactionsDf.orderBy("predError").desc_nulls_last()` No. While `transactionsDf.orderBy("predError")` is correct syntax (although it sorts the DataFrame by column `predError` in ascending order) and returns a DataFrame, there is no method `DataFrame.desc_nulls_last()` in the Spark API. There is a Spark function `desc_nulls_last()` however (link see below).
`transactionsDf.orderBy("predError").asc_nulls_last()` Incorrect. There is no method `DataFrame.asc_nulls_last()` in the Spark API (see above). More info: `pyspark.sql.functions.desc_nulls_last` -- PySpark 3.1.2 documentation and `pyspark.sql.DataFrame.sort` -- PySpark 3.1.2 documentation ([https:// bit.ly/3g1Jtbl](https://bit.ly/3g1Jtbl) , <https://bit.ly/2R90NCS>) Static notebook | Dynamic notebook: See test 1, 32 (Databricks import instructions) (https://flrs.github.io/spark_practice_tests_code/#1/32.html , https://bit.ly/sparkpracticeexams_import_instructions)

QUESTION 3

Which of the following code blocks reads in parquet file /FileStore/imports.parquet as a DataFrame?

- A. `spark.mode("parquet").read("/FileStore/imports.parquet")`
- B. `spark.read.path("/FileStore/imports.parquet", source="parquet")`
- C. `spark.read().parquet("/FileStore/imports.parquet")`
- D. `spark.read.parquet("/FileStore/imports.parquet")`
- E. `spark.read().format("\\parquet\\").open("/FileStore/imports.parquet")`

Correct Answer: D

QUESTION 4

The code block displayed below contains an error. The code block is intended to join DataFrame itemsDf with the larger DataFrame transactionsDf on column itemId. Find the error.

Code block:

```
transactionsDf.join(itemsDf, "itemId", how="broadcast")
```

- A. The syntax is wrong, how= should be removed from the code block.
- B. The join method should be replaced by the broadcast method.
- C. Spark will only perform the broadcast operation if this behavior has been enabled on the Spark cluster.
- D. The larger DataFrame transactionsDf is being broadcasted, rather than the smaller DataFrame itemsDf.
- E. broadcast is not a valid join type.

Correct Answer: E

QUESTION 5

Which of the following code blocks prints out in how many rows the expression Inc. appears in the stringtype column supplier of DataFrame itemsDf?

- A. `1.counter = 0`
- 2.
- 3. `for index, row in itemsDf.iterrows():`
- 4.

if '\\Inc.\\' in row['\\supplier\\']:

5.

counter = counter + 1

6.

7.print(counter)

B. 1.counter = 0

2.

3.def count(x):

4.

if '\\Inc.\\' in x['\\supplier\\']:

5.

counter = counter + 1

6.

7.itemsDf.foreach(count)

8.print(counter)

C. print(itemsDf.foreach(lambda x: '\\Inc.\\' in x))

D. print(itemsDf.foreach(lambda x: '\\Inc.\\' in x).sum())

E. 1.accum=sc.accumulator(0)

2.

3.def check_if_inc_in_supplier(row):

4.

if '\\Inc.\\' in row['\\supplier\\']:

5.

accum.add(1)

6.

7.itemsDf.foreach(check_if_inc_in_supplier)

8.print(accum.value)

Correct Answer: E

Correct code block:

```
accum=sc.accumulator(0)

def check_if_inc_in_supplier(row):

if '\\Inc.\\' in row[\\supplier\\]:

accum.add(1)

itemsDf.foreach(check_if_inc_in_supplier)

print(accum.value)
```

To answer this correctly, you need to know both about the DataFrame.foreach() method and accumulators.

When Spark runs the code, it executes it on the executors. The executors do not have any information about variables outside of their scope. This is why simply using a Python variable counter, like in the two examples that start with counter = 0, will not work. You need to tell the executors explicitly that counter is a special shared variable, an Accumulator, which is managed by the driver and can be accessed by all executors for the purpose of adding to it. If you have used Pandas in the past, you might be familiar with the iterrows() command.

Notice that there is no such command in PySpark.

The two examples that start with print do not work, since DataFrame.foreach() does not have a return value.

More info: [pyspark.sql.DataFrame.foreach -- PySpark 3.1.2 documentation](#)

Static notebook | Dynamic notebook: See test 3, 22 (Databricks import instructions)

[Latest DATABRICKS-CERTIFIED-ASSOCIATE-DEVELOPER-FOR-APACHE-SPARK Dumps](#)

[DATABRICKS-CERTIFIED-ASSOCIATE-DEVELOPER-FOR-APACHE-SPARK VCE Dumps](#)

[DATABRICKS-CERTIFIED-ASSOCIATE-DEVELOPER-FOR-APACHE-SPARK Exam Questions](#)