DATABRICKS-CERTIFIED-ASSOCIAT

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.leads4pass.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

QUESTION 1

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

A. Task

B. Executor

C. Slot

D. Job

E. Stage

Correct Answer: D

QUESTION 2

Which of the following code blocks sorts DataFrame transactionsDf both by column storeId in ascending and by column productId in descending order, in this priority?

A. transactionsDf.sort("storeId", asc("productId"))

B. transactionsDf.sort(col(storeId)).desc(col(productId))

C. transactionsDf.order_by(col(storeId), desc(col(productId)))

D. transactionsDf.sort("storeId", desc("productId"))

E. transactionsDf.sort("storeId").sort(desc("productId"))

Correct Answer: D

In this it is important to realize that you are asked to sort transactionDf by two columns. This means that the sorting of the second column depends on the sorting of the first column. So, any option that sorts the entire DataFrame (through chaining sort statements) will not work. The two columns need to be channeled through the same call to sort(). Also, order_by is not a valid DataFrame API method. More info: pyspark.sql.DataFrame.sort -- PySpark 3.1.2 documentation Static notebook | Dynamic notebook: See test 2, 22 (Databricks import instructions)

QUESTION 3

Which of the following code blocks returns all unique values across all values in columns value and productId in DataFrame transactionsDf in a one-column DataFrame?

A. tranactionsDf.select(\\'value\\').join(transactionsDf.select(\\'productId\\'), col(\\'value\\')==col(\\'productId\\'), \\'outer\\')

B. transactionsDf.select(col(\\'value\\'), col(\\'productId\\')).agg({\\'*\\': \\'count\\'})

C. transactionsDf.select(\\'value\\', \\'productId\\').distinct()

D. transactionsDf.select(\\'value\\').union(transactionsDf.select(\\'productId\\')).distinct()

E. transactionsDf.agg({\\'value\\': \\'collect_set\\', \\'productId\\': \\'collect_set\\'})

Correct Answer: D

QUESTION 4

In which order should the code blocks shown below be run in order to create a DataFrame that shows the mean of column predError of DataFrame transactionsDf per column storeId and productId, where productId should be either 2 or 3 and the returned DataFrame should be sorted in ascending order by column storeId, leaving out any nulls in that column?

DataFrame transactionsDf:

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

2.|transactionId|predError|value|storeId|productId| f|

3.+-------------+---------+-----+-------+---------+----+

4.| 1| 3| 4| 25| 1|null|

5.| 2| 6| 7| 2| 2|null|

6.| 3| 3| null| 25| 3|null|

7.| 4| null| null| 3| 2|null|

8.| 5| null| null| null| 2|null|

9.| 6| 3| 2| 25| 2|null|

10.+-------------+---------+-----+-------+---------+----+

1.

.mean("predError")

2.

.groupBy("storeId")

3.

.orderBy("storeId")

4.

transactionsDf.filter(transactionsDf.storeId.isNotNull())

5.

.pivot("productId", [2, 3])

A. 4, 5, 2, 3, 1

B. 4, 2, 1

C. 4, 1, 5, 2, 3

D. 4, 2, 5, 1, 3

E. 4, 3, 2, 5, 1

Correct Answer: D

Correct code block:

transactionsDf.filter(transactionsDf.storeId.isNotNull()).groupBy("storeId").pivot("productId", [2, 3]).mean

("predError").orderBy("storeId")

Output of correct code block:

+-------+----+----+

|storeId| 2| 3|

+-------+----+----+

| 2| 6.0|null|

| 3|null|null|

| 25| 3.0| 3.0|

+-------+----+----+

This is quite convoluted and requires you to think hard about the correct order of operations. The pivot

method also makes an appearance - a method that you may not know all that much about (yet).

At the first position in all answers is code block 4, so the is essentially just about the ordering of the

remaining 4 code blocks. The states that the returned DataFrame should be sorted by column storeId. So,

it should make sense to have code block 3 which includes the orderBy operator at the very end of the code

block. This leaves you with only two answer options. Now, it is useful to know more about the context of

pivot in PySpark. A common pattern is groupBy, pivot, and then another aggregating function, like mean.

In the documentation linked below you can see that pivot is a method of pyspark.sql.GroupedData meaning that before pivoting, you have to use groupBy. The only answer option matching this requirement

is the one in which code block 2 (which includes groupBy) is stated before code block 5 (which includes pivot).

More info: pyspark.sql.GroupedData.pivot -- PySpark 3.1.2 documentation

Static notebook | Dynamic notebook: See test 3, 43 (Databricks import instructions)

QUESTION 5

Which of the following statements about lazy evaluation is incorrect?

A. Predicate pushdown is a feature resulting from lazy evaluation.

B. Execution is triggered by transformations.

C. Spark will fail a job only during execution, but not during definition.

D. Accumulators do not change the lazy evaluation model of Spark.

E. Lineages allow Spark to coalesce transformations into stages

Correct Answer: B