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.geekcert.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 of the following describes properties of a shuffle?

A. Operations involving shuffles are never evaluated lazily.

B. Shuffles involve only single partitions.

C. Shuffles belong to a class known as "full transformations".

D. A shuffle is one of many actions in Spark.

E. In a shuffle, Spark writes data to disk.

Correct Answer: E

In a shuffle, Spark writes data to disk.

Correct! Spark\\'s architecture dictates that intermediate results during a shuffle are written to disk.

A shuffle is one of many actions in Spark.

Incorrect. A shuffle is a transformation, but not an action.

Shuffles involve only single partitions.

No, shuffles involve multiple partitions. During a shuffle, Spark generates output partitions from multiple

input partitions.

Operations involving shuffles are never evaluated lazily. Wrong. A shuffle is a costly operation and Spark

will evaluate it as lazily as other transformations. This is, until a subsequent action triggers its evaluation.

Shuffles belong to a class known as "full transformations". Not quite. Shuffles belong to a class known as

"wide transformations". "Full transformation" is not a relevant term in Spark.

More info: Spark ?The Definitive Guide, Chapter 2 and Spark: disk I/O on stage boundaries explanation Stack Overflow

QUESTION 2

The code block shown below should return a two-column DataFrame with columns transactionId and supplier, with combined information from DataFrames itemsDf and transactionsDf. The code block should merge rows in which column productId of DataFrame transactionsDf matches the value of column itemId in DataFrame itemsDf, but only where column storeId of DataFrame

transactionsDf does not match column itemId of DataFrame itemsDf. Choose the answer that correctly fills the blanks in the code block to accomplish this.

Code block:

transactionsDf.__1__(itemsDf, __2__).__3__(__4__)

A. 1. join

2.

transactionsDf.productId==itemsDf.itemId, how="inner"

3.

select

4.

"transactionId", "supplier"

B. 1. select

2.

"transactionId", "supplier"

3.

join

4.

[transactionsDf.storeId!=itemsDf.itemId, transactionsDf.productId==itemsDf.itemId]

C. 1. join

2.

[transactionsDf.productId==itemsDf.itemId, transactionsDf.storeId!=itemsDf.itemId]

3.

select

4.

"transactionId", "supplier"

D. 1. filter

2.

"transactionId", "supplier"

3.

join

4.

"transactionsDf.storeId!=itemsDf.itemId, transactionsDf.productId==itemsDf.itemId"

E. 1. join

2.

transactionsDf.productId==itemsDf.itemId, transactionsDf.storeId!=itemsDf.itemId

3.

filter

4.

"transactionId", "supplier"

Correct Answer: C

This is pretty complex and, in its complexity, is probably above what you would encounter in the exam. However, reading the carefully, you can use your logic skills to weed out the wrong answers here. First, you should examine the join statement which is common to all answers. The first argument of the join () operator (documentation linked below) is the DataFrame to be joined with. Where join is in gap 3, the first argument of gap 4 should therefore be another DataFrame. For none of the questions where join is in the third gap, this is the case. So you can immediately discard two answers. For all other answers, join is in gap 1, followed by .(itemsDf, according to the code block. Given how the join() operator is called, there are now three remaining candidates. Looking further at the join() statement, the second argument (on=) expects "a string for the join column name, a list of column names, a join expression (Column), or a list of Columns", according to the documentation. As one answer option includes a list of join expressions (transactionsDf.productId==itemsDf.itemId, transactionsDf.storeId!=itemsDf.itemId) which is unsupported according to the documentation, we can discard that answer, leaving us with two remaining candidates. Both candidates have valid syntax, but only one of them fulfills the condition in the "only where column storeId of DataFrame transactionsDf does not match column itemId of DataFrame itemsDf". So, this one remaining answer option has to be the correct one! As you can see, although sometimes overwhelming at first, even more complex questions can be figured out by rigorously applying the knowledge you can gain from the documentation during the exam.

More info: pyspark.sql.DataFrame.join -- PySpark 3.1.2 documentation

Static notebook | Dynamic notebook: See test 3, 47 (Databricks import instructions)

QUESTION 3

The code block displayed below contains an error. The code block should return DataFrame transactionsDf, but with the column storeId renamed to storeNumber. Find the error.

Code block:

transactionsDf.withColumn("storeNumber", "storeId")

A. Instead of withColumn, the withColumnRenamed method should be used.

B. Arguments "storeNumber" and "storeId" each need to be wrapped in a col() operator.

C. Argument "storeId" should be the first and argument "storeNumber" should be the second argument to the withColumn method.

D. The withColumn operator should be replaced with the copyDataFrame operator.

E. Instead of withColumn, the withColumnRenamed method should be used and argument "storeId" should be the first

and argument "storeNumber" should be the second argument to that method.

Correct Answer: E

Correct code block:

transactionsDf.withColumnRenamed("storeId", "storeNumber") More info:

pyspark.sql.DataFrame.withColumnRenamed -- PySpark 3.1.1 documentation Static notebook | Dynamic

notebook: See test 1, 38 (Databricks import instructions)

QUESTION 4

Which of the following code blocks returns a single-row DataFrame that only has a column corr which shows the Pearson correlation coefficient between columns predError and value in DataFrame transactionsDf?

A. transactionsDf.select(corr(["predError", "value"]).alias("corr")).first()

B. transactionsDf.select(corr(col("predError"), col("value")).alias("corr")).first()

C. transactionsDf.select(corr(predError, value).alias("corr"))

D. transactionsDf.select(corr(col("predError"), col("value")).alias("corr"))

E. transactionsDf.select(corr("predError", "value"))

Correct Answer: D

In difficulty, this is above what you can expect from the exam. What this wants to teach you, however, is

to pay attention to the useful details included in the

documentation.

pyspark.sql.corr is not a very common method, but it deals with Spark\\'s data structure in an interesting

way. The command takes two columns over multiple rows and returns a single row - similar to

an aggregation function. When examining the documentation (linked below), you will find this code

example:

a = range(20)

b = [2 * x for x in range(20)]

df = spark.createDataFrame(zip(a, b), ["a", "b"])

df.agg(corr("a", "b").alias(\\'c\\')).collect()

[Row(c=1.0)]

See how corr just returns a single row? Once you understand this, you should be suspicious about answers that include first(), since there is no need to just select a single row. A reason to eliminate those answers is that DataFrame.first() returns an object of type Row, but not DataFrame, as requested in the question.

transactionsDf.select(corr(col("predError"), col("value")).alias("corr")) Correct! After calculating the Pearson correlation coefficient, the resulting column is correctly renamed to corr.

transactionsDf.select(corr(predError, value).alias("corr")) No. In this answer, Python will interpret column names predError and value as variable names.

transactionsDf.select(corr(col("predError"), col("value")).alias("corr")).first() Incorrect. first() returns a row, not a DataFrame (see above and linked documentation below).

transactionsDf.select(corr("predError", "value"))

Wrong. Whie this statement returns a DataFrame in the desired shape, the column will have the name corr (predError, value) and not corr.

transactionsDf.select(corr(["predError", "value"]).alias("corr")).first() False. In addition to first() returning a row, this code block also uses the wrong call structure for command corr which takes two arguments (the two columns to correlate).

More info:

-pyspark.sql.functions.corr -- PySpark 3.1.2 documentation

-pyspark.sql.DataFrame.first -- PySpark 3.1.2 documentation

Static notebook | Dynamic notebook: See test 3, 53 (Databricks import instructions)

QUESTION 5

The code block displayed below contains an error. The code block should save DataFrame transactionsDf at path path as a parquet file, appending to any existing parquet file. Find the error.

Code block:

A. transactionsDf.format("parquet").option("mode", "append").save(path)

B. The code block is missing a reference to the DataFrameWriter.

C. save() is evaluated lazily and needs to be followed by an action.

D. The mode option should be omitted so that the command uses the default mode.

E. The code block is missing a bucketBy command that takes care of partitions.

F. Given that the DataFrame should be saved as parquet file, path is being passed to the wrong method.

Correct Answer: B

Correct code block:

transactionsDf.write.format("parquet").option("mode", "append").save(path)