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.passapply.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

The code block displayed below contains one or more errors. The code block should load parquet files at location filePath into a DataFrame, only loading those files that have been modified before 2029-03-20

05:44:46. Spark should enforce a schema according to the schema shown below. Find the error.

Schema:

1.root

2.

|-- itemId: integer (nullable = true)

3.

|-- attributes: array (nullable = true)

4.

| |-- element: string (containsNull = true)

5.

|-- supplier: string (nullable = true)

Code block:

1.schema = StructType([

2.

StructType("itemId", IntegerType(), True),

3.

StructType("attributes", ArrayType(StringType(), True), True),

4.

StructType("supplier", StringType(), True)

5.])

6.

7.spark.read.options("modifiedBefore", "2029-03-20T05:44:46").schema(schema).load(filePath)

A. The attributes array is specified incorrectly, Spark cannot identify the file format, and the syntax of the call to Spark\\'s DataFrameReader is incorrect.

B. Columns in the schema definition use the wrong object type and the syntax of the call to Spark\\'s DataFrameReader is incorrect.

C. The data type of the schema is incompatible with the schema() operator and the modification date threshold is specified incorrectly.

D. Columns in the schema definition use the wrong object type, the modification date threshold is specified incorrectly, and Spark cannot identify the file format.

E. Columns in the schema are unable to handle empty values and the modification date threshold is specified incorrectly.

Correct Answer: D

QUESTION 2

Which of the following code blocks reorders the values inside the arrays in column attributes of DataFrame

itemsDf from last to first one in the alphabet?

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

2.|itemId|attributes |supplier |

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

4.|1 |[blue, winter, cozy] |Sports Company Inc.|

5.|2 |[red, summer, fresh, cooling]|YetiX |

6.|3 |[green, summer, travel] |Sports Company Inc.|

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

A. itemsDf.withColumn(\\'attributes\\', sort_array(col(\\'attributes\\').desc()))

B. itemsDf.withColumn(\\'attributes\\', sort_array(desc(\\'attributes\\')))

C. itemsDf.withColumn(\\'attributes\\', sort(col(\\'attributes\\'), asc=False))

D. itemsDf.withColumn("attributes", sort_array("attributes", asc=False))

E. itemsDf.select(sort_array("attributes"))

Correct Answer: D

QUESTION 3

Which of the following code blocks produces the following output, given DataFrame transactionsDf?

Output:

1.root

2.

|-- transactionId: integer (nullable = true)

3.

|-- predError: integer (nullable = true)

4.

|-- value: integer (nullable = true)

5.

|-- storeId: integer (nullable = true)

6.

|-- productId: integer (nullable = true)

7.

|-- f: integer (nullable = true)

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.+-------------+---------+-----+-------+---------+----+

A. transactionsDf.schema.print()

B. transactionsDf.rdd.printSchema()

C. transactionsDf.rdd.formatSchema()

D. transactionsDf.printSchema()

E. print(transactionsDf.schema)

Correct Answer: D

The output is the typical output of a DataFrame.printSchema() call. The DataFrame\\'s RDD representation does not

have a printSchema or formatSchema method (find available methods in the RDD documentation linked below). The output of print(transactionsDf.schema) is this:
StructType(List(StructField(transactionId,IntegerType,true),StructField(predError,IntegerTy pe,true),StructField (value,IntegerType,true),StructField(storeId,IntegerType,true),StructField(productId,Integer Type,true),StructField(f,IntegerType,true))). It includes the same information as the nicely formatted original output, but is not nicely formatted itself. Lastly, the DataFrame\\'s schema attribute does not have a print() method.

More info:

-pyspark.RDD: pyspark.RDD -- PySpark 3.1.2 documentation

-DataFrame.printSchema(): pyspark.sql.DataFrame.printSchema -- PySpark 3.1.2 documentation

Static notebook | Dynamic notebook: See test 2, 52 (Databricks import instructions)

QUESTION 4

Which of the following code blocks reads the parquet file stored at filePath into DataFrame itemsDf, using a valid schema for the sample of itemsDf shown below?

Sample of itemsDf:

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

2.|itemId|attributes |supplier |

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

4.|1 |[blue, winter, cozy] |Sports Company Inc.|

5.|2 |[red, summer, fresh, cooling]|YetiX |

6.|3 |[green, summer, travel] |Sports Company Inc.|

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

A. 1.itemsDfSchema = StructType([

2.

StructField("itemId", IntegerType()),

3.

StructField("attributes", StringType()),

4.

StructField("supplier", StringType())])

5.

6.itemsDf = spark.read.schema(itemsDfSchema).parquet(filePath)

B. 1.itemsDfSchema = StructType([

2.

StructField("itemId", IntegerType),

3.

StructField("attributes", ArrayType(StringType)),

4.

StructField("supplier", StringType)])

5.

6.itemsDf = spark.read.schema(itemsDfSchema).parquet(filePath)

C. 1.itemsDf = spark.read.schema(\\'itemId integer, attributes , supplier string\\').parquet(filePath)

D. 1.itemsDfSchema = StructType([

2.

StructField("itemId", IntegerType()),

3.

StructField("attributes", ArrayType(StringType())),

4.

StructField("supplier", StringType())])

5.

6.itemsDf = spark.read.schema(itemsDfSchema).parquet(filePath)

E. 1.itemsDfSchema = StructType([

2.

StructField("itemId", IntegerType()),

3.

StructField("attributes", ArrayType([StringType()])),

4.

StructField("supplier", StringType())])

5.

6.itemsDf = spark.read(schema=itemsDfSchema).parquet(filePath)

Correct Answer: D

The challenge in this comes from there being an array variable in the schema. In addition, you should know how to pass a schema to the DataFrameReader that is invoked by spark.read. The correct way to define an array of strings in a schema is through ArrayType(StringType()). A schema can be passed to the DataFrameReader by simply appending schema(structType) to the read() operator. Alternatively, you can also define a schema as a string. For example, for the schema of itemsDf, the following string would make sense: itemId integer, attributes array, supplier string. A thing to keep in mind is that in schema definitions, you always need to instantiate the types, like so: StringType(). Just using StringType does not work in pySpark and will fail. Another concern with schemas is whether columns should be nullable, so allowed to have null values. In the case at hand, this is not a concern however, since the just asks for a "valid" schema. Both non-nullable and nullable column schemas would be valid here, since no null value appears in the DataFrame sample. More info: Learning Spark, 2nd Edition, Chapter 3 Static notebook | Dynamic notebook: See test 3, 19 (Databricks import instructions)

QUESTION 5

The code block shown below should show information about the data type that column storeId of DataFrame transactionsDf contains. Choose the answer that correctly fills the blanks in the code block to accomplish this.

Code block:

transactionsDf.__1__(__2__).__3__

A. 1. select

2.

"storeId"

3.

print_schema()

B. 1. limit

2.

1

3.

columns

C. 1. select

2.

"storeId"

3.

printSchema()

D. 1. limit

2.

"storeId"

3.

printSchema()

E. 1. select

2.

storeId

3.

dtypes

Correct Answer: B

Correct code block: transactionsDf.select("storeId").printSchema() The difficulty of this is that it is hard to solve with the stepwise first-to-last- gap approach that has worked well for similar questions, since the answer options are so different from one another. Instead, you might want to eliminate answers by looking for patterns of frequently wrong answers. A first pattern that you may recognize by now is that column names are not expressed in quotes. For this reason, the answer that includes storeId should be eliminated. By now, you may have understood that the DataFrame.limit() is useful for returning a specified amount of rows. It has nothing to do with specific columns. For this reason, the answer that resolves to

limit("storeId") can be eliminated. Given that we are interested in information about the data type, you should