



Vendor: Databricks

Exam Code: Databricks-Certified-Associate-Developer-for-Apache-Spark

Exam Name: Databricks Certified Associate Developer for Apache Spark

Version: DEMO

QUESTION 1

Which of the following code blocks returns a DataFrame containing only the rows from DataFrame storesDF where the value in column sqft is less than or equal to 25,000?

- A. storesDF.filter("sqft" <= 25000)
- B. storesDF.filter(sqft > 25000)
- C. storesDF.where(storesDF[sqft] > 25000)
- D. storesDF.where(sqft > 25000)
- E. storesDF.filter(col("sqft") <= 25000)

Answer: E

QUESTION 2

Which of the following code blocks returns a DataFrame containing only the rows from DataFrame storesDF where the value in column sqft is less than or equal to 25,000 OR the value in column customerSatisfaction is greater than or equal to 30?

- A. storesDF.filter(col("sqft") <= 25000 | col("customerSatisfaction") >= 30)
- B. storesDF.filter(col("sqft") <= 25000 or col("customerSatisfaction") >= 30)
- C. storesDF.filter(sqft <= 25000 or customerSatisfaction >= 30)
- D. storesDF.filter(col(sqft) <= 25000 | col(customerSatisfaction) >= 30)
- E. storesDF.filter((col("sqft") <= 25000) | (col("customerSatisfaction") >= 30))

Answer: E

QUESTION 3

Which of the following code blocks returns a new DataFrame from DataFrame storesDF where column storeId is of the type string?

- A. storesDF.withColumn("storeId", cast(col("storeId"), StringType()))
- B. storesDF.withColumn("storeId", col("storeId").cast(StringType()))
- C. storesDF.withColumn("storeId", cast(storeId).as(StringType))
- D. storesDF.withColumn("storeId", col(storeId).cast(StringType))
- E. storesDF.withColumn("storeId", cast("storeId").as(StringType()))

Answer: B

QUESTION 4

Which of the following code blocks returns a new DataFrame with a new column employeesPerSqft that is the quotient of column numberOfEmployees and column sqft, both of which are from DataFrame storesDF? Note that column employeesPerSqft is not in the original DataFrame storesDF.

- A. storesDF.withColumn("employeesPerSqft", col("numberOfEmployees") / col("sqft"))
- B. storesDF.withColumn("employeesPerSqft", "numberOfEmployees" / "sqft")
- C. storesDF.select("employeesPerSqft", "numberOfEmployees" / "sqft")
- D. storesDF.select("employeesPerSqft", col("numberOfEmployees") / col("sqft"))
- E. storesDF.withColumn(col("employeesPerSqft"), col("numberOfEmployees") / col("sqft"))

Answer: A

QUESTION 5

The code block shown below should return a new DataFrame from DataFrame storesDF where column modality is the constant string "PHYSICAL", Assume DataFrame storesDF is the only defined language variable. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

```
storesDF. _1_( _2_, _3_( _4_ ) )
```

- A. 1. withColumn
2. "modality"
3. col
4. "PHYSICAL"
- B. 1. withColumn
2. "modality"
3. lit
4. PHYSICAL
- C. 1. withColumn
2. "modality"
3. lit
4. "PHYSICAL"
- D. 1. withColumn
2. "modality"
3. SrtringType
4. "PHYSICAL"
- E. 1. newColumn
2. modality
3. SrtringType
4. PHYSICAL

Answer: C

QUESTION 6

Which of the following code blocks returns a DataFrame where column storeCategory from DataFrame storesDF is split at the underscore character into column storeValueCategory and column storeSizeCategory?

A sample of DataFrame storesDF is displayed below:

storeId	open	openDate	storeCategory
0	true	1100746394	VALUE_MEDIUM
1	true	944572255	MAINSTREAM_SMALL
2	false	925495628	PREMIUM_LARGE
3	true	1397353092	VALUE_MEDIUM
4	true	986505057	VALUE_LARGE
5	true	955988614	PREMIUM_LARGE
...	...	...	...

- A. (storesDF.withColumn("storeValueCategory", split(col("storeCategory"), "_")[1])
.withColumn("storeSizeCategory", split(col("storeCategory"), "_")[2]))
- B. (storesDF.withColumn("storeValueCategory", col("storeCategory").split("_")[0])
.withColumn("storeSizeCategory", col("storeCategory").split("_")[1]))
- C. (storesDF.withColumn("storeValueCategory", split(col("storeCategory"), "_")[0])
.withColumn("storeSizeCategory", split(col("storeCategory"), "_")[1]))
- D. (storesDF.withColumn("storeValueCategory", split("storeCategory", "_")[0])
.withColumn("storeSizeCategory", split("storeCategory", "_")[1]))
- E. (storesDF.withColumn("storeValueCategory", col("storeCategory").split("_")[1])
.withColumn("storeSizeCategory", col("storeCategory").split("_")[2]))

Answer: C

QUESTION 7

Which of the following code blocks returns a new DataFrame where column productCategories only has one word per row, resulting in a DataFrame with many more rows than DataFrame storesDF?

A sample of storesDF is displayed below:

storeId	productCategories
0	[netus, pellentes...
1	[consequat enim,...
2	[massa, a, vitae,...
3	[aliquam, donec,...
4	[condimentum, fer...
5	[viverra habitan...
...	...

- A. storesDF.withColumn("productCategories", explode(col("productCategories")))
- B. storesDF.withColumn("productCategories", split(col("productCategories")))
- C. storesDF.withColumn("productCategories", col("productCategories").explode())

- D. `storesDF.withColumn("productCategories", col("productCategories").split())`
- E. `storesDF.withColumn("productCategories", explode("productCategories"))`

Answer: A

QUESTION 8

Which of the following code blocks returns a new DataFrame with column `storeDescription` where the pattern "Description: " has been removed from the beginning of column `storeDescription` in DataFrame `storesDF`?

A sample of DataFrame `storesDF` is below:

storeId	storeDescription
0	Description: ultr...
1	Description: sagi...
2	Description: port...
3	Description: tris...
4	Description: ulla...
...	

- A. `storesDF.withColumn("storeDescription", regexp_replace(col("storeDescription"), "^Description: "))`
- B. `storesDF.withColumn("storeDescription", col("storeDescription").regexp_replace("^Description: ", ""))`
- C. `storesDF.withColumn("storeDescription", regexp_extract(col("storeDescription"), "^Description: ", ""))`
- D. `storesDF.withColumn("storeDescription", regexp_replace("storeDescription", "^Description: ", ""))`
- E. `storesDF.withColumn("storeDescription", regexp_replace(col("storeDescription"), "^Description: ", ""))`

Answer: E

QUESTION 9

Which of the following code blocks returns a new DataFrame where column `division` from DataFrame `storesDF` has been replaced and renamed to column `state` and column `managerName` from DataFrame `storesDF` has been replaced and renamed to column `managerFullName`?

- A. `(storesDF.withColumnRenamed(["division", "state"], ["managerName", "managerFullName"]))`
- B. `(storesDF.withColumn("state", col("division")).withColumn("managerFullName", col("managerName")))`
- C. `(storesDF.withColumn("state", "division").withColumn("managerFullName", "managerName"))`
- D. `(storesDF.withColumnRenamed("state", "division").withColumnRenamed("managerFullName", "managerName"))`
- E. `(storesDF.withColumnRenamed("division", "state").withColumnRenamed("managerName", "managerFullName"))`

Answer: E

QUESTION 10

The code block shown contains an error. The code block is intended to return a new DataFrame where column sqft from DataFrame storesDF has had its missing values replaced with the value 30,000. Identify the error.

A sample of DataFrame storesDF is displayed below:

storeId	sqft
0	43161
1	51200
2	null
3	78367
4	null
...	...

Code block:

```
storesDF.na.fill(30000, col("sqft"))
```

- A. The argument to the subset parameter of fill() should be a string column name or a list of string column names rather than a Column object.
- B. The na.fill() operation does not work and should be replaced by the dropna() operation.
- C. The argument to the subset parameter of fill() should be the numerical position of the column rather than a Column object.
- D. The na.fill() operation does not work and should be replaced by the na.fill() operation.
- E. The na.fill() operation does not work and should be replaced by the fillna() operation.

Answer: A

QUESTION 11

Which of the following operations fails to return a DataFrame with no duplicate rows?

- A. DataFrame.dropDuplicates()
- B. DataFrame.distinct()
- C. DataFrame.drop_duplicates()
- D. DataFrame.drop_duplicates(subset = None)
- E. DataFrame.drop_duplicates(subset = "all")

Answer: E

QUESTION 12

Which of the following code blocks will most quickly return an approximation for the number of distinct values in column division in DataFrame storesDF?

- A. storesDF.agg(approx_count_distinct(col("division")).alias("divisionDistinct"))
- B. storesDF.agg(approx_count_distinct(col("division"), 0.01).alias("divisionDistinct"))
- C. storesDF.agg(approx_count_distinct(col("division"), 0.15).alias("divisionDistinct"))
- D. storesDF.agg(approx_count_distinct(col("division"), 0.0).alias("divisionDistinct"))
- E. storesDF.agg(approx_count_distinct(col("division"), 0.05).alias("divisionDistinct"))

Answer: C

QUESTION 13

The code block shown below contains an error. The code block is intended to return a new DataFrame with the mean of column sqft from DataFrame storesDF in column sqftMean. Identify the error.

Code block:

```
storesDF.agg(mean("sqft").alias("sqftMean"))
```

- A. The argument to the mean() operation should be a Column object rather than a string column name.
- B. The argument to the mean() operation should not be quoted.
- C. The mean() operation is not a standalone function - it's a method of the Column object.
- D. The agg() operation is not appropriate here - the withColumn() operation should be used instead.
- E. The only way to compute a mean of a column is with the mean() method from a DataFrame.

Answer: E

QUESTION 14

Which of the following operations can be used to return the number of rows in a DataFrame?

- A. DataFrame.numberOfRows()
- B. DataFrame.n()
- C. DataFrame.sum()
- D. DataFrame.count()
- E. DataFrame.countDistinct()

Answer: D

QUESTION 15

Which of the following operations returns a GroupedData object?

- A. DataFrame.GroupBy()
- B. DataFrame.cubed()
- C. DataFrame.group()
- D. DataFrame.groupBy()
- E. DataFrame.grouping_id()

Answer: D

QUESTION 16

Which of the following code blocks returns a collection of summary statistics for all columns in

DataFrame storesDF?

- A. storesDF.summary("mean")
- B. storesDF.describe(all = True)
- C. storesDF.describe("all")
- D. storesDF.summary("all")
- E. storesDF.describe()

Answer: E

QUESTION 17

Which of the following code blocks fails to return a DataFrame reverse sorted alphabetically based on column division?

- A. storesDF.orderBy("division", ascending - False)
- B. storesDF.orderBy(["division"], ascending = [0])
- C. storesDF.orderBy(col("division").asc())
- D. storesDF.sort("division", ascending - False)
- E. storesDF.sort(desc("division"))

Answer: C

QUESTION 18

Which of the following code blocks returns a 15 percent sample of rows from DataFrame storesDF without replacement?

- A. storesDF.sample(fraction = 0.10)
- B. storesDF.sampleBy(fraction = 0.15)
- C. storesDF.sample(True, fraction = 0.10)
- D. storesDF.sample()
- E. storesDF.sample(fraction = 0.15)

Answer: E

QUESTION 19

Which of the following code blocks returns all the rows from DataFrame storesDF?

- A. storesDF.head()
- B. storesDF.collect()
- C. storesDF.count()
- D. storesDF.take()
- E. storesDF.show()

Answer: B

QUESTION 20

Which of the following code blocks applies the function assessPerformance() to each row of DataFrame storesDF?

- A. [assessPerformance(row) for row in storesDF.take(3)]
- B. [assessPerformance() for row in storesDF]
- C. storesDF.collect().apply(lambda: assessPerformance)
- D. [assessPerformance(row) for row in storesDF.collect()]
- E. [assessPerformance(row) for row in storesDF]

Answer: D

QUESTION 21

The code block shown below contains an error. The code block is intended to print the schema of DataFrame storesDF. Identify the error.

Code block:
storesDF.printSchema

- A. There is no printSchema member of DataFrame - schema and the print() function should be used instead.
- B. The entire line needs to be a string - it should be wrapped by str().
- C. There is no printSchema member of DataFrame - the getSchema() operation should be used instead.
- D. There is no printSchema member of DataFrame - the schema() operation should be used instead.
- E. The printSchema member of DataFrame is an operation and needs to be followed by parentheses.

Answer: E

QUESTION 22

The code block shown below should create and register a SQL UDF named "ASSESS_PERFORMANCE" using the Python function assessPerformance() and apply it to column customerSatisfaction in table stores. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:
spark._1_.2_(3_, 4_)
spark.sql("SELECT customerSatisfaction, _5_(customerSatisfaction) AS
result FROM stores")

- A. 1. udf
2. register
3. "ASSESS_PERFORMANCE"
4. assessPerformance
5. ASSESS_PERFORMANCE
- B. 1. udf
2. register
3. assessPerformance
4. "ASSESS_PERFORMANCE"
5. "ASSESS_PERFORMANCE"
- C. 1. udf
2. register
3. "ASSESS_PERFORMANCE"
4. assessPerformance
5. "ASSESS_PERFORMANCE"

- D. 1. register
2. udf
3. "ASSESS_PERFORMANCE"
4. assessPerformance
5. "ASSESS_PERFORMANCE"
- E. 1. udf
2. register
3. ASSESS_PERFORMANCE
4. assessPerformance
5. ASSESS_PERFORMANCE

Answer: A

QUESTION 23

The code block shown below contains an error. The code block is intended to create a Python UDF `assessPerformanceUDF()` using the integer-returning Python function `assessPerformance()` and apply it to column `customerSatisfaction` in DataFrame `storesDF`. Identify the error.

Code block:

```
assessPerformanceUDF = udf(assessPerformance)
storesDF.withColumn("result",
assessPerformanceUDF(col("customerSatisfaction")))
```

- A. The `assessPerformance()` operation is not properly registered as a UDF.
- B. The `withColumn()` operation is not appropriate here - UDFs should be applied by iterating over rows instead.
- C. UDFs can only be applied via SQL and not through the DataFrame API.
- D. The return type of the `assessPerformanceUDF()` is not specified in the `udf()` operation.
- E. The `assessPerformance()` operation should be used on column `customerSatisfaction` rather than the `assessPerformanceUDF()` operation.

Answer: D

QUESTION 24

The code block shown below contains an error. The code block is intended to use SQL to return a new DataFrame containing column `storeId` and column `managerName` from a table created from DataFrame `storesDF`. Identify the error.

Code block:

```
storesDF.createOrReplaceTempView("stores")
storesDF.sql("SELECT storeId, managerName FROM stores")
```

- A. The `createOrReplaceTempView()` operation does not make a DataFrame accessible via SQL.
- B. The `sql()` operation should be accessed via the spark variable rather than DataFrame `storesDF`.
- C. There is the `sql()` operation in DataFrame `storesDF`. The operation `query()` should be used instead.
- D. This cannot be accomplished using SQL - the DataFrame API should be used instead.
- E. The `createOrReplaceTempView()` operation should be accessed via the spark variable rather than DataFrame `storesDF`.

Answer: B

Thank You for Trying Our Product

Braindump2go Certification Exam Features:

- ★ More than **99,900** Satisfied Customers Worldwide.
- ★ Average **99.9%** Success Rate.
- ★ **Free Update** to match latest and real exam scenarios.
- ★ **Instant Download** Access! No Setup required.
- ★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.
- ★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad**.
- ★ **100%** Guaranteed Success or **100%** Money Back Guarantee.
- ★ **Fast**, helpful support **24x7**.



View list of all certification exams: <http://www.braindump2go.com/all-products.html>



Microsoft



ORACLE



CITRIX



JUNIPER
NETWORKS



EMC²
where information lives

10% Discount Coupon Code: ASTR14