



VCE & PDF

GeekCert.com

<https://www.geekcert.com/databricks-certified-associate-developer-for-apache-spark>
2024 Latest geekcert DATABRICKS-CERTIFIED-ASSOCIATE-DEVELOPER-
FOR-APACHE-SPARK PDF and VCE dumps Download

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.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



VCE & PDF

GeekCert.com

<https://www.geekcert.com/databricks-certified-associate-developer-for-apache-spark/>
2024 Latest geekcert DATABRICKS-CERTIFIED-ASSOCIATE-DEVELOPER-FOR-APACHE-SPARK PDF and VCE dumps Download

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





QUESTION 1

The code block displayed below contains an error. The code block should produce a DataFrame with color as the only column and three rows with color values of red, blue, and green, respectively.

Find the error.

Code block:

```
1.spark.createDataFrame([("red",), ("blue",), ("green",)], "color")
```

2.Instead of calling spark.createDataFrame, just DataFrame should be called.

- A. The commas in the tuples with the colors should be eliminated.
- B. The colors red, blue, and green should be expressed as a simple Python list, and not a list of tuples.
- C. Instead of color, a data type should be specified.
- D. The "color" expression needs to be wrapped in brackets, so it reads ["color"].

Correct Answer: D

QUESTION 2

Which of the following describes slots?

- A. Slots are dynamically created and destroyed in accordance with an executor's workload.
- B. To optimize I/O performance, Spark stores data on disk in multiple slots.
- C. A Java Virtual Machine (JVM) working as an executor can be considered as a pool of slots for task execution.
- D. A slot is always limited to a single core. Slots are the communication interface for executors and are used for receiving commands and sending results to the driver.

Correct Answer: C

Slots are the communication interface for executors and are used for receiving commands and sending results to the driver. Wrong, executors communicate with the driver directly. Slots are dynamically created and destroyed in accordance with an executor's workload. No, Spark does not actively create and destroy slots in accordance with the workload. Per executor, slots are made available in accordance with how many cores per executor (property `spark.executor.cores`) and how many CPUs per task (property `spark.task.cpus`) the Spark configuration calls for. A slot is always limited to a single core. No, a slot can span multiple cores. If a task would require multiple cores, it would have to be executed through a slot that spans multiple cores. In Spark documentation, "core" is often used interchangeably with "thread", although "thread" is the more accurate word. A single physical core may be able to make multiple threads available. So, it is better to say that a slot can span multiple threads. To optimize I/O performance, Spark stores data on disk in multiple slots. No ?Spark stores data on disk in multiple partitions, not slots. More info: Spark Architecture | Distributed Systems Architecture



QUESTION 3

The code block displayed below contains an error. The code block should arrange the rows of DataFrame transactionsDf using information from two columns in an ordered fashion, arranging first by column value, showing smaller numbers at the top and greater numbers at the bottom, and then by column predError, for which all values should be arranged in the inverse way of the order of items in column value. Find the error.

Code block:

```
transactionsDf.orderBy(\'\'value\'', asc_nulls_first(col(\'\'predError\'')))
```

- A. Two orderBy statements with calls to the individual columns should be chained, instead of having both columns in one orderBy statement.
- B. Column value should be wrapped by the col() operator.
- C. Column predError should be sorted in a descending way, putting nulls last.
- D. Column predError should be sorted by desc_nulls_first() instead.
- E. Instead of orderBy, sort should be used.

Correct Answer: C

QUESTION 4

The code block shown below should return all rows of DataFrame itemsDf that have at least 3 items in column itemNameElements. Choose the answer that correctly fills the blanks in the code block to accomplish this.

Example of DataFrame itemsDf:

```
1. +-----+-----+-----+-----+
2. |itemId|itemName |supplier |itemNameElements |
3. +-----+-----+-----+-----+
4. |1 |Thick Coat for Walking in the Snow|Sports Company Inc.|[Thick, Coat, for, Walking, in, the, Snow]|
5. |2 |Elegant Outdoors Summer Dress |YetiX |[Elegant, Outdoors, Summer, Dress] |
6. |3 |Outdoors Backpack |Sports Company Inc.|[Outdoors, Backpack] |
7. +-----+-----+-----+-----+
```

Code block:

```
itemsDf.__1__(__2__(__3__)__4__)
```

- A. 1. select
- 2.



count

3.

col("itemNameElements")

4.

>3

B. 1. filter

2.

count

3.

itemNameElements

4.

>=3

C. 1. select

2.

count

3.

"itemNameElements"

4.

>3

D. 1. filter

2.

size

3.

"itemNameElements"

4.

>=3

E. 1. select

2.

size



3.

"itemNameElements"

4.

>3

Correct Answer: D

Correct code block:

```
itemsDf.filter(size("itemNameElements")>3)
```

Output of code block:

```
+-----+-----+-----+-----+ |itemId|itemName |
supplier |itemNameElements |
+-----+-----+-----+-----+ |1 |Thick Coat for
Walking in the Snow|Sports Company Inc.|[Thick, Coat, for, Walking, in, the, Snow]|
|2 |Elegant Outdoors Summer Dress |YetiX |[Elegant, Outdoors, Summer, Dress] |
+-----+-----+-----+-----+ The big difficulty with this is
```

in knowing the difference between count and size (refer to documentation below). size is the correct function to choose here since it returns the number of elements in an array on a per-row basis.

The other consideration for solving this is the difference between select and filter. Since we want to return the rows in the original DataFrame, filter is the right choice. If we would use select, we would simply get a single-column DataFrame showing which rows match the criteria, like so:

```
+-----+
|(size(itemNameElements) > 3)|
+-----+
|true |
|true |
|false |
+-----+
```

More info:

Count documentation: [pyspark.sql.functions.count](#) -- PySpark 3.1.1 documentation Size documentation:



pyspark.sql.functions.size -- PySpark 3.1.1 documentation Static notebook | Dynamic notebook: See test

1, 47 (Databricks import instructions)

QUESTION 5

Which of the following code blocks selects all rows from DataFrame transactionsDf in which column productId is zero or smaller or equal to 3?

A. transactionsDf.filter(productId==3 or productId