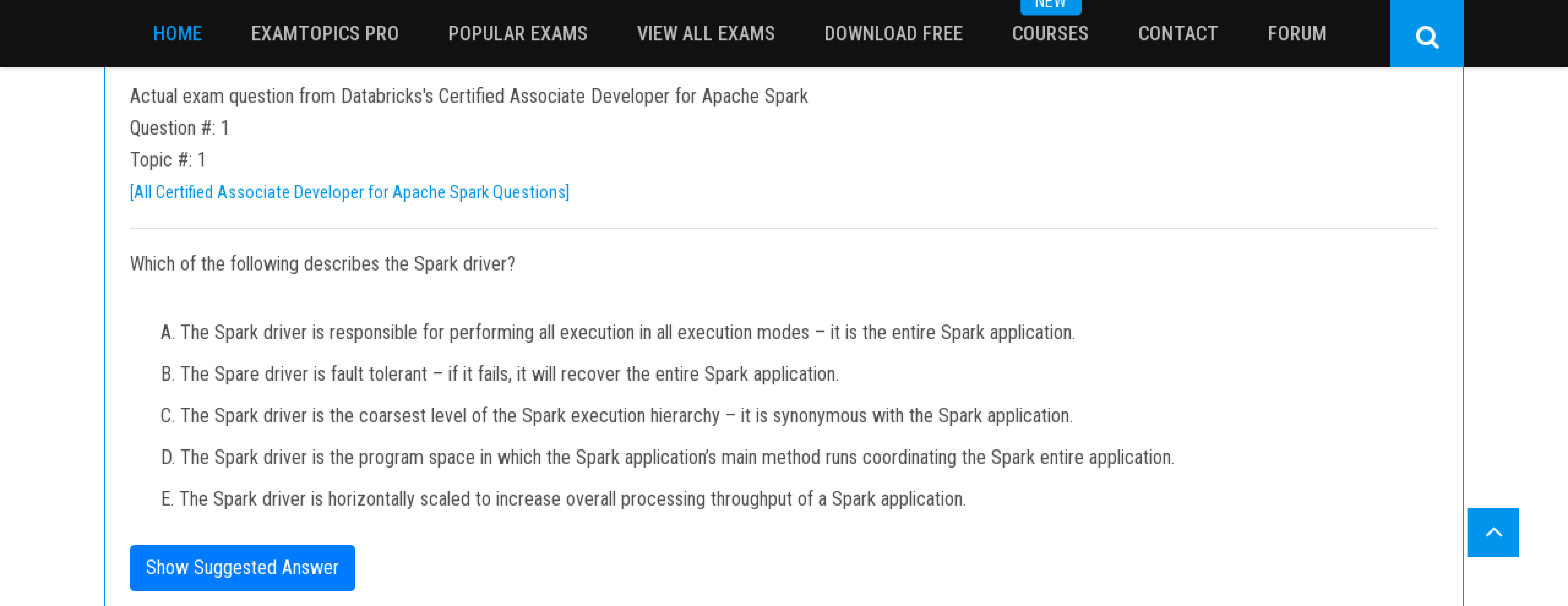


Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 1

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following describes the Spark driver?

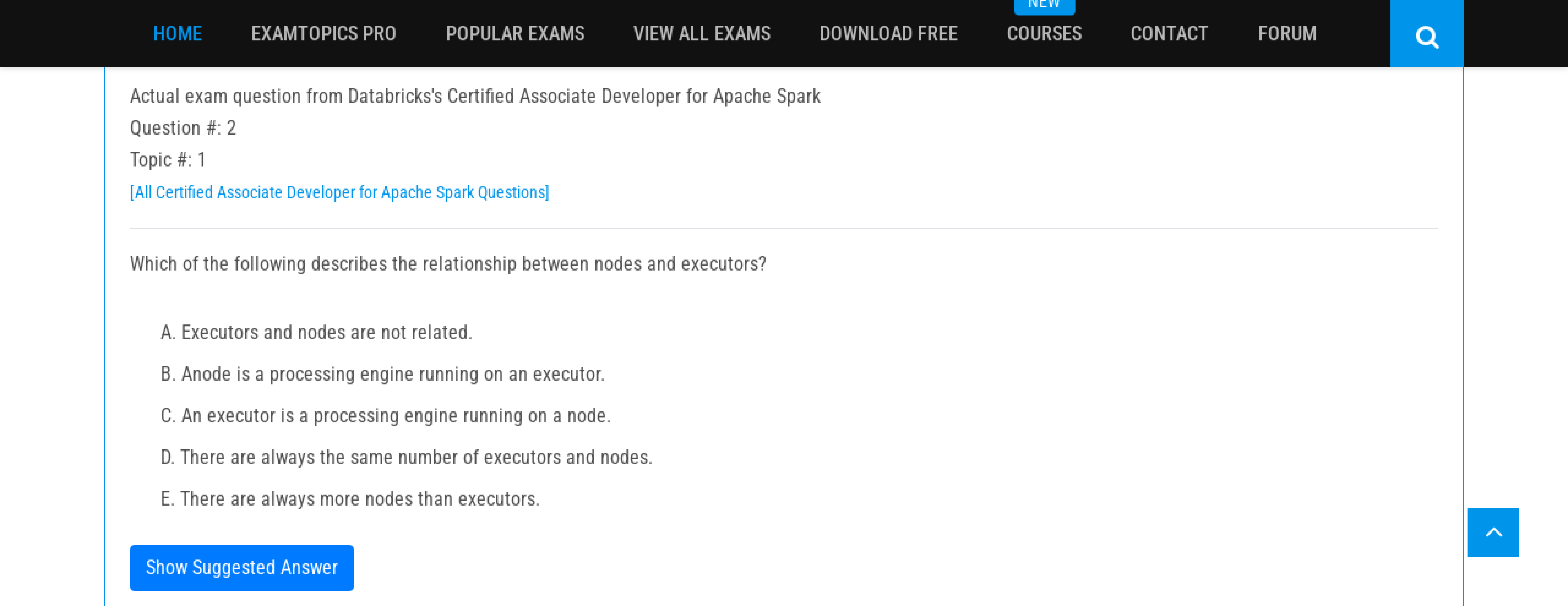
A. The Spark driver is responsible for performing all execution in all execution modes – it is the entire Spark application.

B. The Spare driver is fault tolerant – if it fails, it will recover the entire Spark application.

C. The Spark driver is the coarsest level of the Spark execution hierarchy – it is synonymous with the Spark application.

D. The Spark driver is the program space in which the Spark application’s main method runs coordinating the Spark entire application.

E. The Spark driver is horizontally scaled to increase overall processing throughput of a Spark application.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 2

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following describes the relationship between nodes and executors?

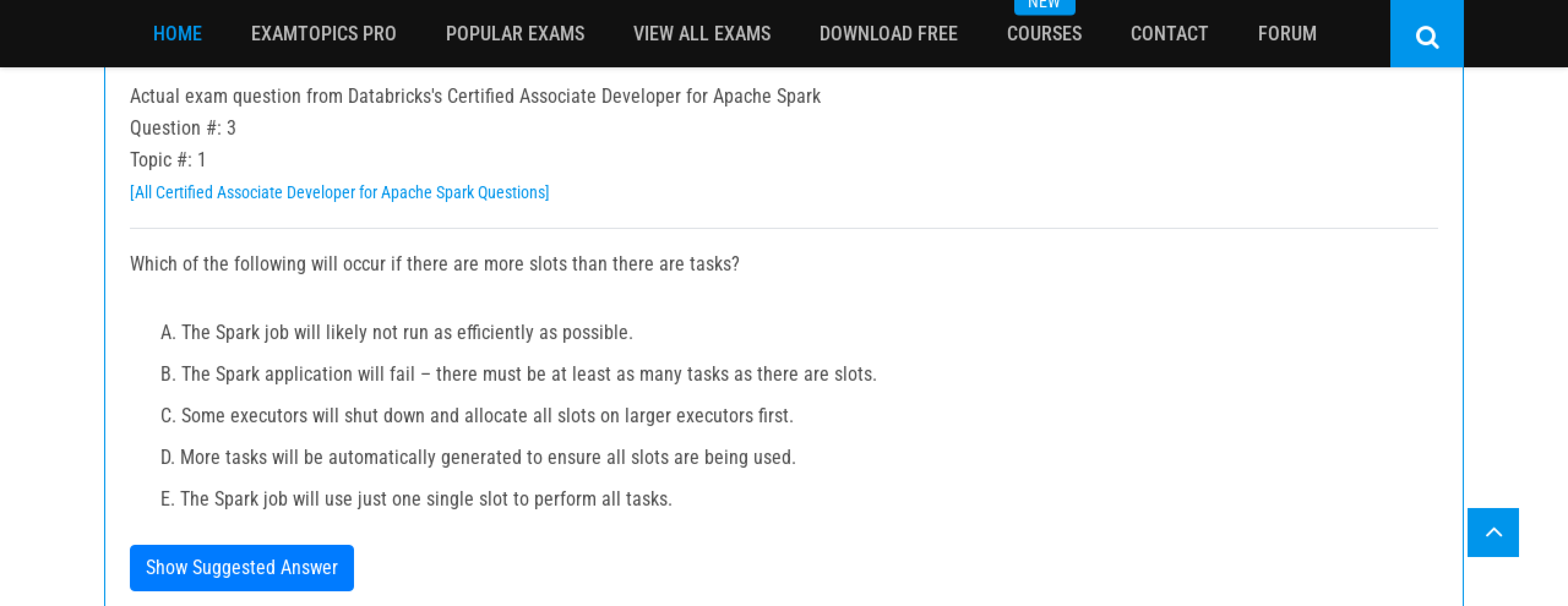
A. Executors and nodes are not related.

B. Anode is a processing engine running on an executor.

C. An executor is a processing engine running on a node.

D. There are always the same number of executors and nodes.

E. There are always more nodes than executors.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 3

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following will occur if there are more slots than there are tasks?

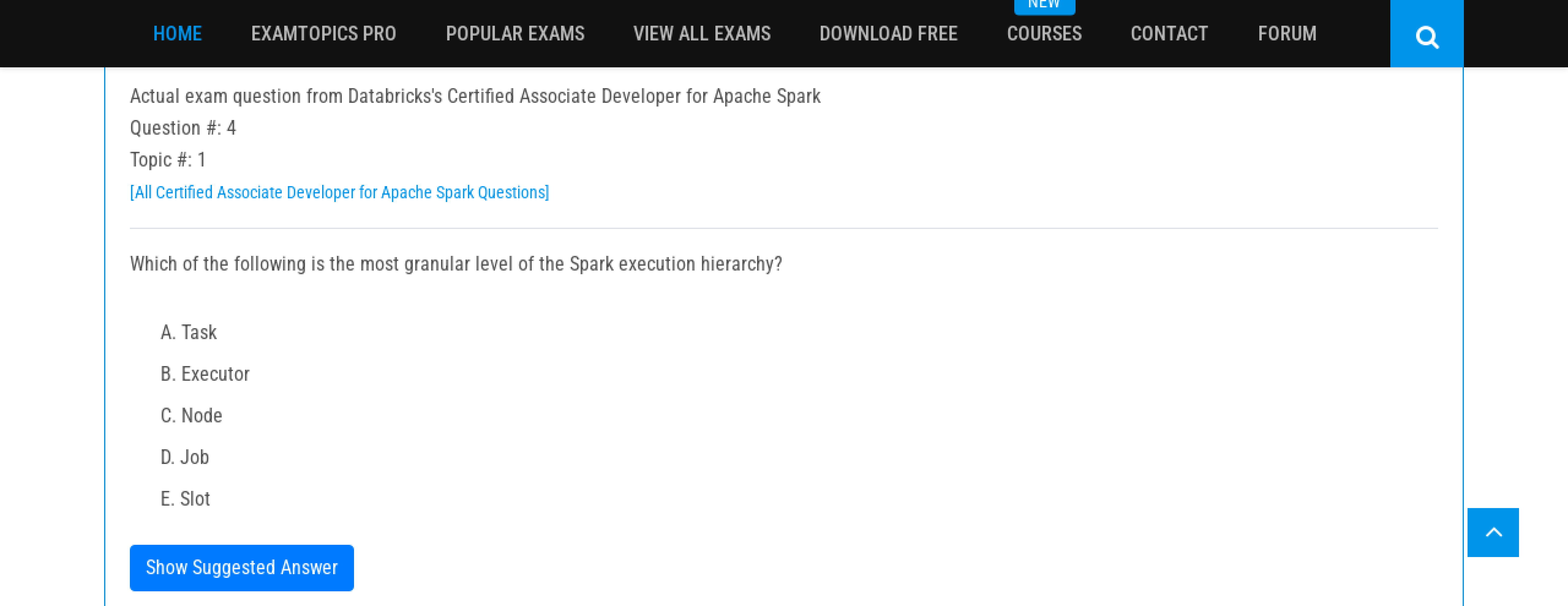
A. The Spark job will likely not run as efficiently as possible.

B. The Spark application will fail – there must be at least as many tasks as there are slots.

C. Some executors will shut down and allocate all slots on larger executors first.

D. More tasks will be automatically generated to ensure all slots are being used.

E. The Spark job will use just one single slot to perform all tasks.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 4

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following is the most granular level of the Spark execution hierarchy?

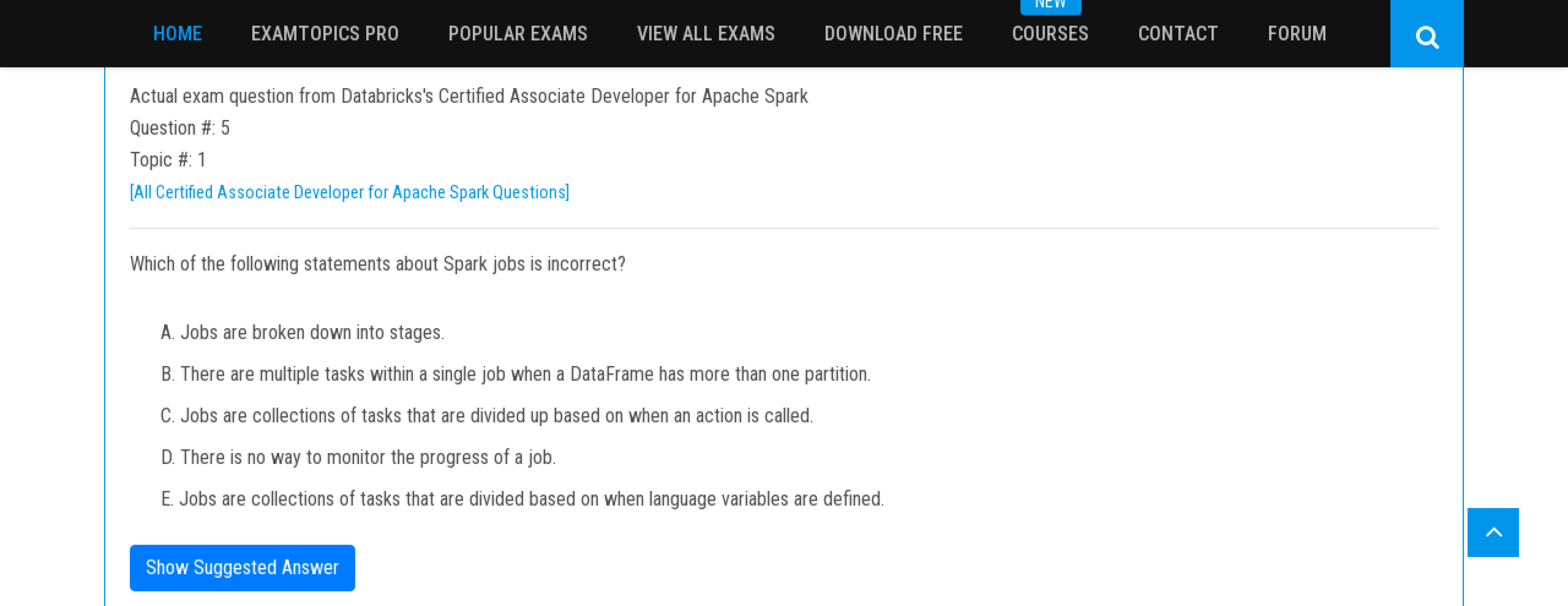
A. Task

B. Executor

C. Node

D. Job

E. Slot



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 5

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following statements about Spark jobs is incorrect?

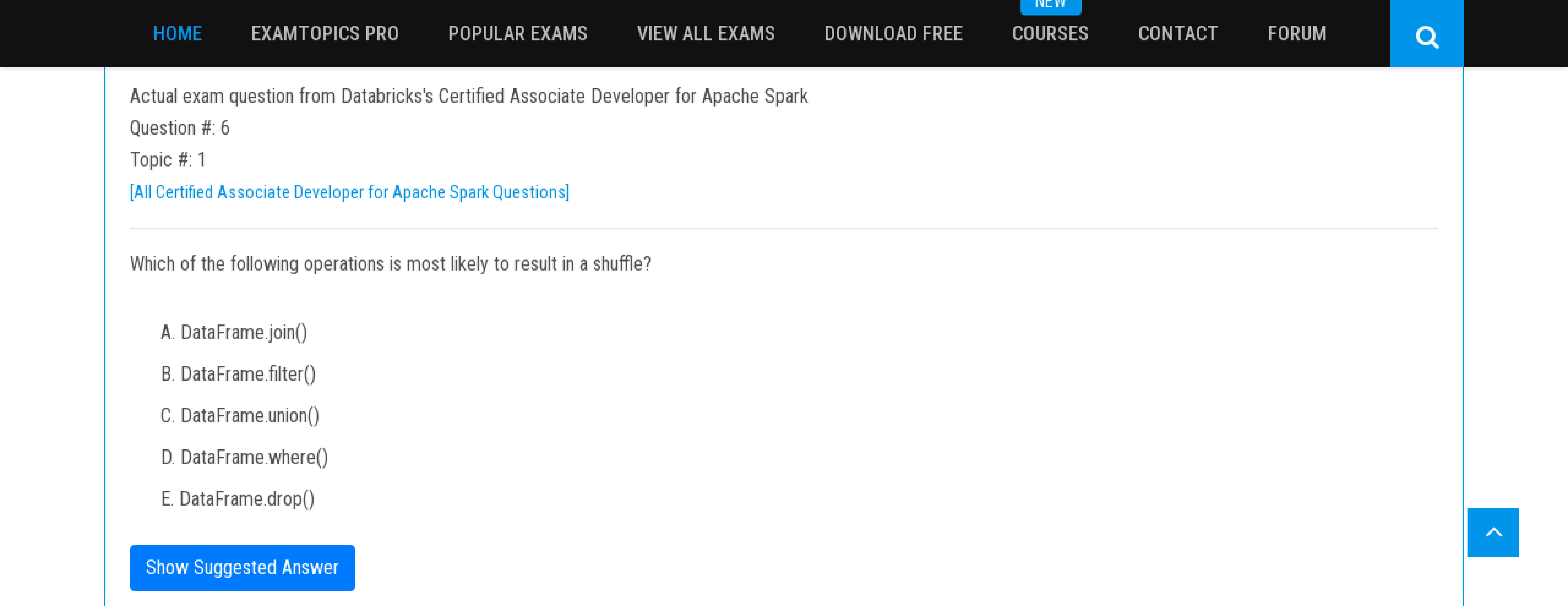
A. Jobs are broken down into stages.

B. There are multiple tasks within a single job when a DataFrame has more than one partition.

C. Jobs are collections of tasks that are divided up based on when an action is called.

D. There is no way to monitor the progress of a job.

E. Jobs are collections of tasks that are divided based on when language variables are defined.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 6

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following operations is most likely to result in a shuffle?

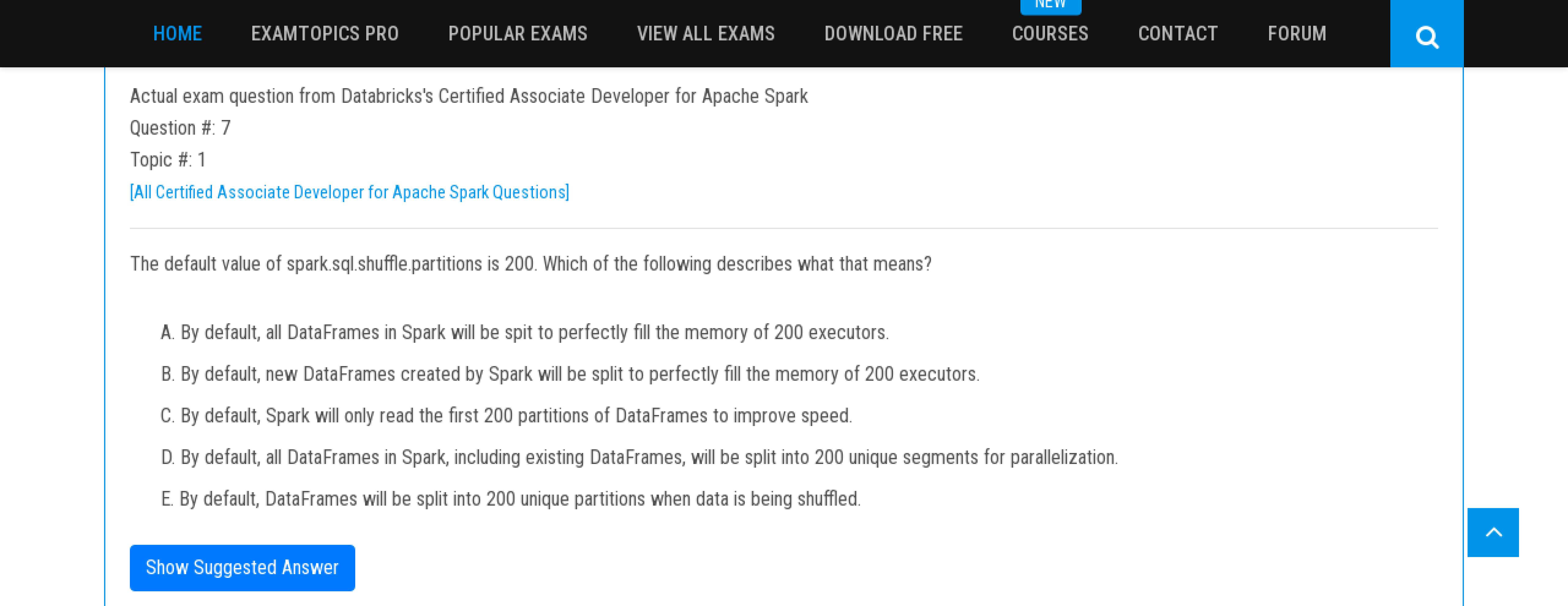
A. DataFrame.join()

B. DataFrame.filter()

C. DataFrame.union()

D. DataFrame.where()

E. DataFrame.drop()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 7

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The default value of spark.sql.shuffle.partitions is 200. Which of the following describes what that means?

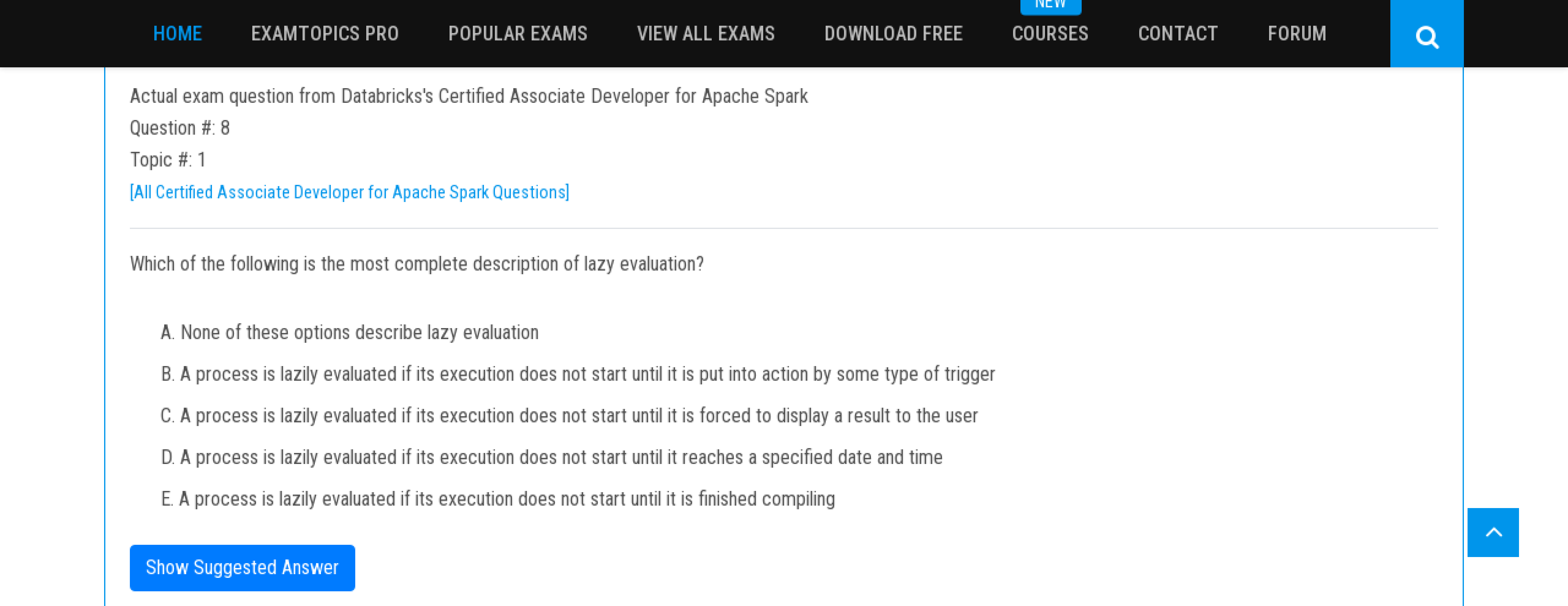
A. By default, all DataFrames in Spark will be spit to perfectly fill the memory of 200 executors.

B. By default, new DataFrames created by Spark will be split to perfectly fill the memory of 200 executors.

C. By default, Spark will only read the first 200 partitions of DataFrames to improve speed.

D. By default, all DataFrames in Spark, including existing DataFrames, will be split into 200 unique segments for parallelization.

E. By default, DataFrames will be split into 200 unique partitions when data is being shuffled.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 8

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following is the most complete description of lazy evaluation?

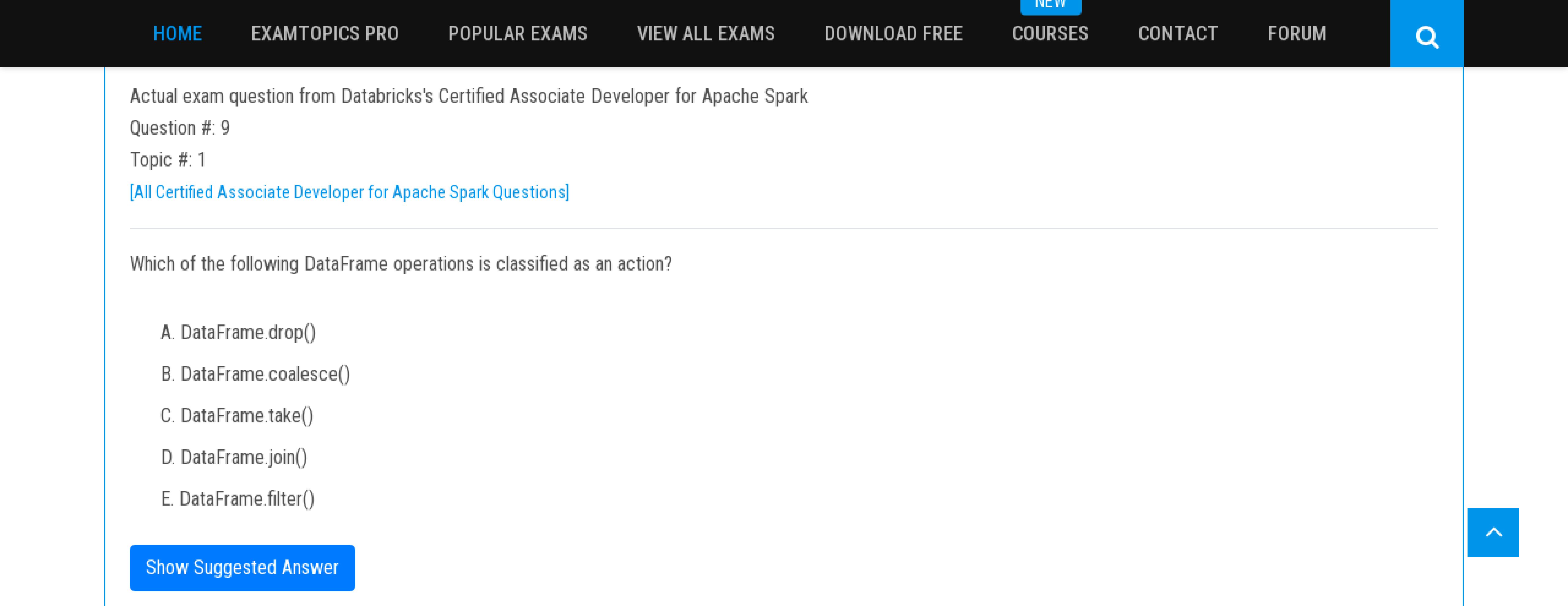
A. None of these options describe lazy evaluation

B. A process is lazily evaluated if its execution does not start until it is put into action by some type of trigger

C. A process is lazily evaluated if its execution does not start until it is forced to display a result to the user

D. A process is lazily evaluated if its execution does not start until it reaches a specified date and time

E. A process is lazily evaluated if its execution does not start until it is finished compiling



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 9

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following DataFrame operations is classified as an action?

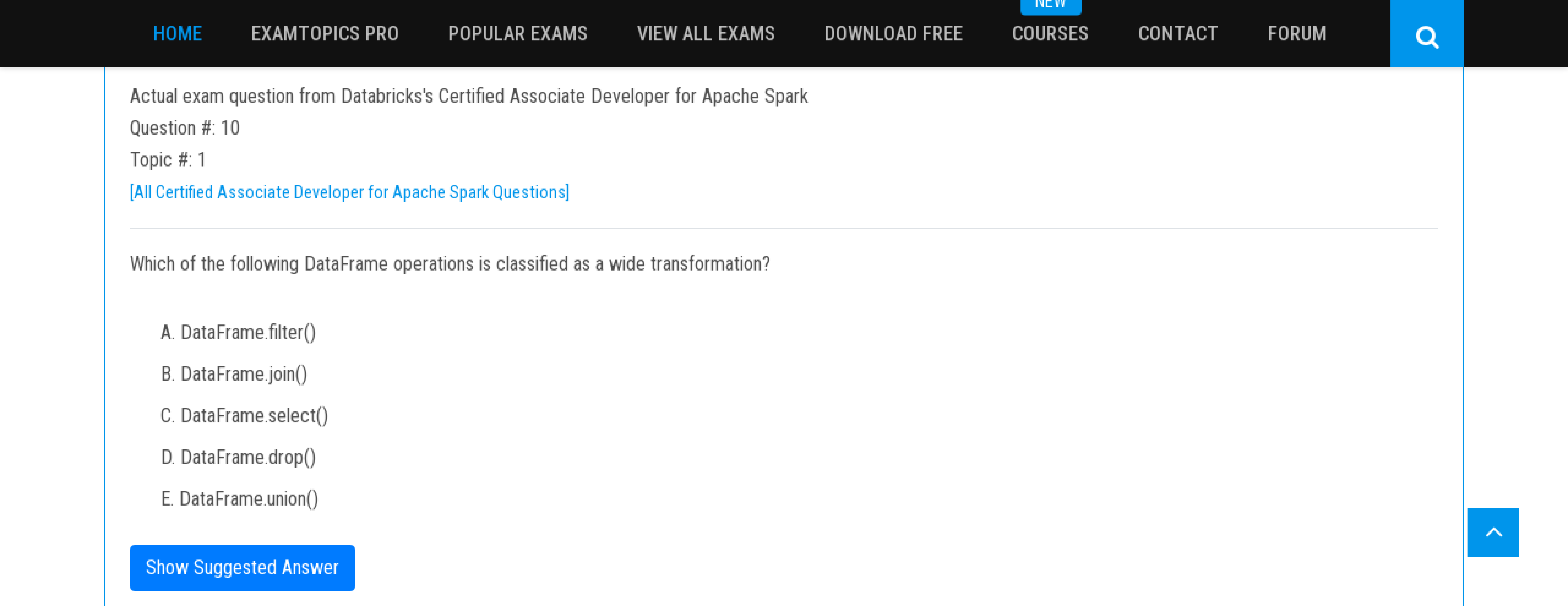
A. DataFrame.drop()

B. DataFrame.coalesce()

C. DataFrame.take()

D. DataFrame.join()

E. DataFrame.filter()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 10

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following DataFrame operations is classified as a wide transformation?

A. DataFrame.filter()

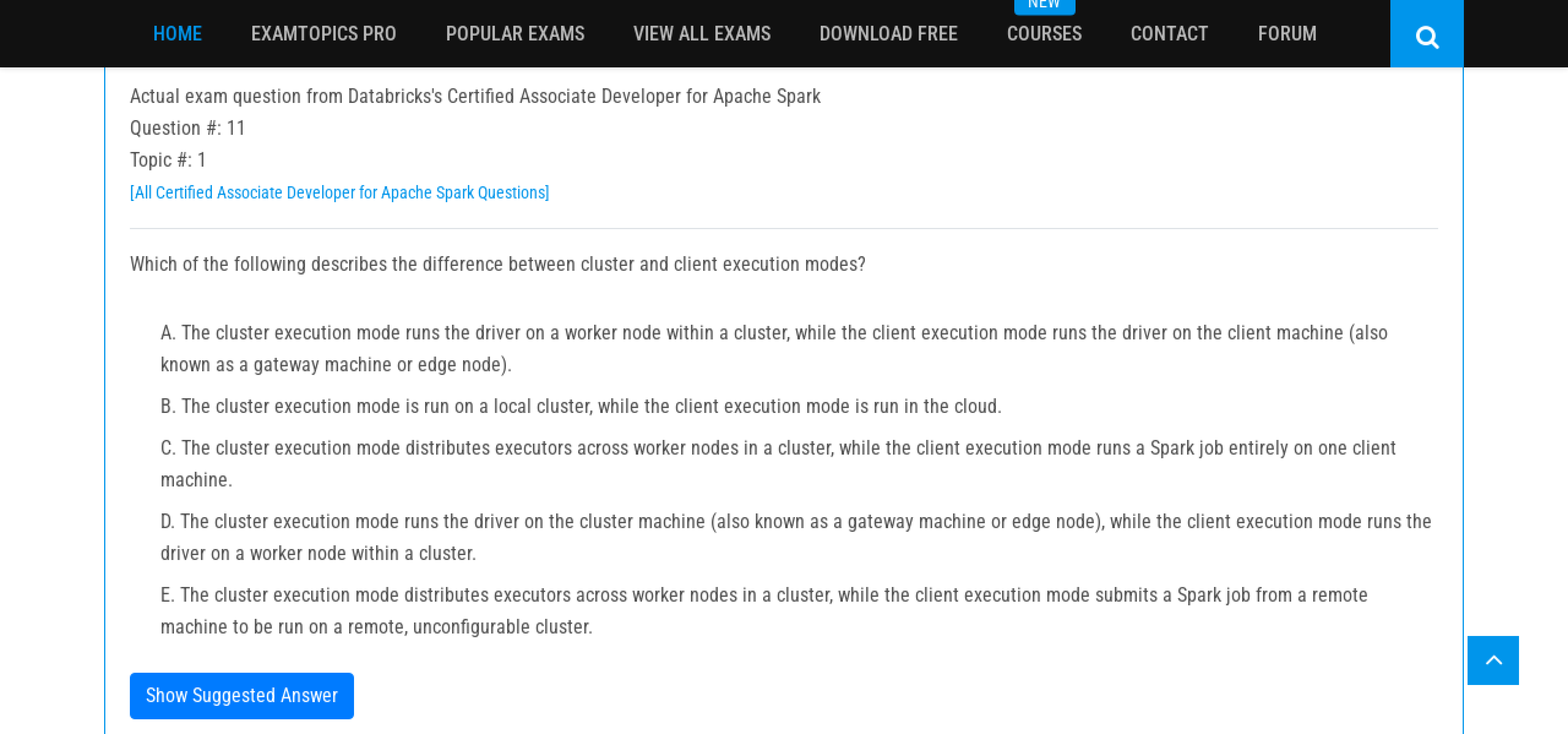
B. DataFrame.join()

C. DataFrame.select()

D. DataFrame.drop()

E. DataFrame.union()

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 11

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following describes the difference between cluster and client execution modes?

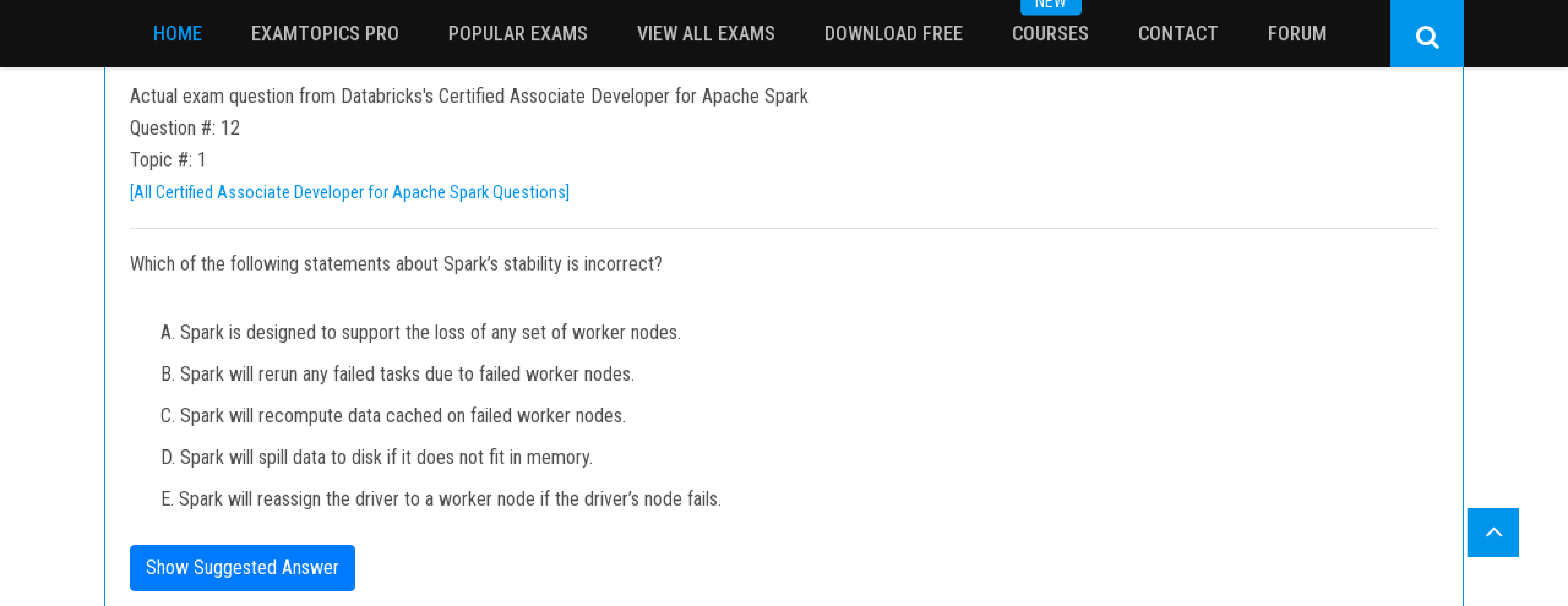
A. The cluster execution mode runs the driver on a worker node within a cluster, while the client execution mode runs the driver on the client machine (also known as a gateway machine or edge node).

B. The cluster execution mode is run on a local cluster, while the client execution mode is run in the cloud.

C. The cluster execution mode distributes executors across worker nodes in a cluster, while the client execution mode runs a Spark job entirely on one client machine.

D. The cluster execution mode runs the driver on the cluster machine (also known as a gateway machine or edge node), while the client execution mode runs the driver on a worker node within a cluster.

E. The cluster execution mode distributes executors across worker nodes in a cluster, while the client execution mode submits a Spark job from a remote machine to be run on a remote, unconfigurable cluster.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 12

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following statements about Spark's stability is incorrect?

A. Spark is designed to support the loss of any set of worker nodes.

B. Spark will rerun any failed tasks due to failed worker nodes.

C. Spark will recompute data cached on failed worker nodes.

D. Spark will spill data to disk if it does not fit in memory.

E. Spark will reassign the driver to a worker node if the driver's node fails.

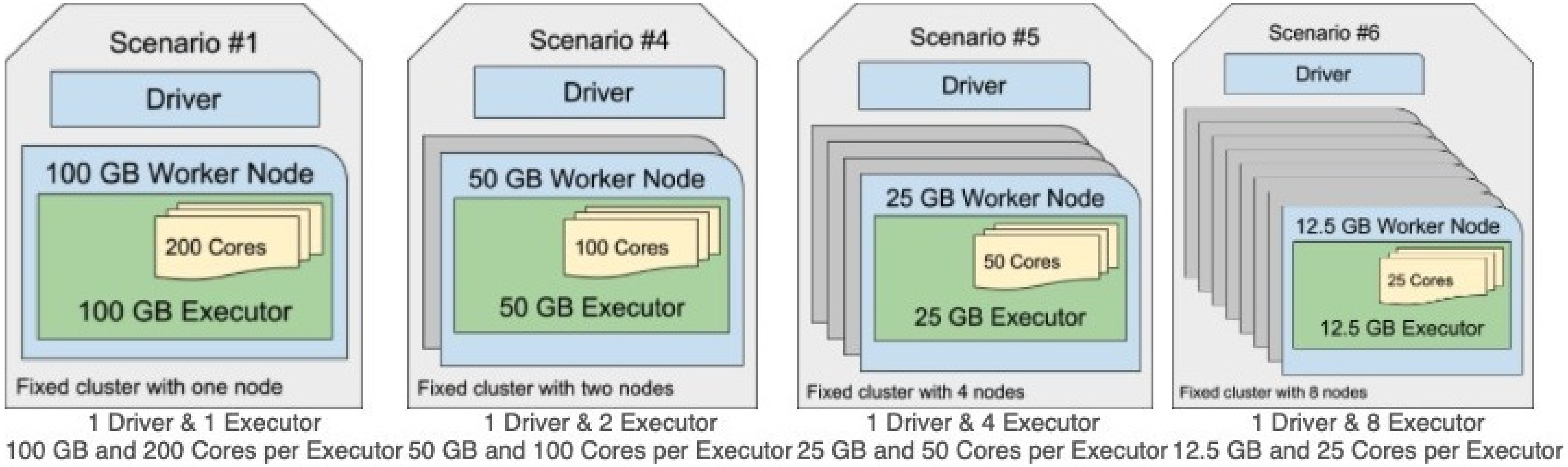
Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 13

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following cluster configurations is most likely to experience an out-of-memory error in response to data skew in a single partition?



Note: each configuration has roughly the same compute power using 100 GB of RAM and 200 cores.

A. Scenario #4

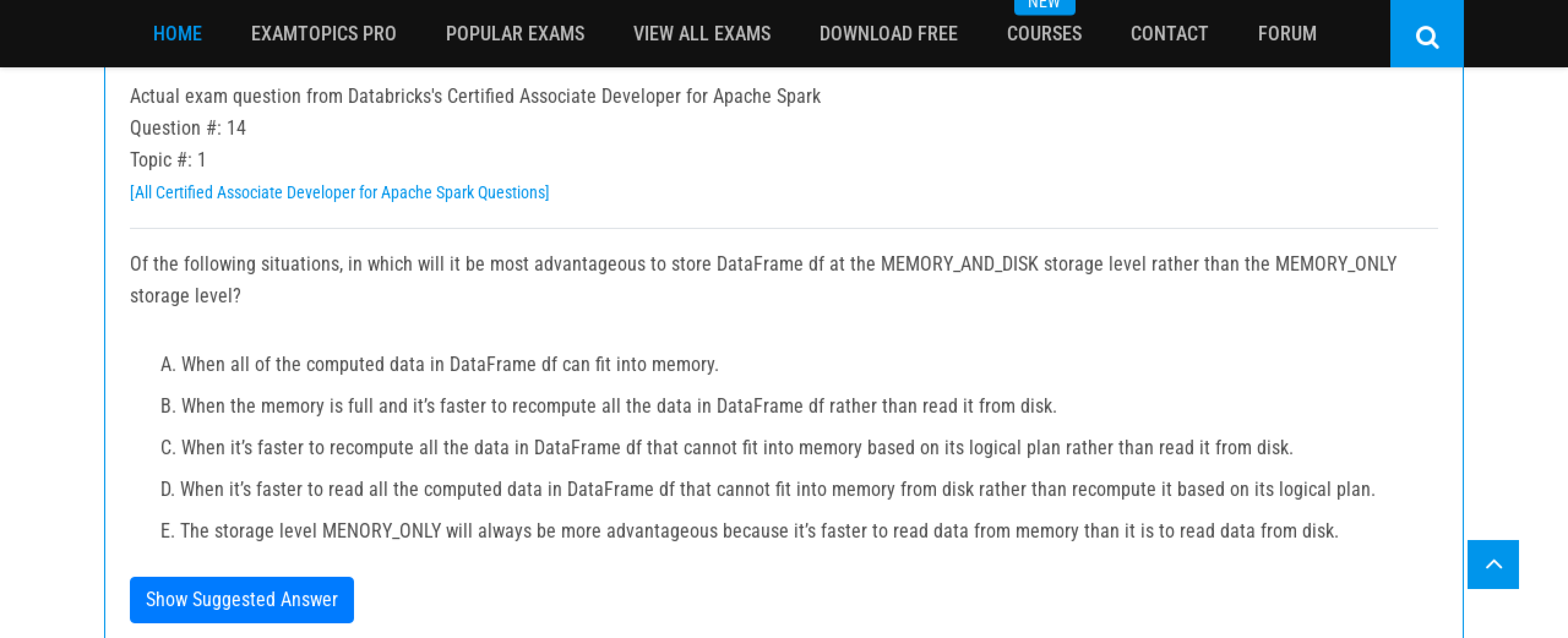
B. Scenario #5

C. Scenario #6

D. More information is needed to determine an answer.

E. Scenario #1

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 14

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Of the following situations, in which will it be most advantageous to store DataFrame df at the MEMORY_AND_DISK storage level rather than the MEMORY_ONLY storage level?

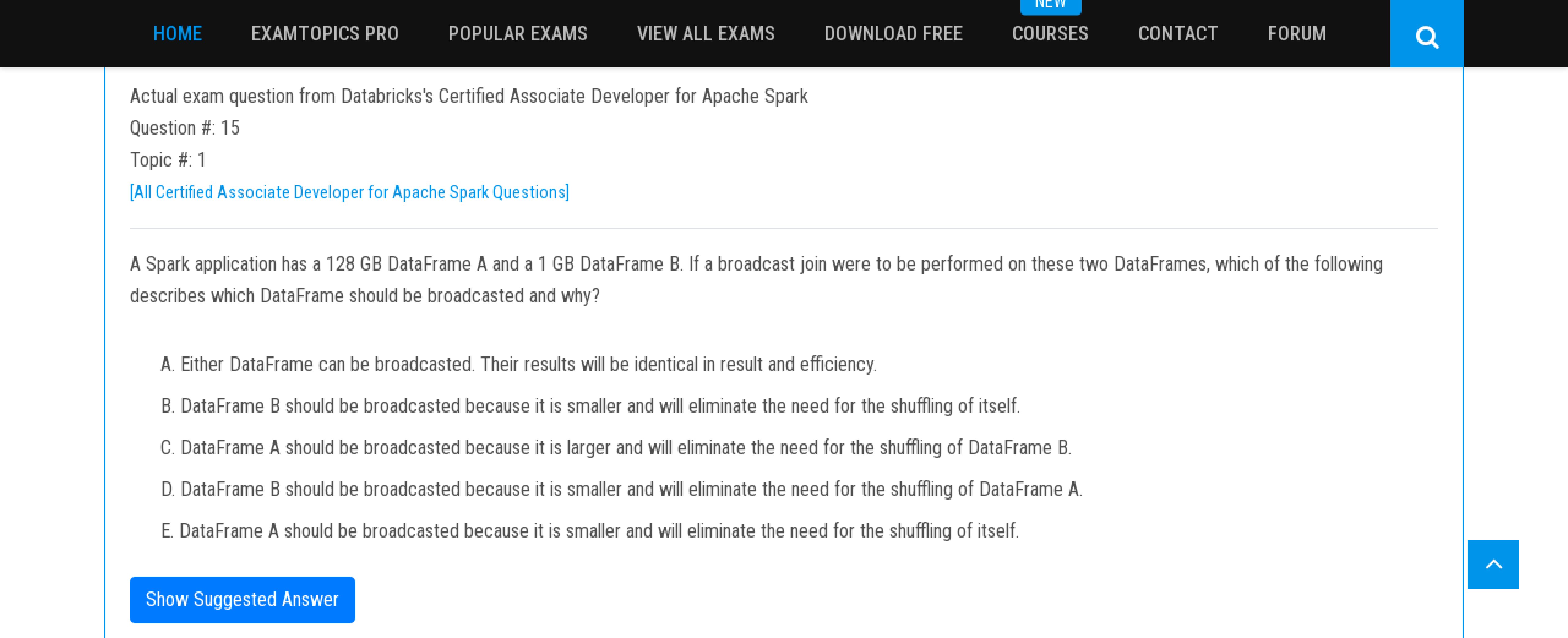
A. When all of the computed data in DataFrame df can fit into memory.

B. When the memory is full and it’s faster to recompute all the data in DataFrame df rather than read it from disk.

C. When it’s faster to recompute all the data in DataFrame df that cannot fit into memory based on its logical plan rather than read it from disk.

D. When it’s faster to read all the computed data in DataFrame df that cannot fit into memory from disk rather than recompute it based on its logical plan.

E. The storage level MENORY_ONLY will always be more advantageous because it’s faster to read data from memory than it is to read data from disk.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 15

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

A Spark application has a 128 GB DataFrame A and a 1 GB DataFrame B. If a broadcast join were to be performed on these two DataFrames, which of the following describes which DataFrame should be broadcasted and why?

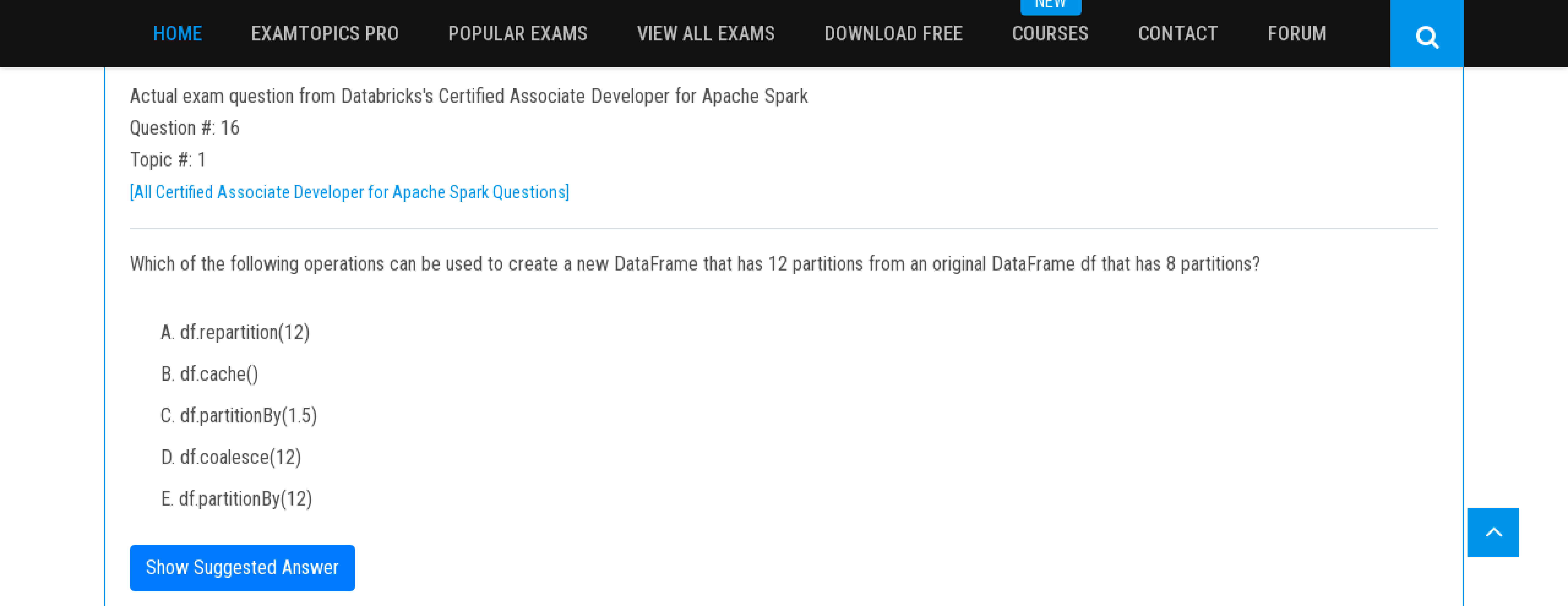
A. Either DataFrame can be broadcasted. Their results will be identical in result and efficiency.

B. DataFrame B should be broadcasted because it is smaller and will eliminate the need for the shuffling of itself.

C. DataFrame A should be broadcasted because it is larger and will eliminate the need for the shuffling of DataFrame B.

D. DataFrame B should be broadcasted because it is smaller and will eliminate the need for the shuffling of DataFrame A.

E. DataFrame A should be broadcasted because it is smaller and will eliminate the need for the shuffling of itself.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 16

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following operations can be used to create a new DataFrame that has 12 partitions from an original DataFrame df that has 8 partitions?

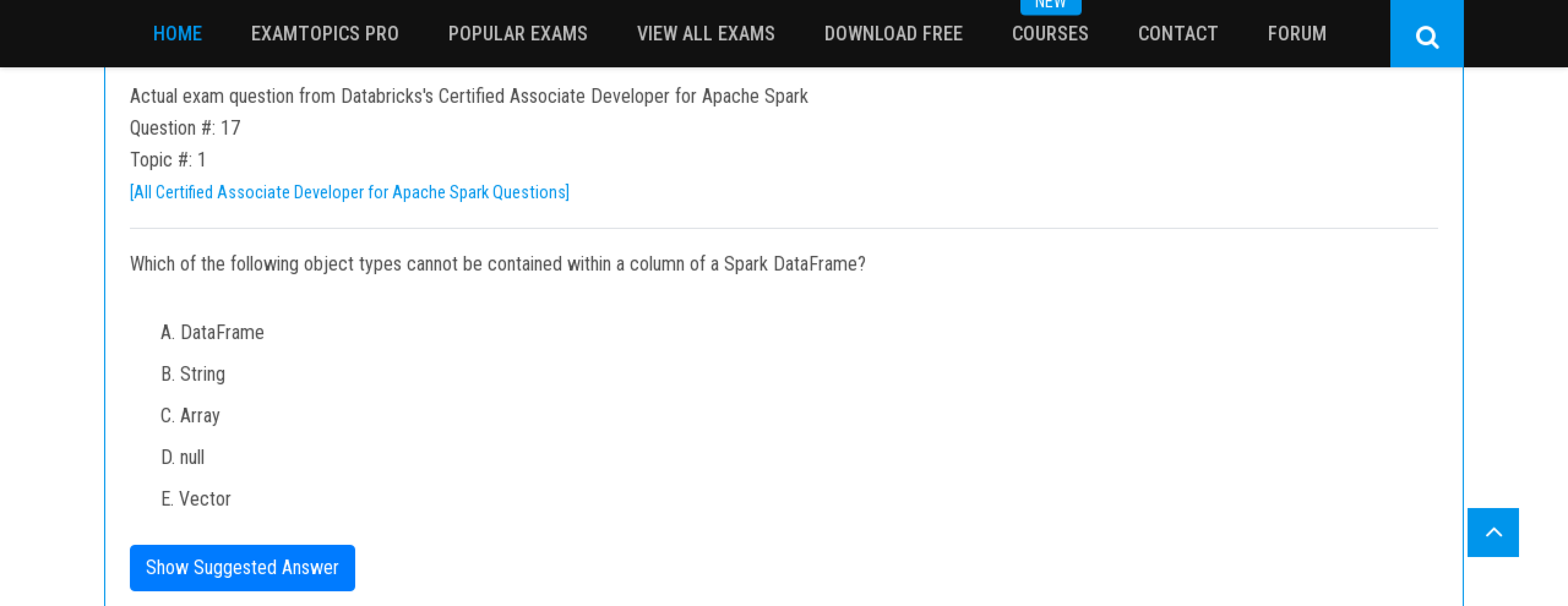
A. df.repartition(12)

B. df.cache()

C. df.partitionBy(1.5)

D. df.coalesce(12)

E. df.partitionBy(12)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 17

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following object types cannot be contained within a column of a Spark DataFrame?

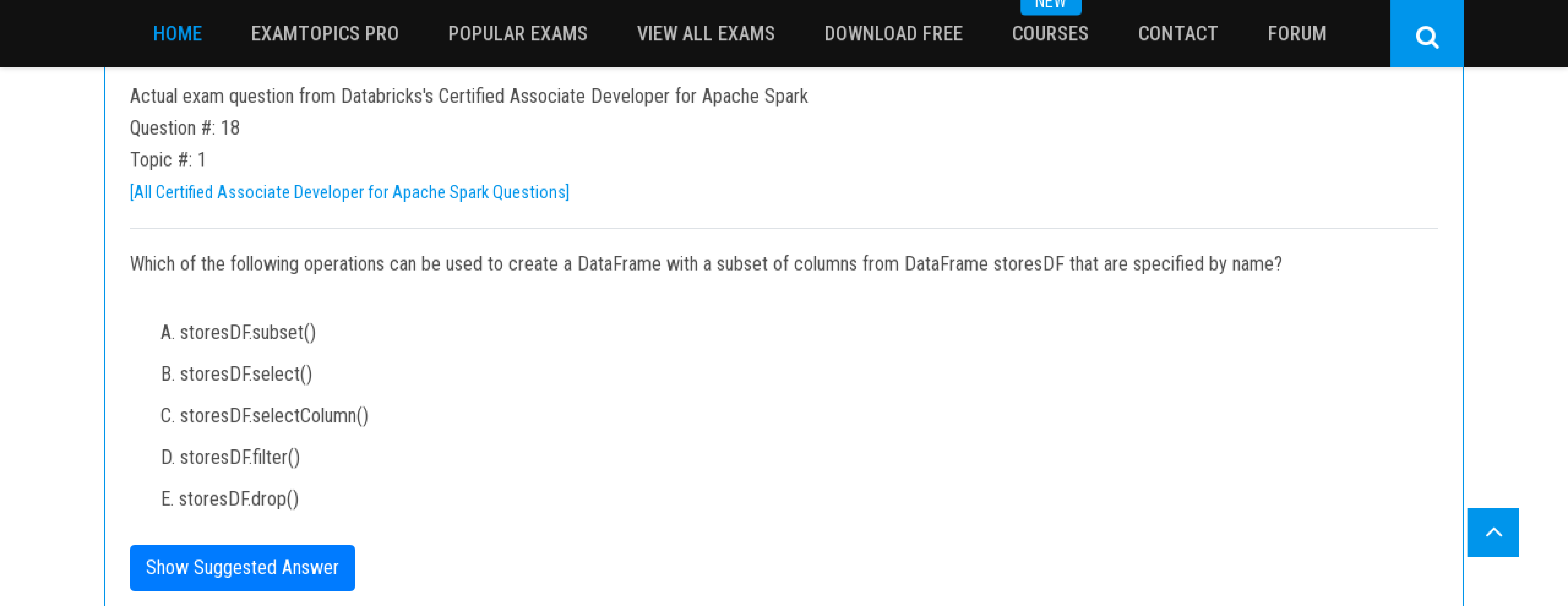
A. DataFrame

B. String

C. Array

D. null

E. Vector



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 18

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following operations can be used to create a DataFrame with a subset of columns from DataFrame storesDF that are specified by name?

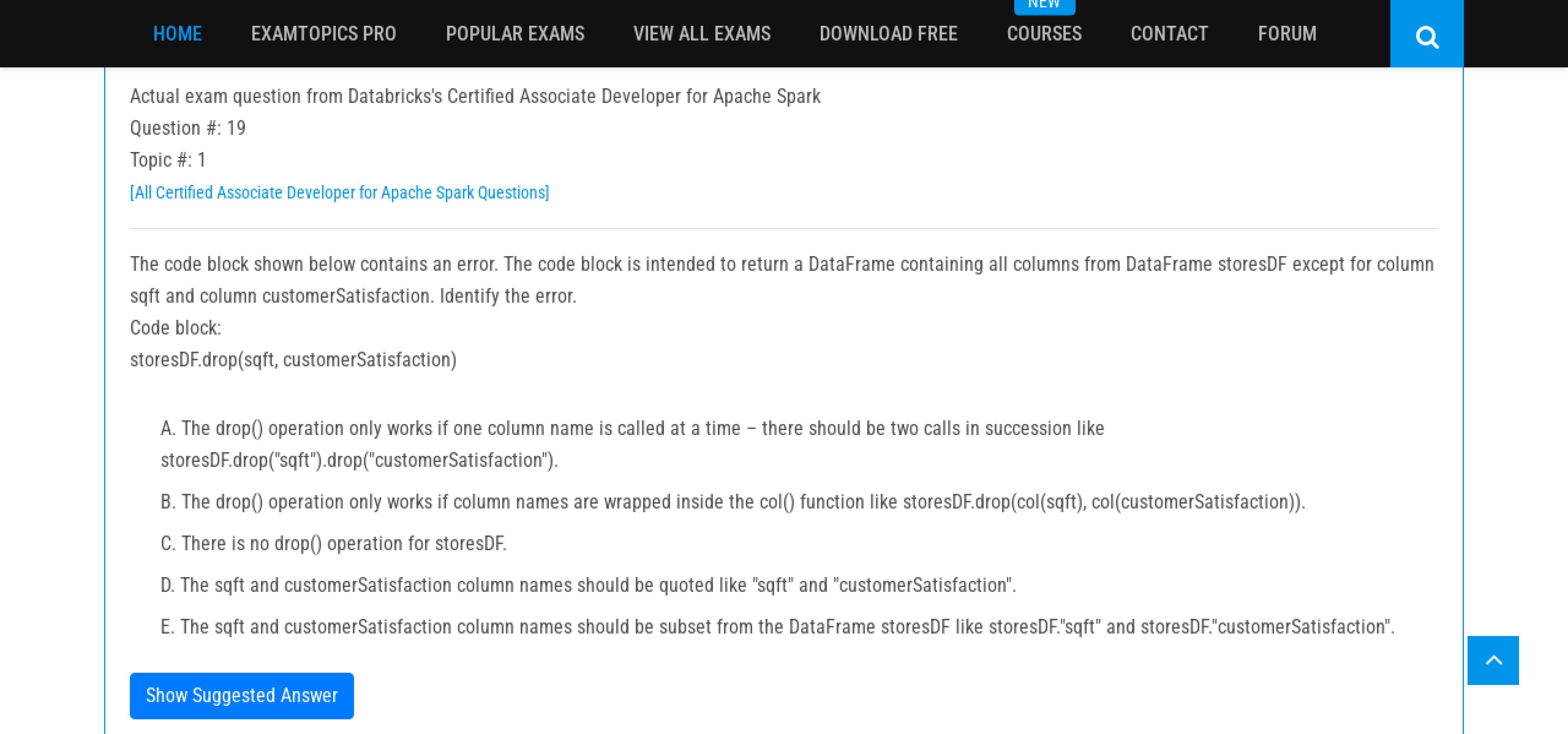
A. storesDF.subset()

B. storesDF.select()

C. storesDF.selectColumn()

D. storesDF.filter()

E. storesDF.drop()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 19

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to return a DataFrame containing all columns from DataFrame storesDF except for column sqft and column customerSatisfaction. Identify the error.

Code block:

```
storesDF.drop(sqft, customerSatisfaction)
```

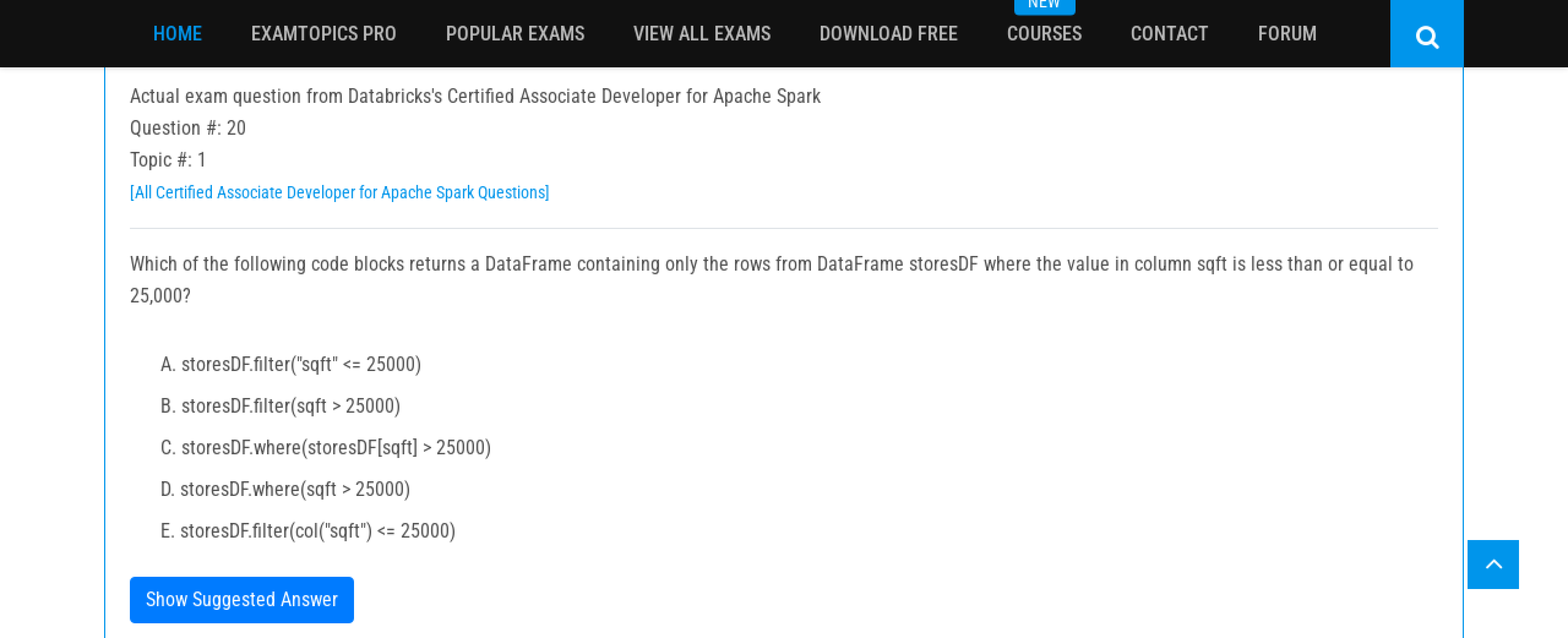
A. The drop() operation only works if one column name is called at a time – there should be two calls in succession like storesDF.drop("sqft").drop("customerSatisfaction").

B. The drop() operation only works if column names are wrapped inside the col() function like storesDF.drop(col(sqft), col(customerSatisfaction)).

C. There is no drop() operation for storesDF.

D. The sqft and customerSatisfaction column names should be quoted like "sqft" and "customerSatisfaction".

E. The sqft and customerSatisfaction column names should be subset from the DataFrame storesDF like storesDF."sqft" and storesDF."customerSatisfaction".



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 20

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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?

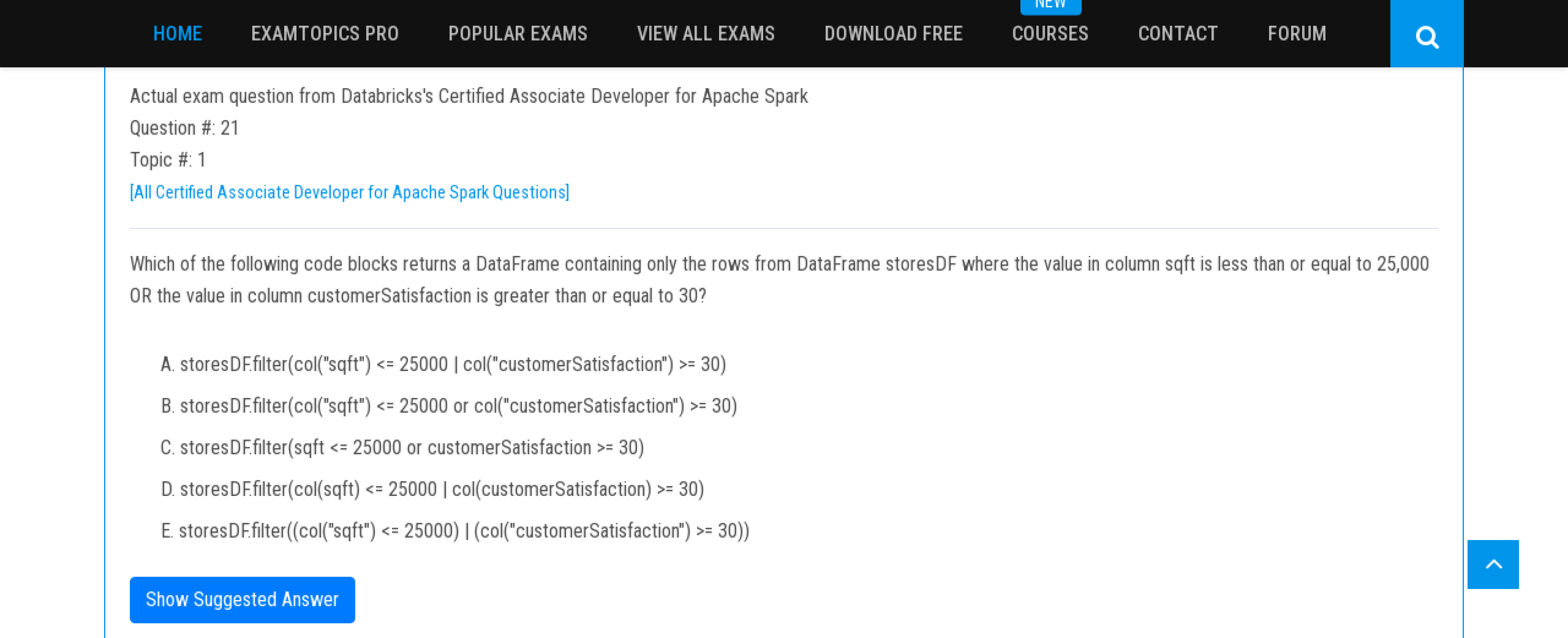
A. storesDF.filter("sqft" <= 25000)

B. storesDF.filter(sqft > 25000)

C. storesDF.where(storesDF[sqft] > 25000)

D. storesDF.where(sqft > 25000)

E. storesDF.filter(col("sqft") <= 25000)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 21

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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?

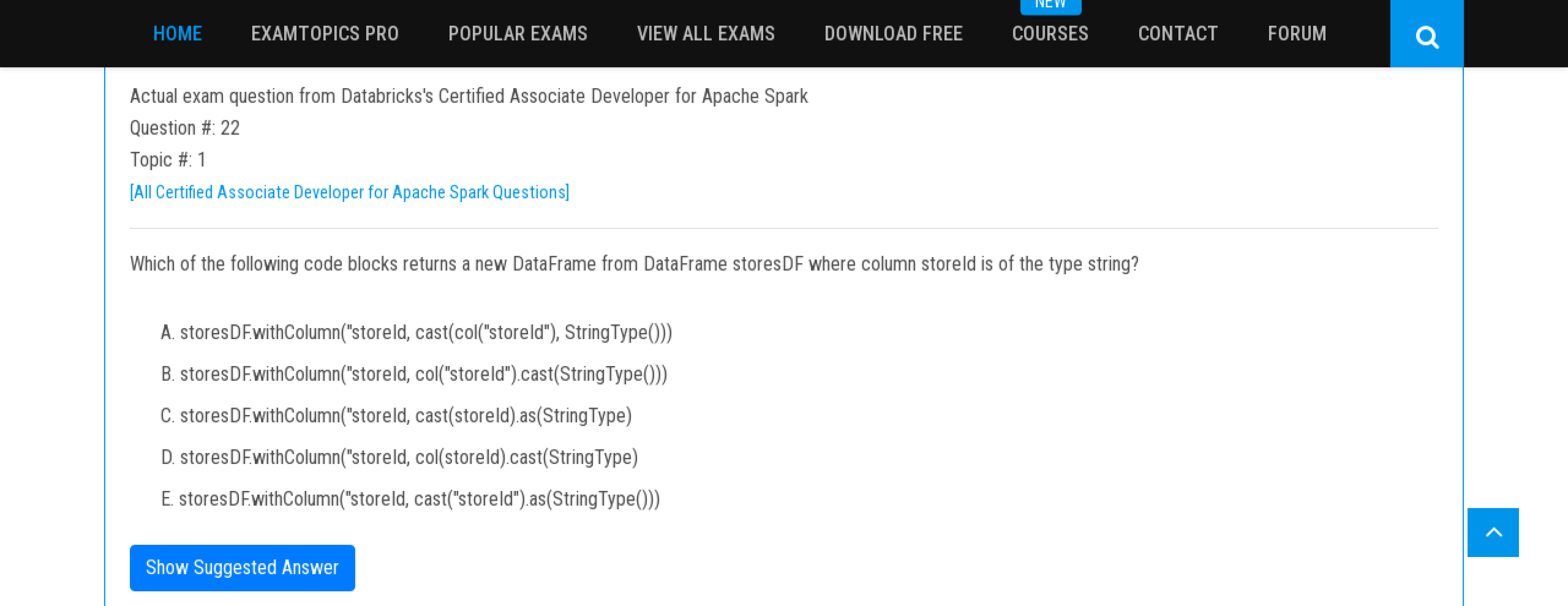
A. storesDF.filter(col("sqft") <= 25000 | col("customerSatisfaction") >= 30)

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))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 22

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

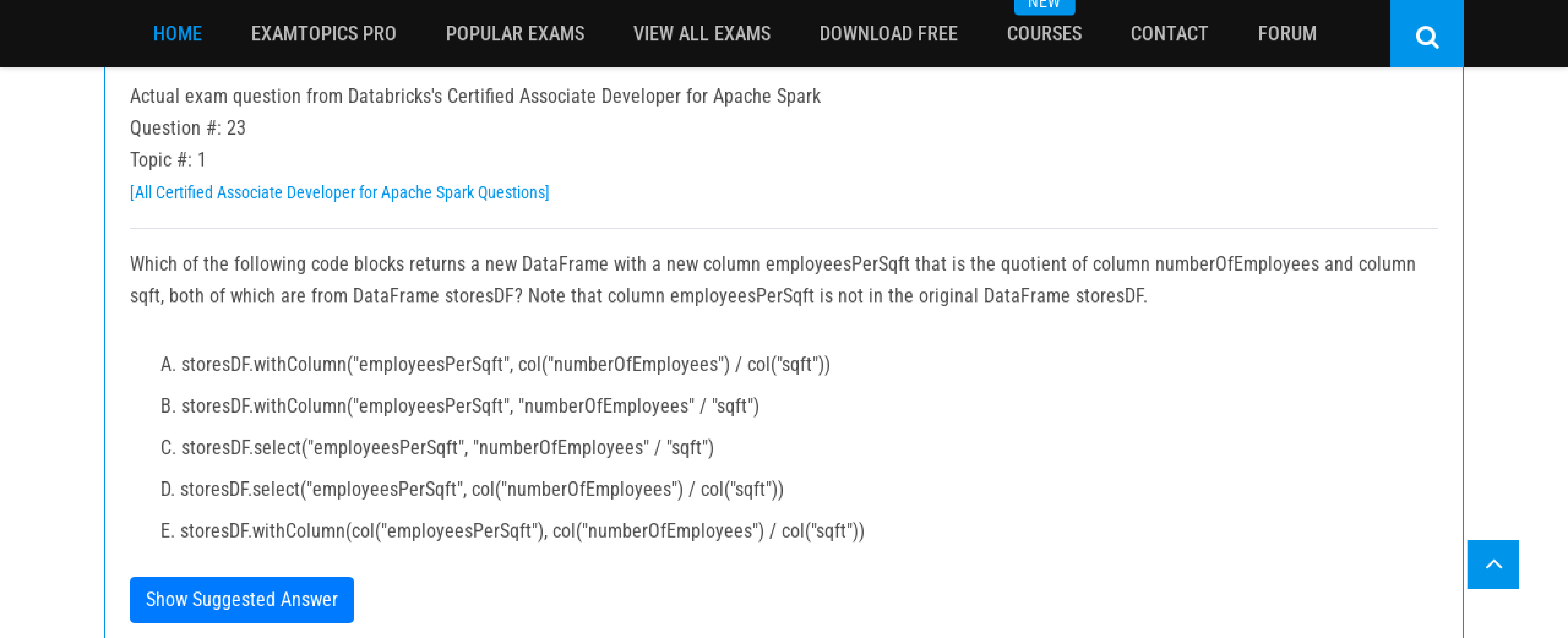
A. storesDF.withColumn("storeId, cast(col("storeId"), StringType()))

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()))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 23

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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"))

Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 24
Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

Show Suggested Answer

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

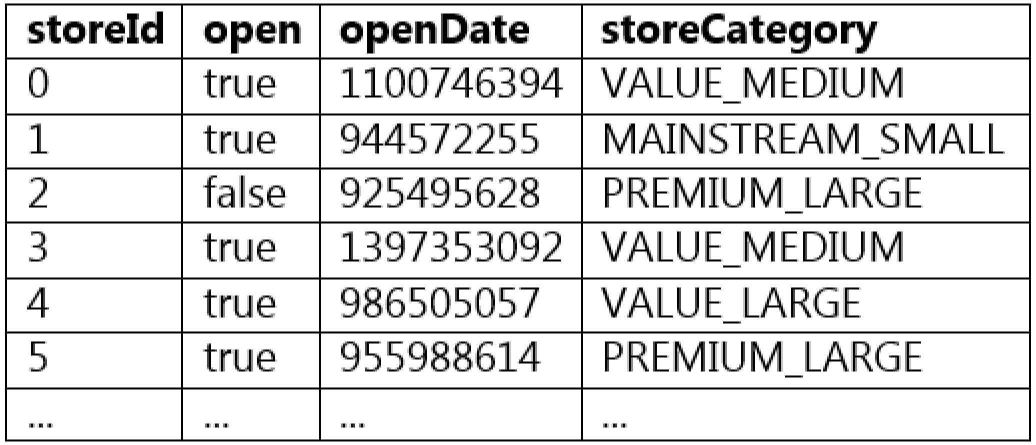
Question #: 25

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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]))

Show Suggested Answer

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

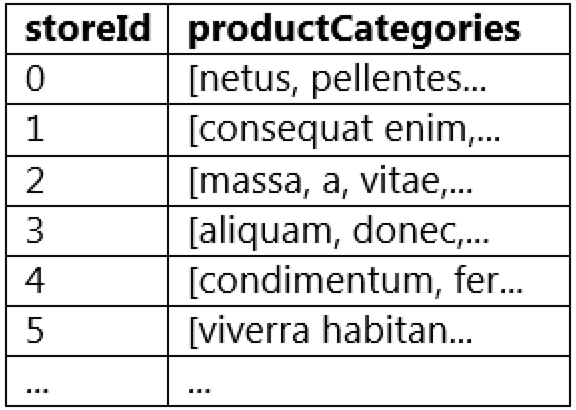
Question #: 26

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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")))

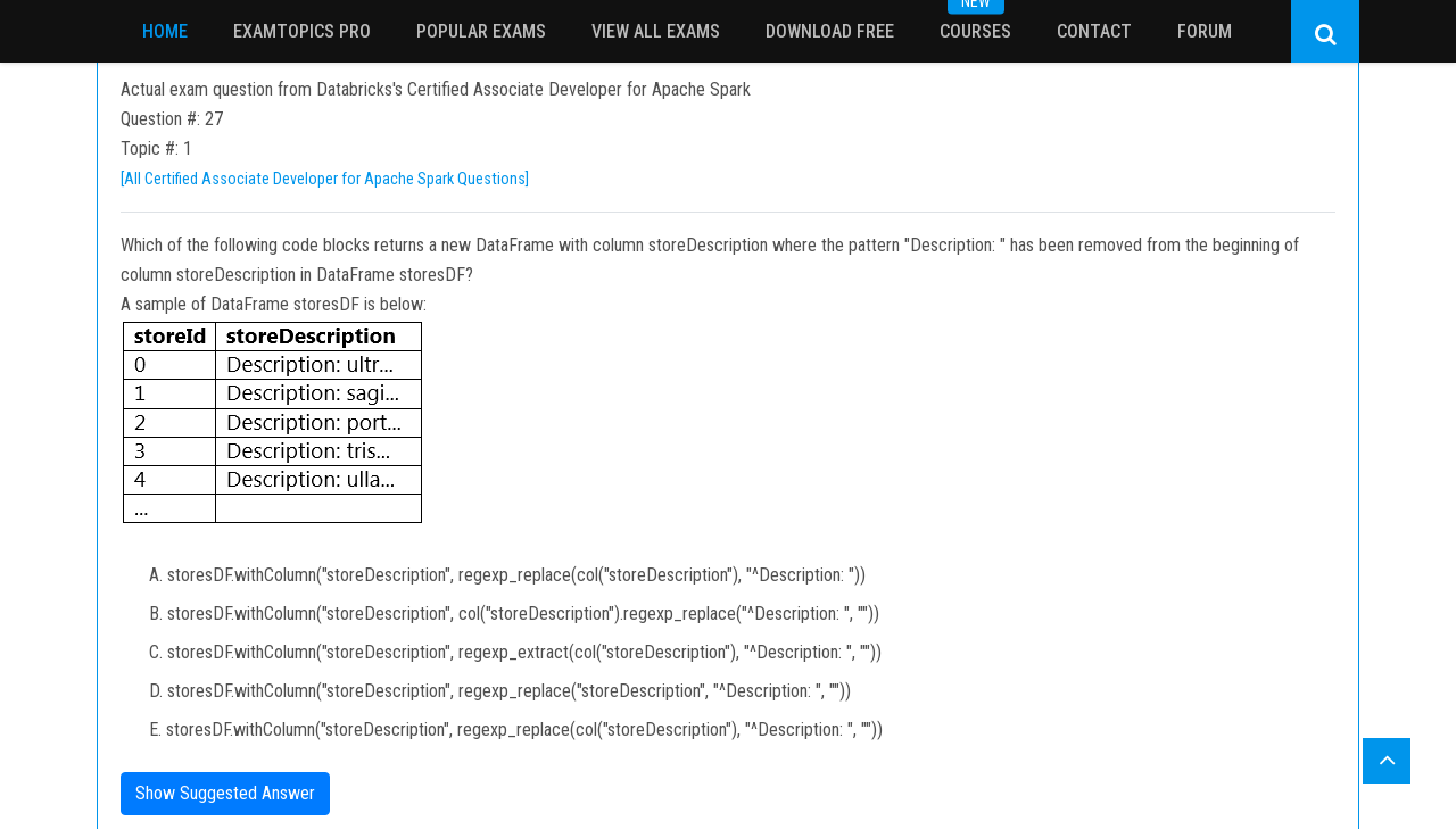
B. storesDF.withColumn("productCategories", split(col("productCategories")))

C. storesDF.withColumn("productCategories", col("productCategories").explode())

D. storesDF.withColumn("productCategories", col("productCategories").split())

E. storesDF.withColumn("productCategories", explode("productCategories"))

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 27

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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...
...	

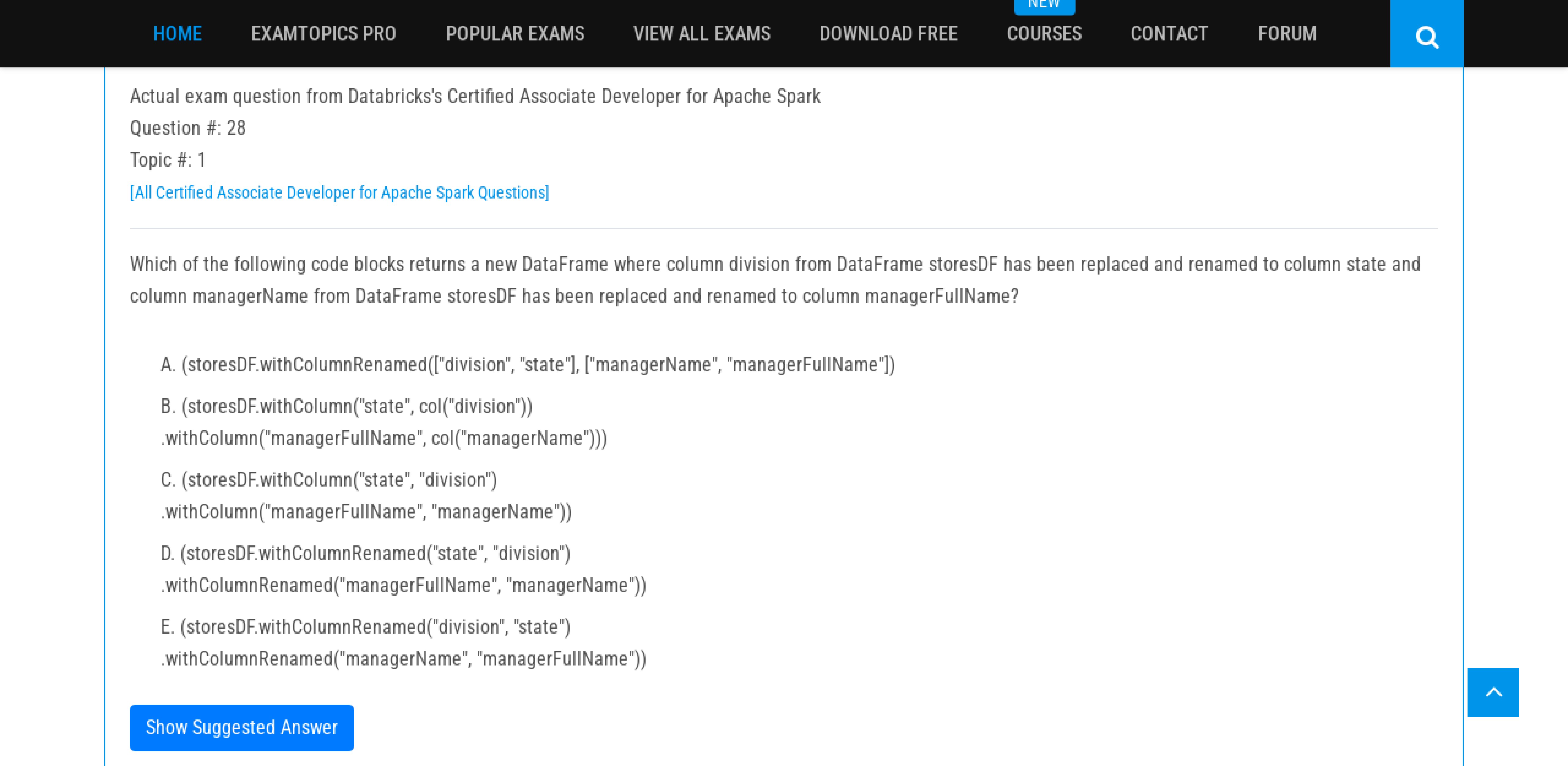
A. storesDF.withColumn("storeDescription", regexp_replace(col("storeDescription"), "^Description: "))

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: ", ""))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 28

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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?

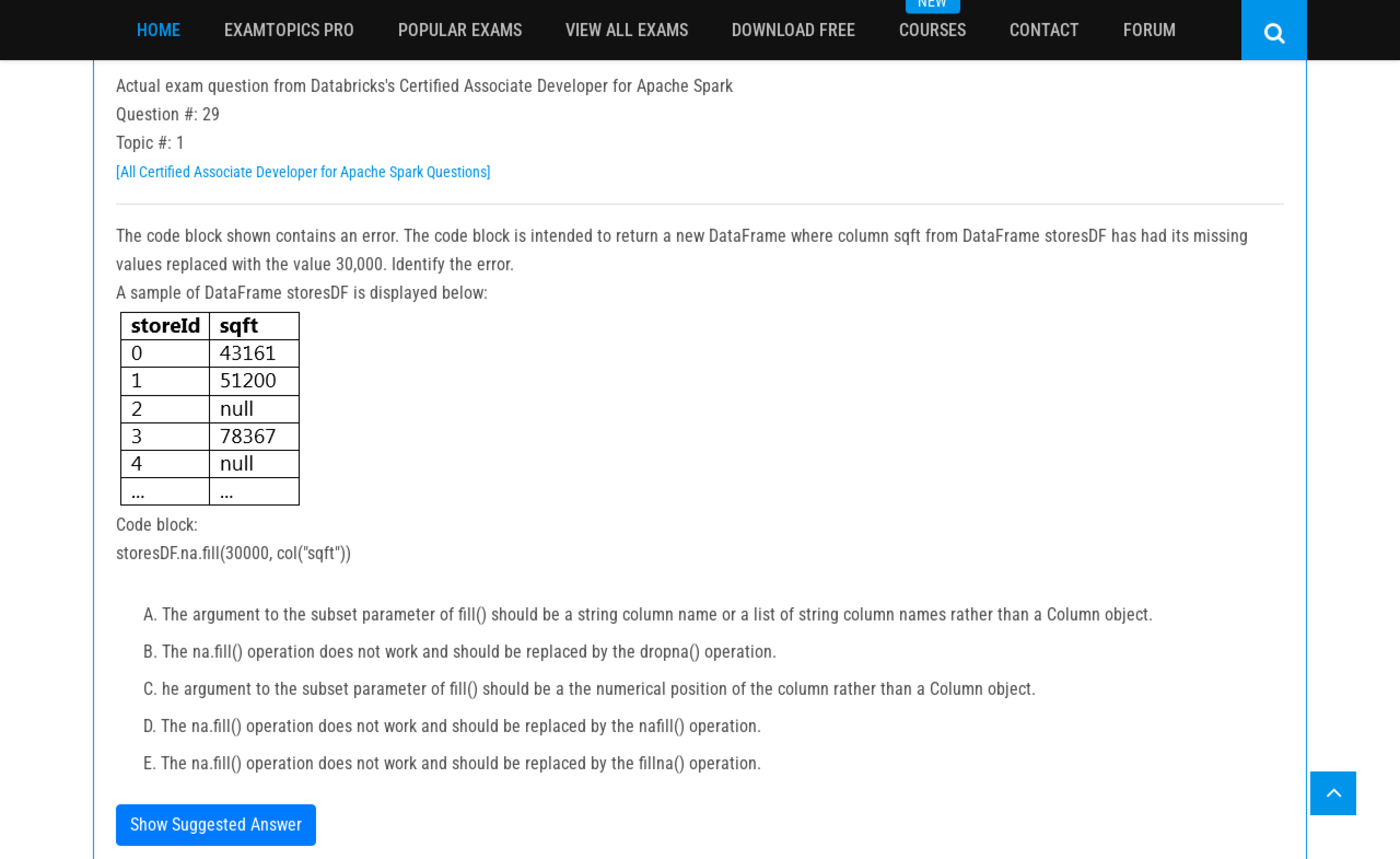
A. (storesDF.withColumnRenamed(["division", "state"], ["managerName", "managerFullName"])

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"))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 29

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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. he argument to the subset parameter of fill() should be a 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 nafill() operation.

E. The na.fill() operation does not work and should be replaced by the fillna() operation.

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 30

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

A. DataFrame.dropDuplicates()

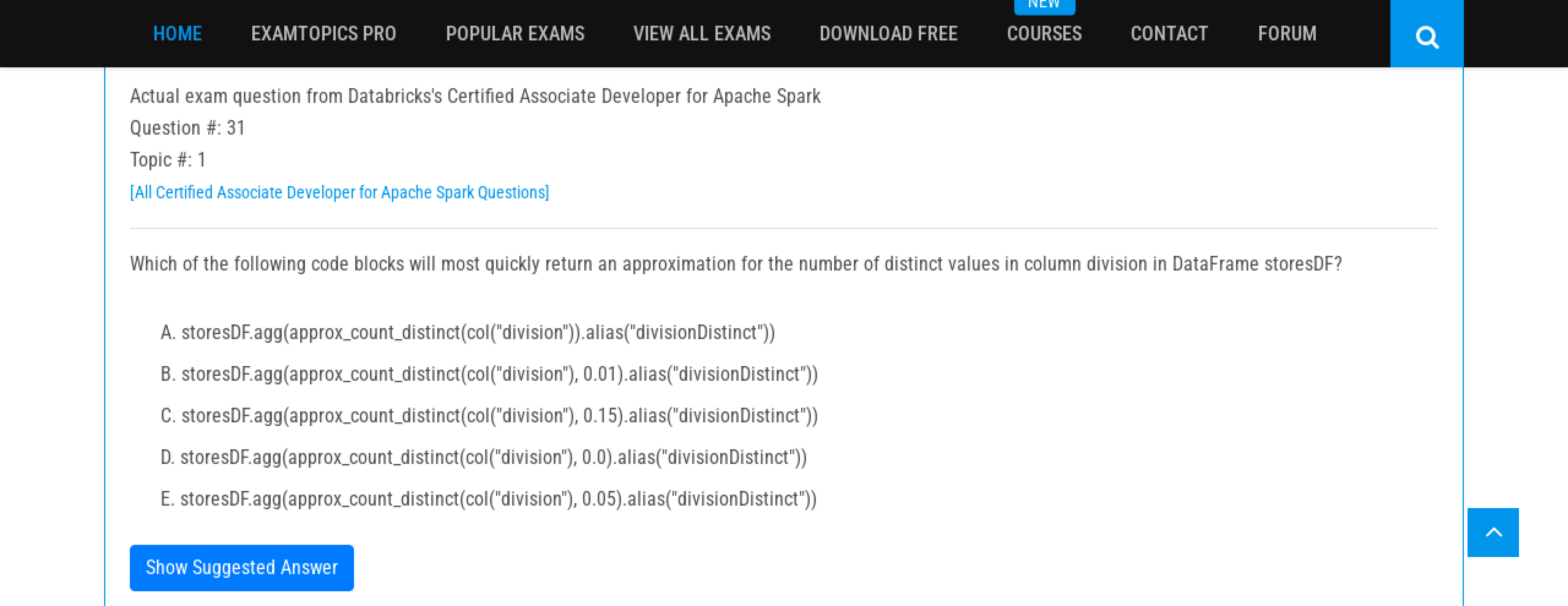
B. DataFrame.distinct()

C. DataFrame.drop_duplicates()

D. DataFrame.drop_duplicates(subset = None)

E. DataFrame.drop_duplicates(subset = "all")

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 31

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

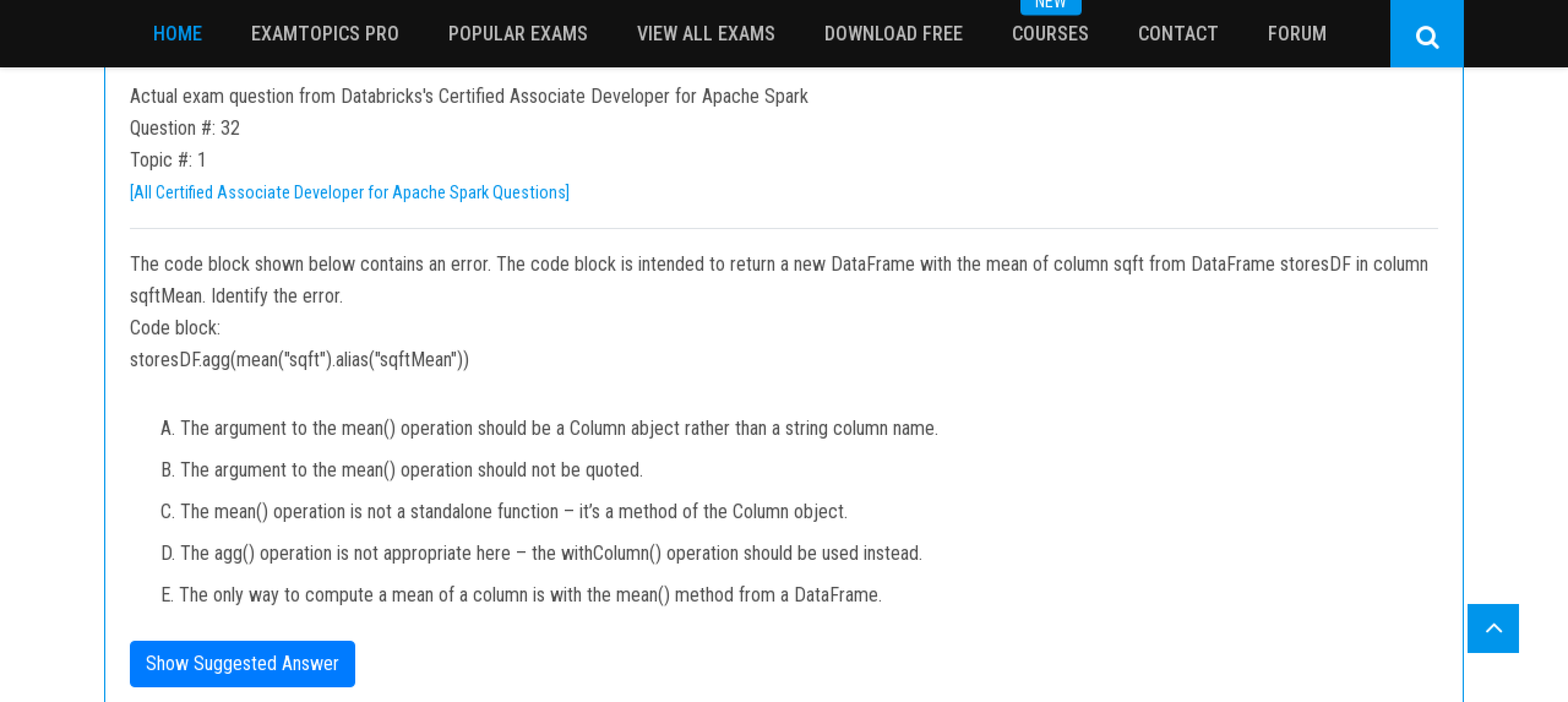
A. storesDF.agg(approx_count_distinct(col("division")).alias("divisionDistinct"))

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"))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 32

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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"))
```

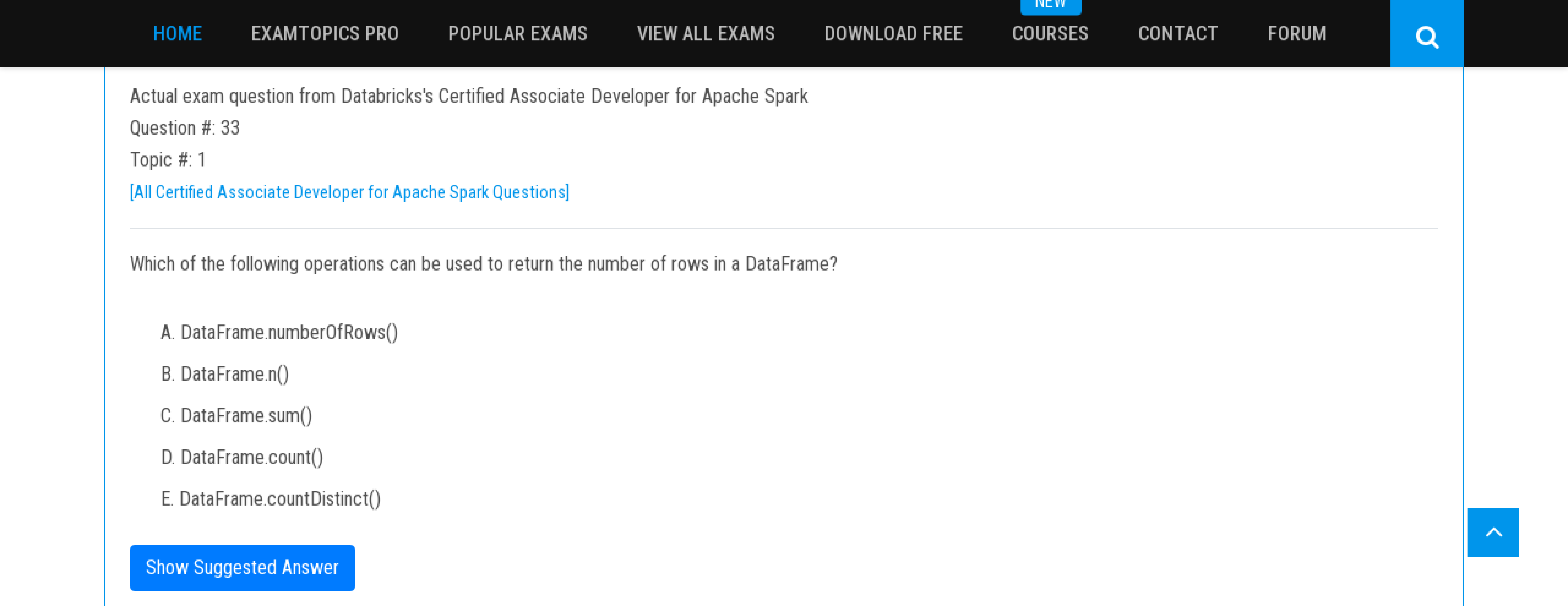
A. The argument to the mean() operation should be a Column abject rather than a string column name.

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.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 33

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

A. DataFrame.numberOfRows()

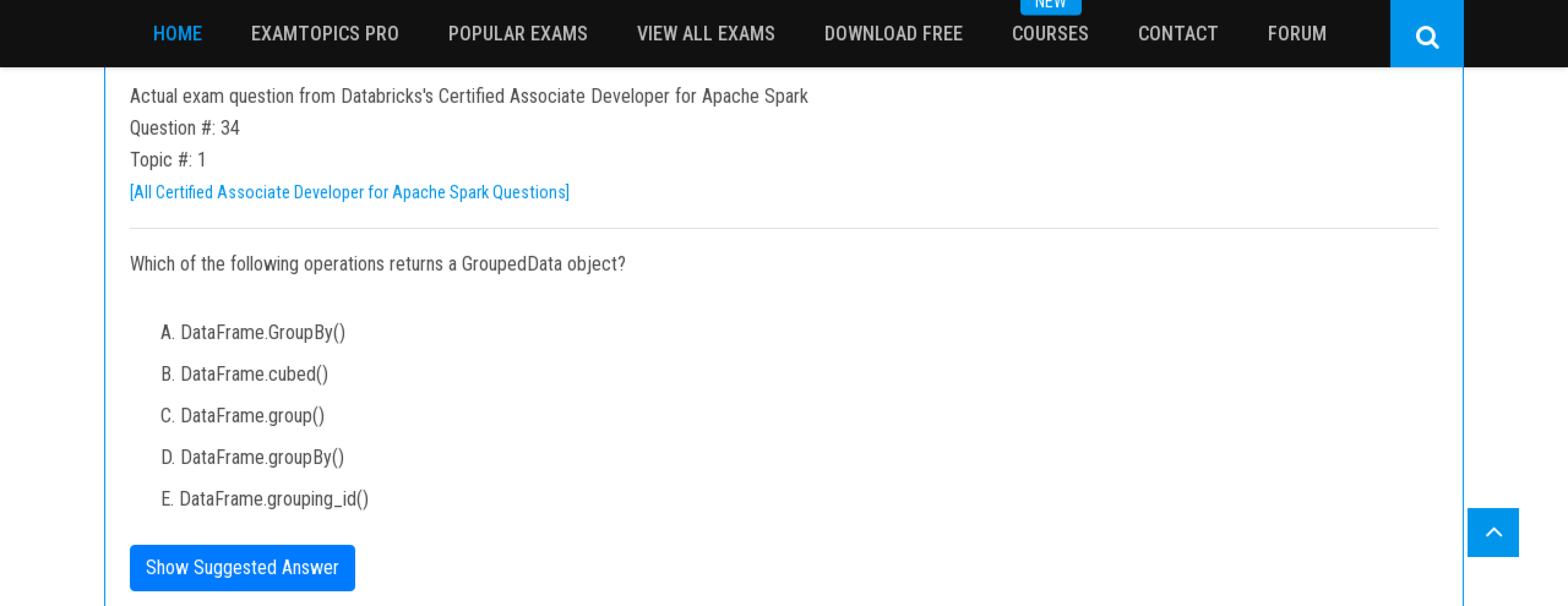
B. DataFrame.n()

C. DataFrame.sum()

D. DataFrame.count()

E. DataFrame.countDistinct()

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 34

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following operations returns a GroupedData object?

A. DataFrame.GroupBy()

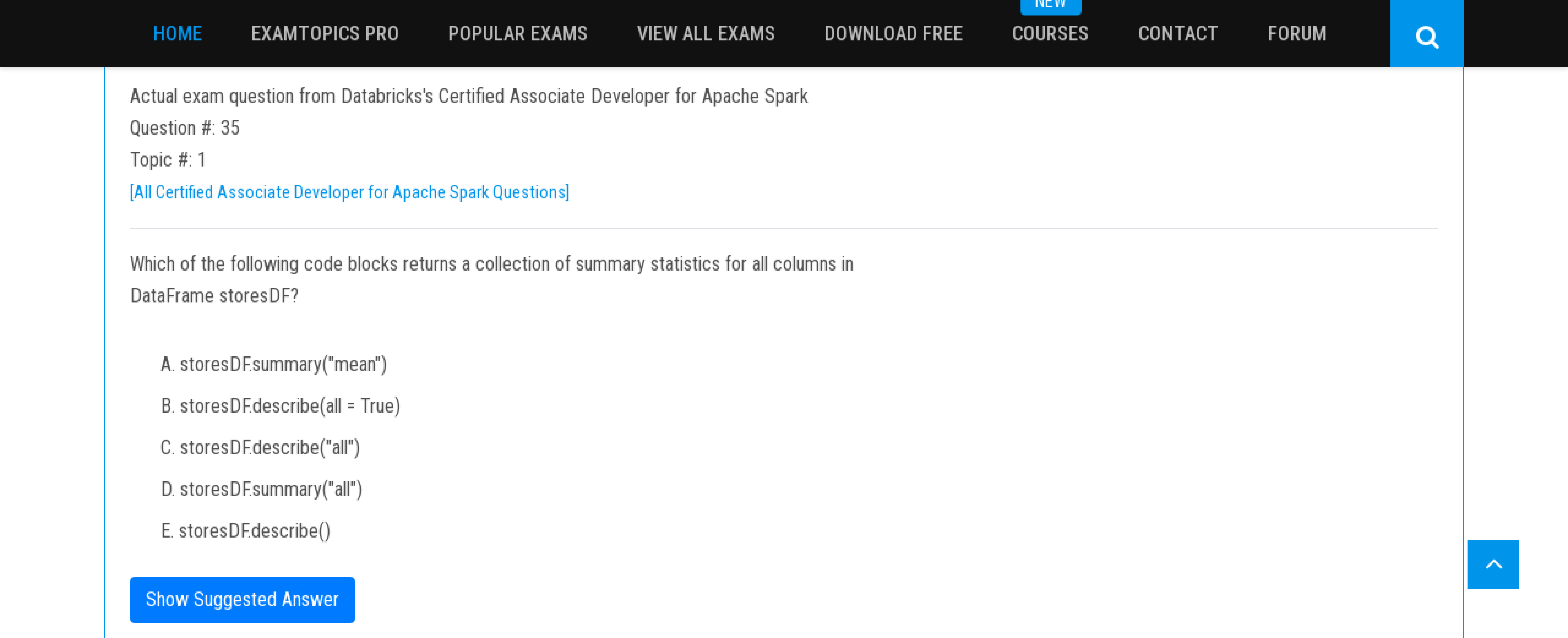
B. DataFrame.cubed()

C. DataFrame.group()

D. DataFrame.groupBy()

E. DataFrame.grouping_id()

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 35

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

A. storesDF.summary("mean")

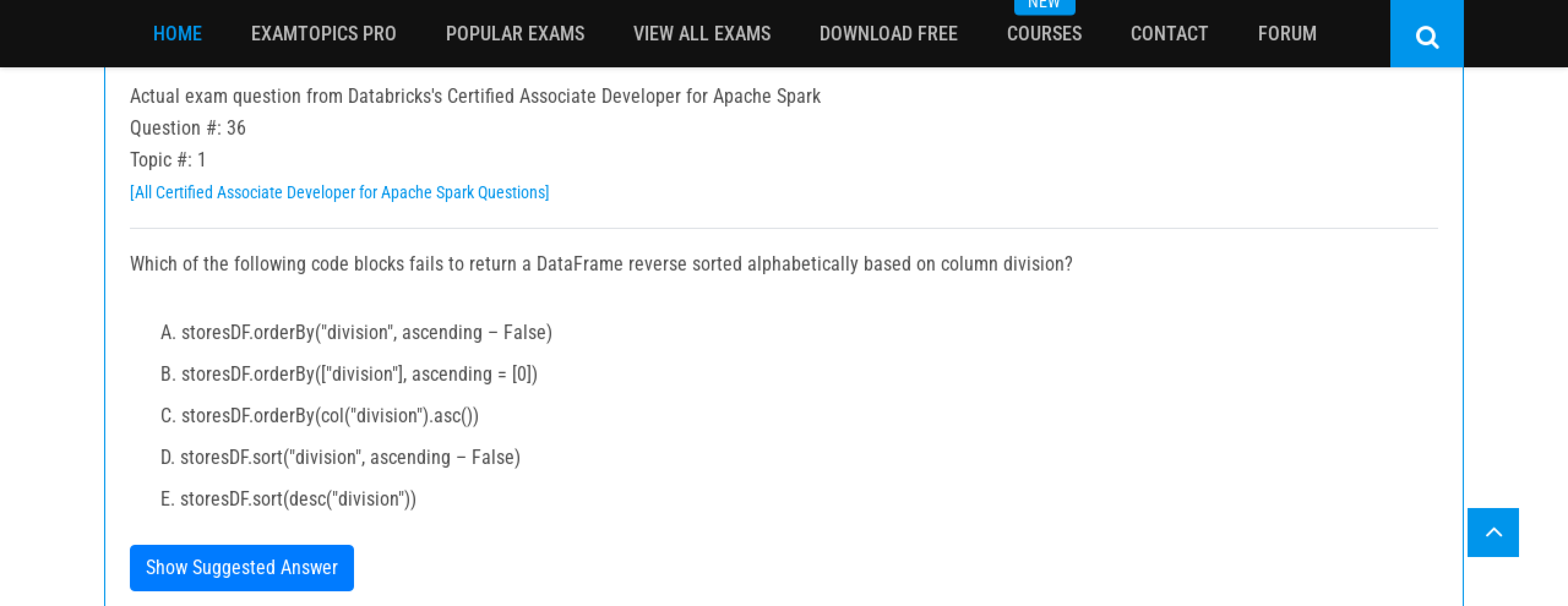
B. storesDF.describe(all = True)

C. storesDF.describe("all")

D. storesDF.summary("all")

E. storesDF.describe()

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 36

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

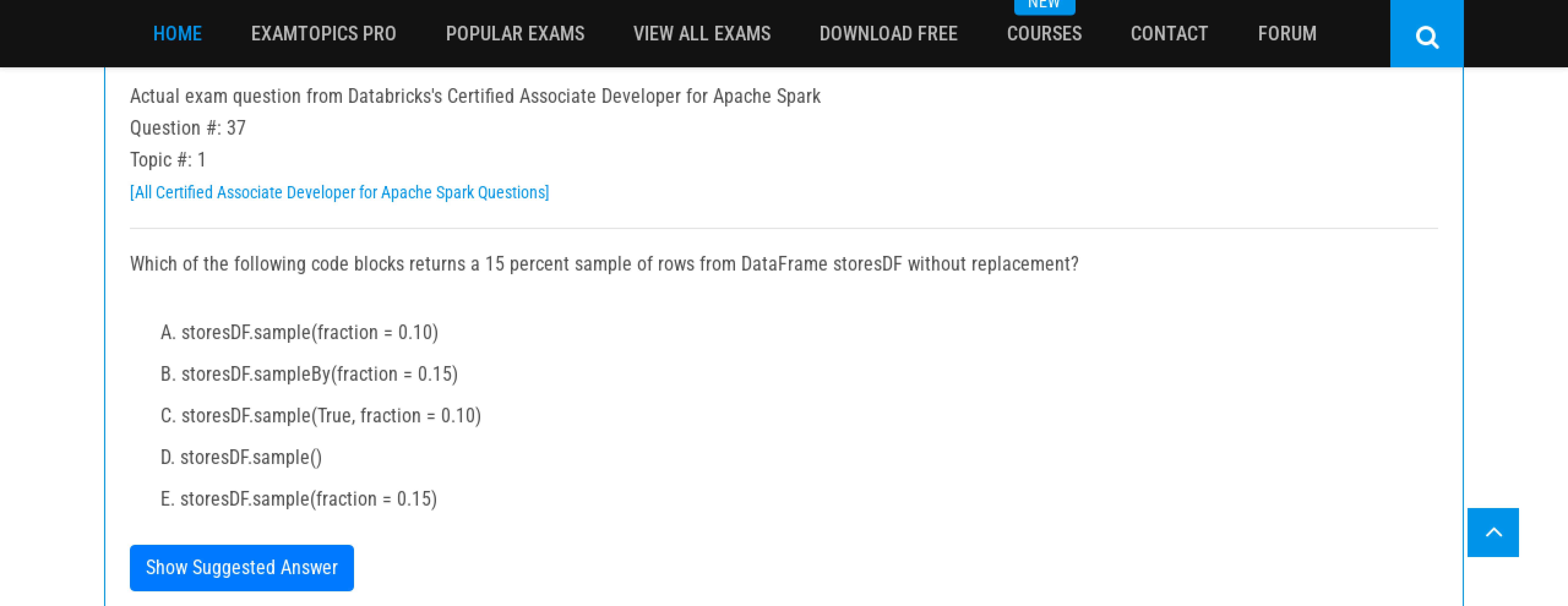
A. storesDF.orderBy("division", ascending – False)

B. storesDF.orderBy(["division"], ascending = [0])

C. storesDF.orderBy(col("division").asc())

D. storesDF.sort("division", ascending – False)

E. storesDF.sort(desc("division"))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 37

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

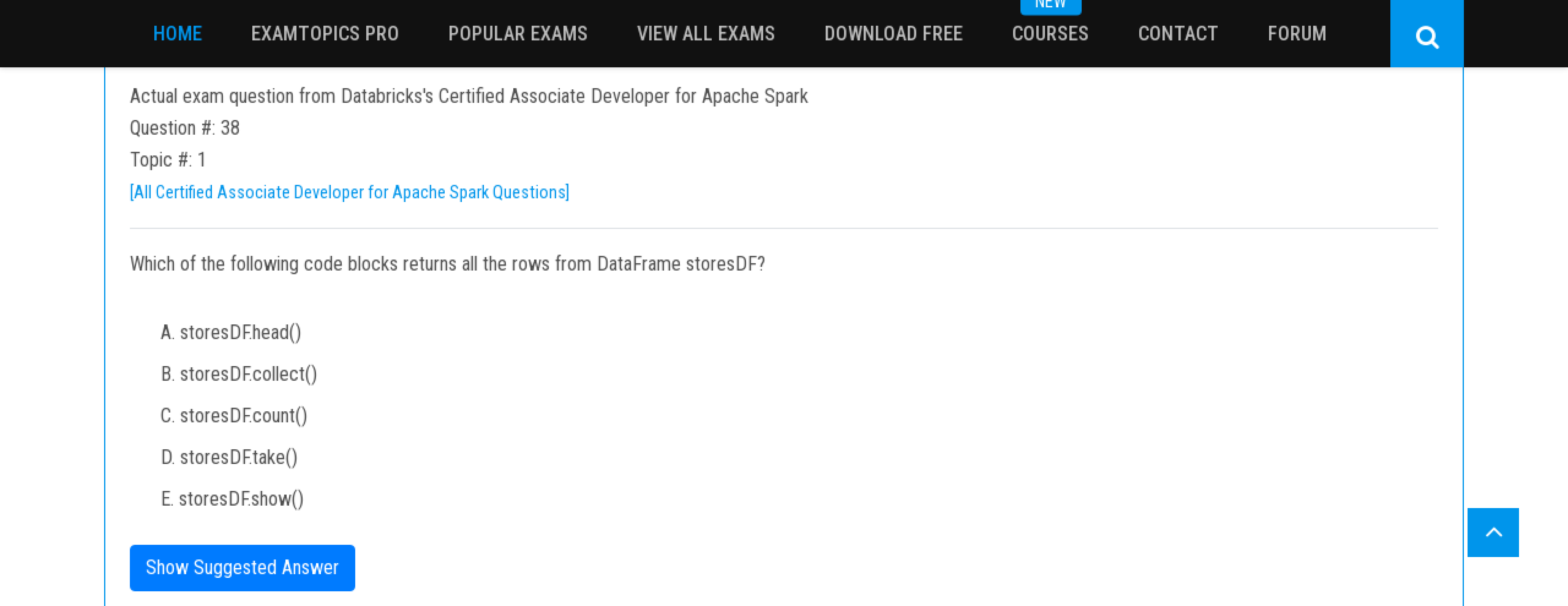
A. storesDF.sample(fraction = 0.10)

B. storesDF.sampleBy(fraction = 0.15)

C. storesDF.sample(True, fraction = 0.10)

D. storesDF.sample()

E. storesDF.sample(fraction = 0.15)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 38

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

A. storesDF.head()

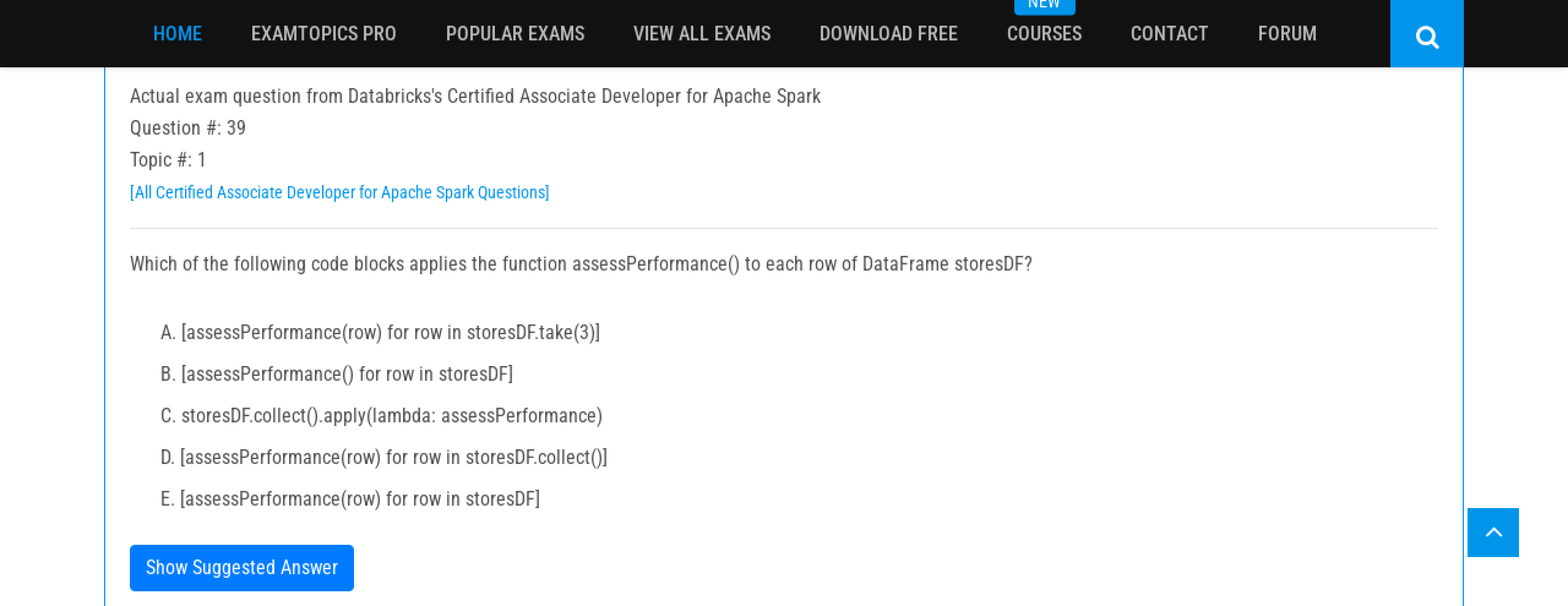
B. storesDF.collect()

C. storesDF.count()

D. storesDF.take()

E. storesDF.show()

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 39

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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)]

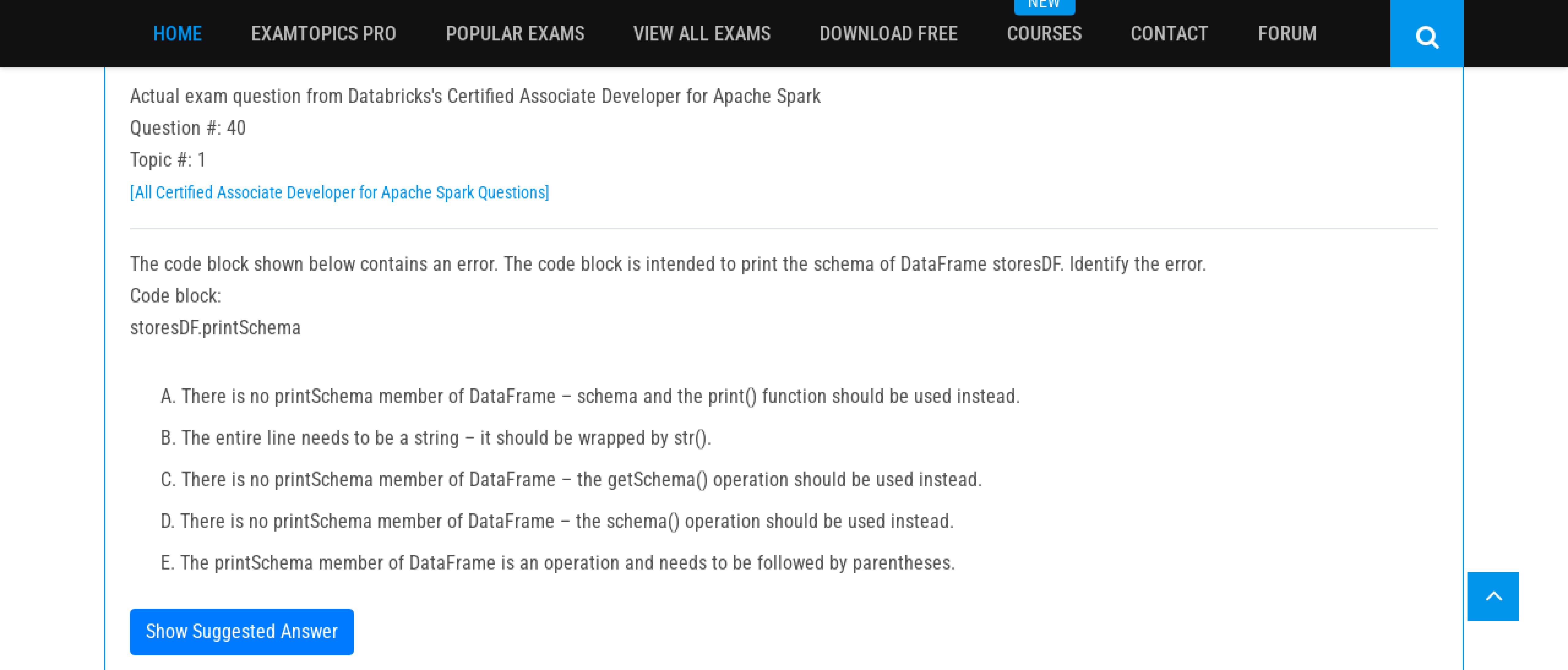
B. [assessPerformance() for row in storesDF]

C. storesDF.collect().apply(lambda: assessPerformance)

D. [assessPerformance(row) for row in storesDF.collect()]

E. [assessPerformance(row) for row in storesDF]

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 40

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

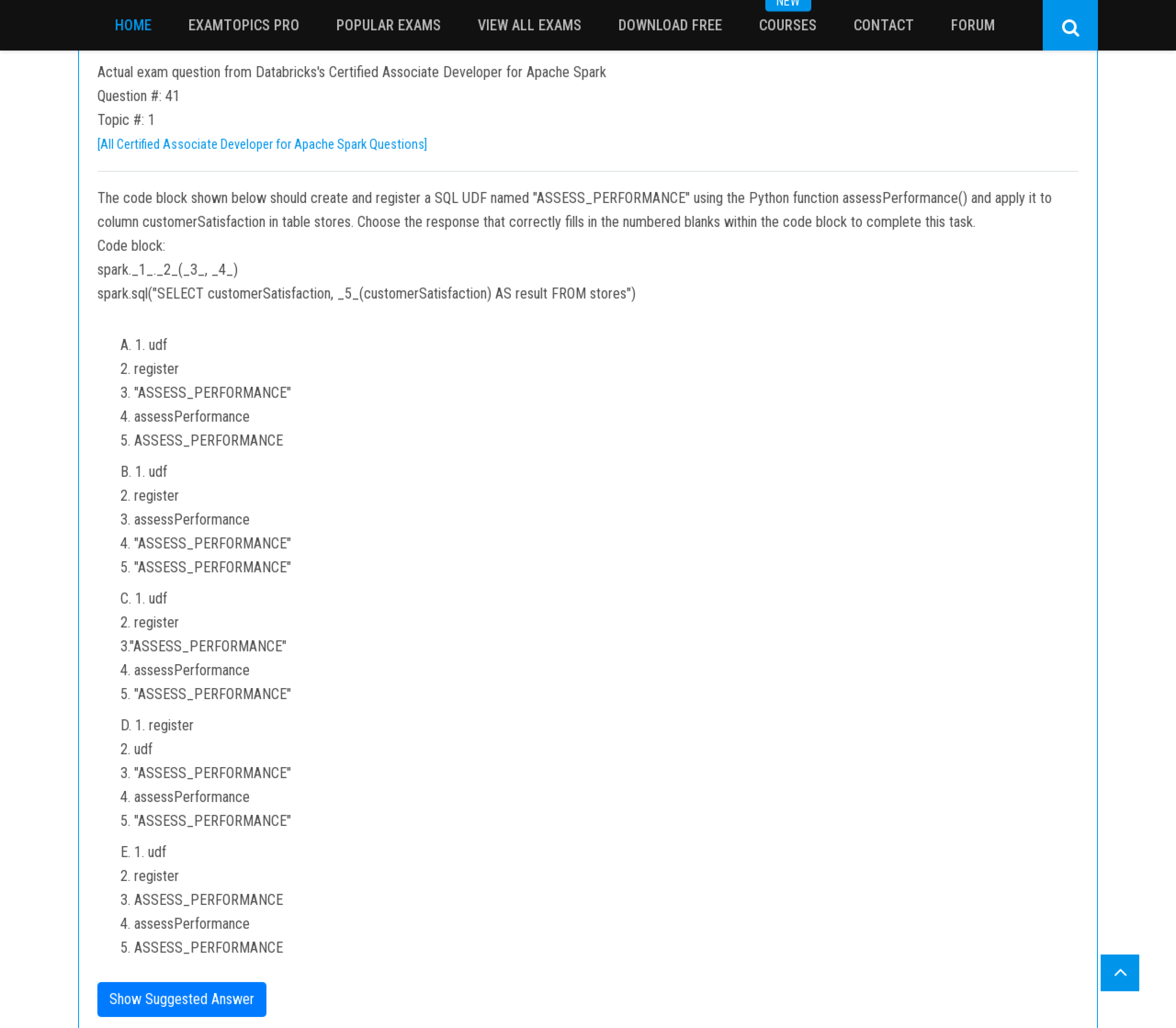
A. There is no printSchema member of DataFrame – schema and the print() function should be used instead.

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.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 41
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

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")
```

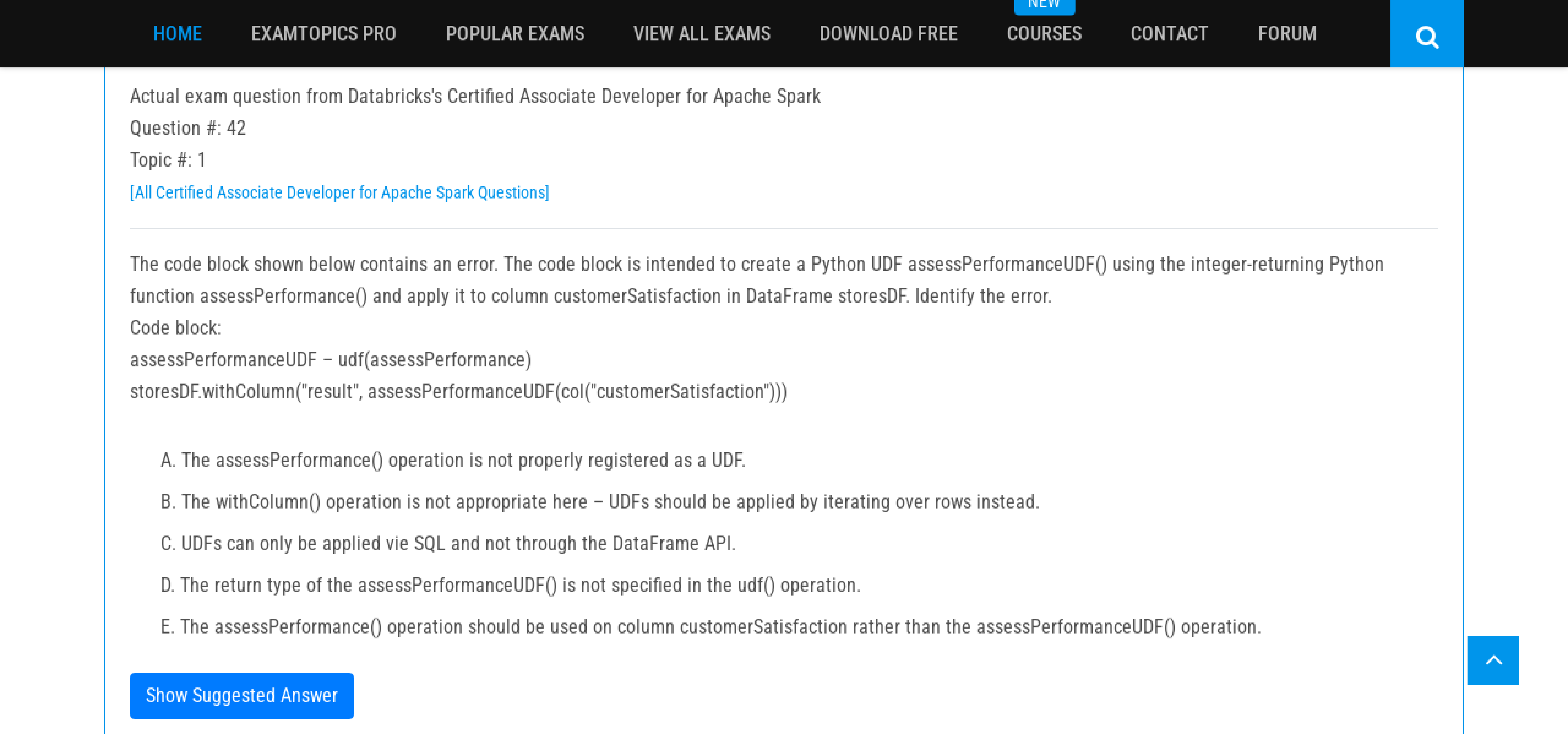
A. 1. udf
2. register
3. "ASSESS_PERFORMANCE"
4. assessPerformance
5. ASSESS_PERFORMANCE

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



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 42

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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")))
```

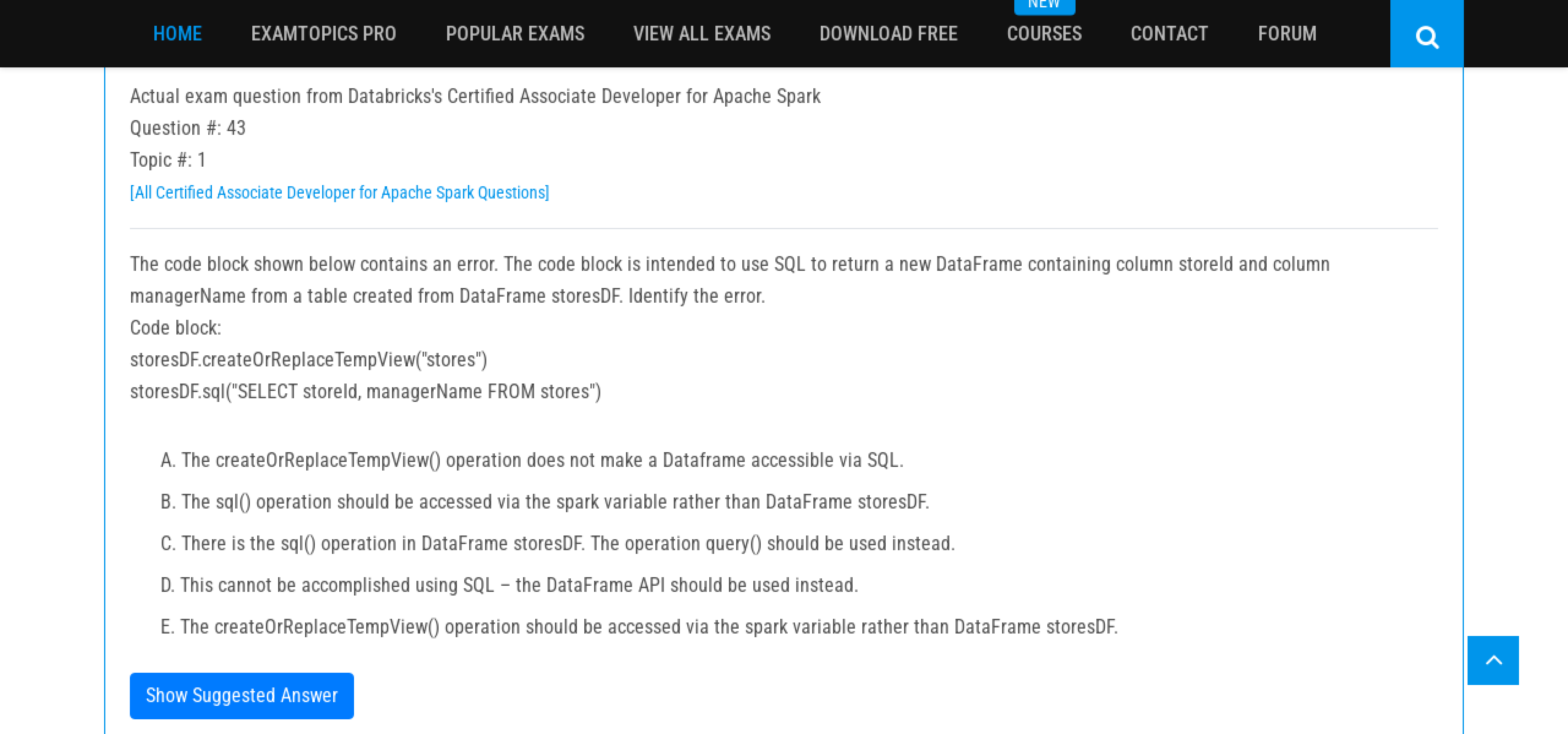
A. The assessPerformance() operation is not properly registered as a UDF.

B. The withColumn() operation is not appropriate here – UDFs should be applied by iterating over rows instead.

C. UDFs can only be applied vie 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.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 43

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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.

Show Suggested Answer

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 44

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should create a single-column DataFrame from Python list years which is made up of integers. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

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

A. 1. spark
2. createDataFrame
3. years
4. IntegerType

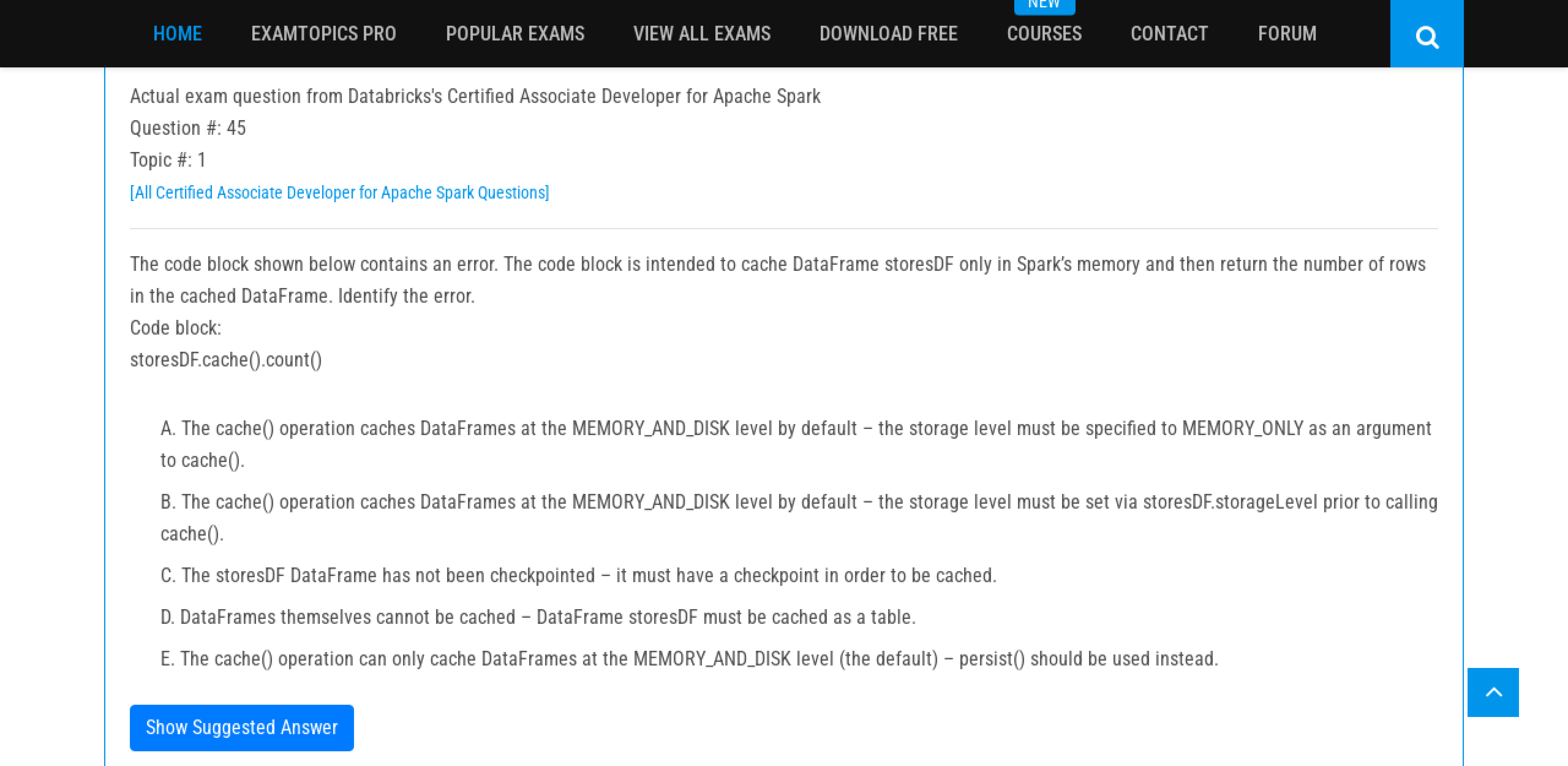
B. 1. DataFrame
2. create
3. [years]
4. IntegerType

C. 1. spark
2. createDataFrame
3. [years]
4. IntegertType

D. 1. spark
2. createDataFrame
3. [years]
4. IntegertType()

E. 1. spark
2. createDataFrame
3. years
4. IntegertType()

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 45
Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to cache DataFrame storesDF only in Spark's memory and then return the number of rows in the cached DataFrame. Identify the error.
Code block:

```
storesDF.cache().count()
```

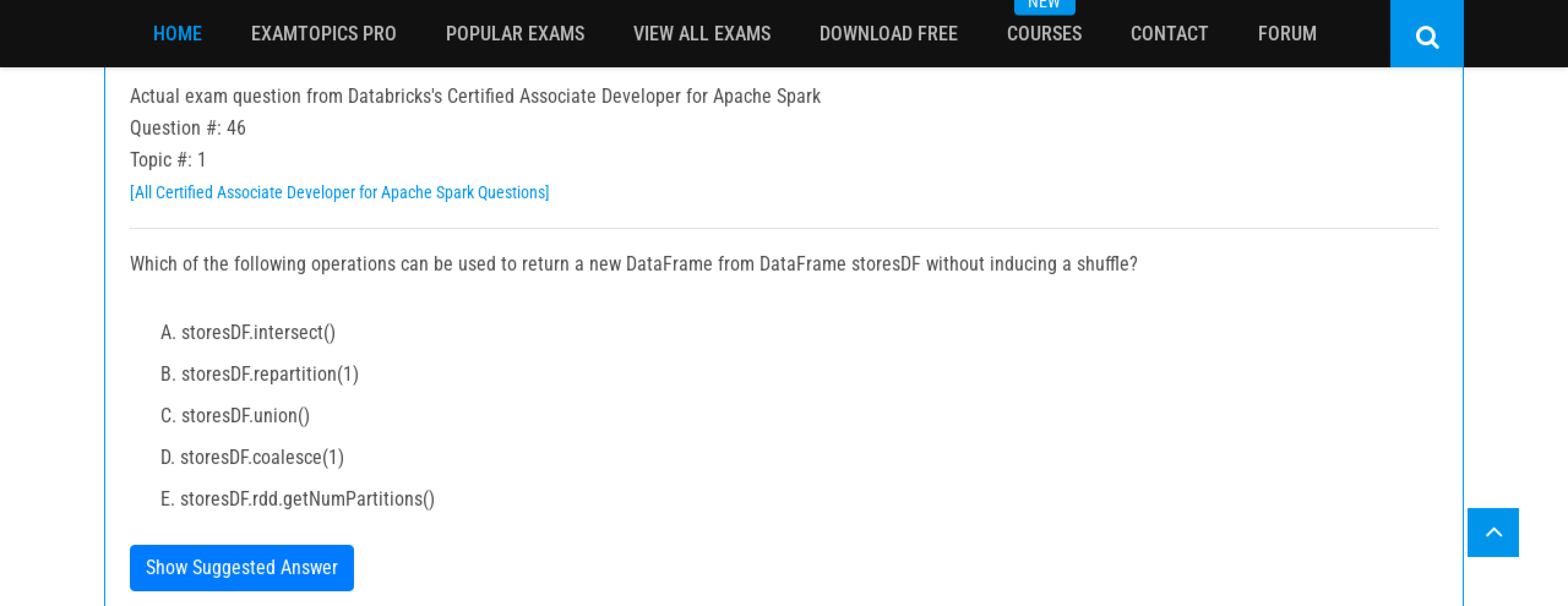
A. The cache() operation caches DataFrames at the MEMORY_AND_DISK level by default – the storage level must be specified to MEMORY_ONLY as an argument to cache().

B. The cache() operation caches DataFrames at the MEMORY_AND_DISK level by default – the storage level must be set via storesDF.storageLevel prior to calling cache().

C. The storesDF DataFrame has not been checkpointed – it must have a checkpoint in order to be cached.

D. DataFrames themselves cannot be cached – DataFrame storesDF must be cached as a table.

E. The cache() operation can only cache DataFrames at the MEMORY_AND_DISK level (the default) – persist() should be used instead.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 46

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following operations can be used to return a new DataFrame from DataFrame storesDF without inducing a shuffle?

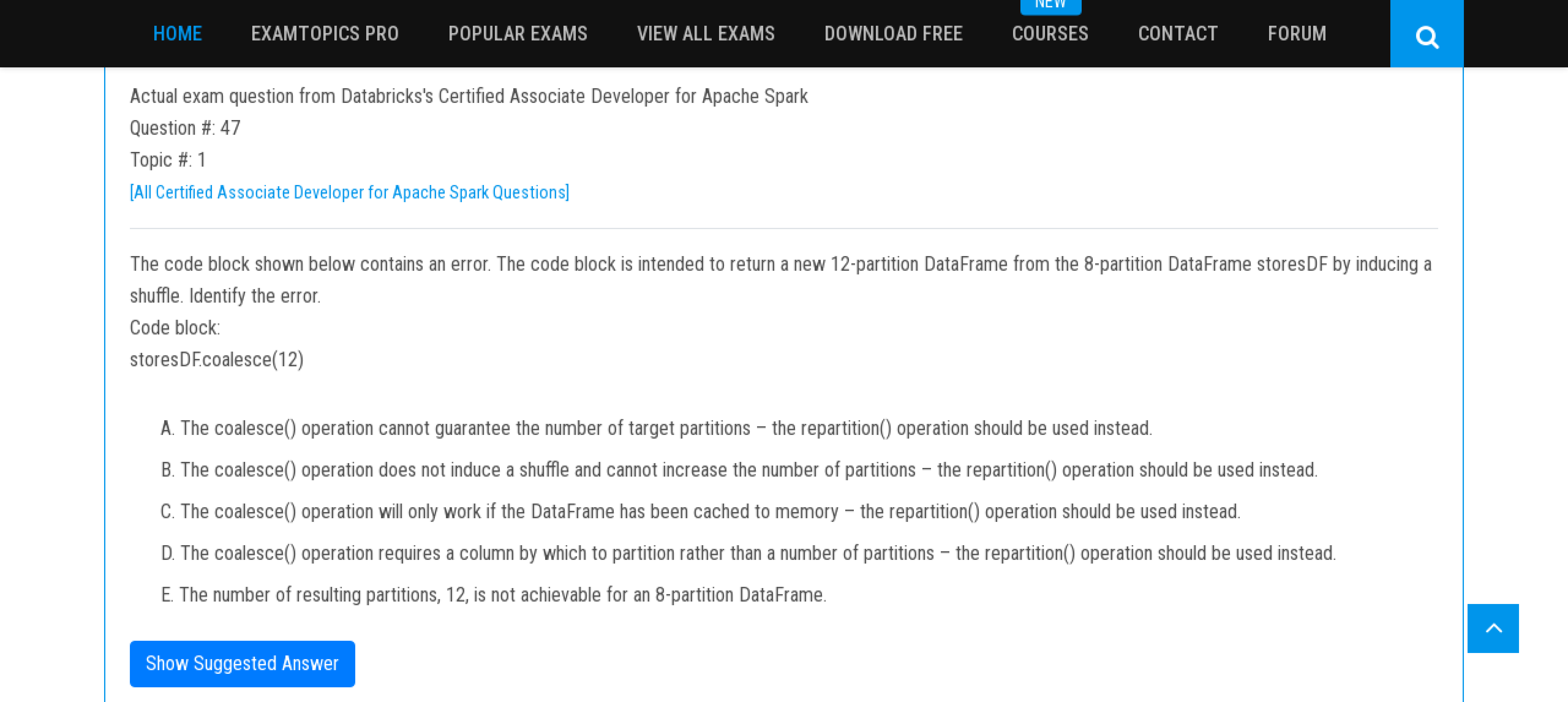
A. storesDF.intersect()

B. storesDF.repartition(1)

C. storesDF.union()

D. storesDF.coalesce(1)

E. storesDF.rdd.getNumPartitions()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 47

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to return a new 12-partition DataFrame from the 8-partition DataFrame storesDF by inducing a shuffle. Identify the error.

Code block:

```
storesDF.coalesce(12)
```

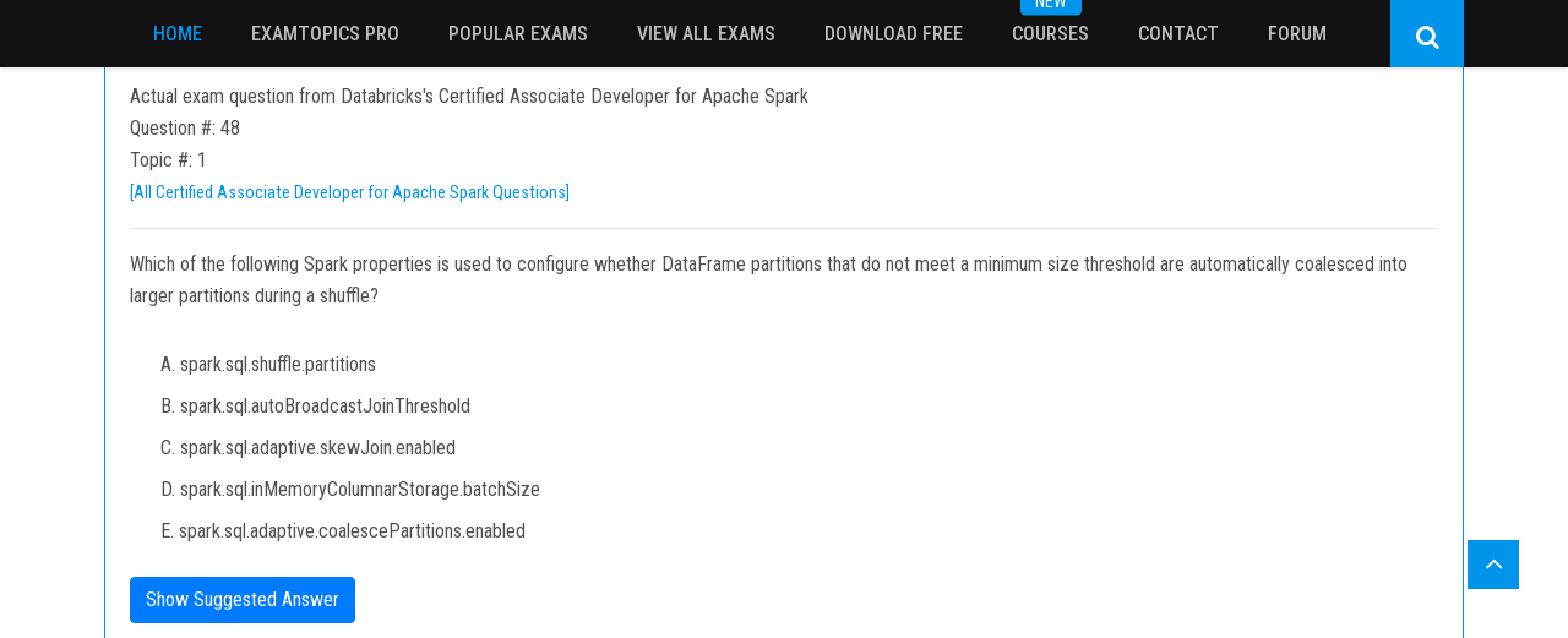
A. The coalesce() operation cannot guarantee the number of target partitions – the repartition() operation should be used instead.

B. The coalesce() operation does not induce a shuffle and cannot increase the number of partitions – the repartition() operation should be used instead.

C. The coalesce() operation will only work if the DataFrame has been cached to memory – the repartition() operation should be used instead.

D. The coalesce() operation requires a column by which to partition rather than a number of partitions – the repartition() operation should be used instead.

E. The number of resulting partitions, 12, is not achievable for an 8-partition DataFrame.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 48

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following Spark properties is used to configure whether DataFrame partitions that do not meet a minimum size threshold are automatically coalesced into larger partitions during a shuffle?

A. spark.sql.shuffle.partitions

B. spark.sql.autoBroadcastJoinThreshold

C. spark.sql.adaptive.skewJoin.enabled

D. spark.sql.inMemoryColumnarStorage.batchSize

E. spark.sql.adaptive.coalescePartitions.enabled

Show Suggested Answer

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 49

Topic #: 1

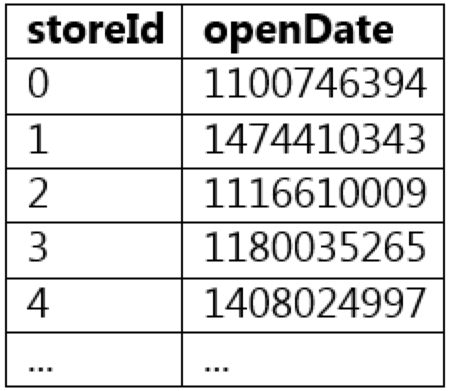
[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to return a DataFrame containing a column openDateString, a string representation of Java's SimpleDateFormat. Identify the error.

Note that column openDate is of type integer and represents a date in the UNIX epoch format – the number of seconds since midnight on January 1st, 1970.

An example of Java's SimpleDateFormat is "Sunday, Dec 4, 2008 1:05 PM".

A sample of storesDF is displayed below:



storeId	openDate
0	1100746394
1	1474410343
2	1116610009
3	1180035265
4	1408024997
...	...

Code block:

```
storesDF.withColumn("openDateString", from_unixtime(col("openDate"), "EEE, MMM d, yyyy h:mm a", TimestampType()))
```

A. The from_unixtime() operation only accepts two parameters – the TimestampTime() arguments not necessary.

B. The from_unixtime() operation only works if column openDate is of type long rather than integer – column openDate must first be converted.

C. The second argument to from_unixtime() is not correct – it should be a variant of TimestampType() rather than a string.

D. The from_unixtime() operation automatically places the input column in java's SimpleDateFormat – there is no need for a second or third argument.

E. The column openDate must first be converted to a timestamp, and then the Date() function can be used to reformat to java's SimpleDateFormat.

Show Suggested Answer

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 50

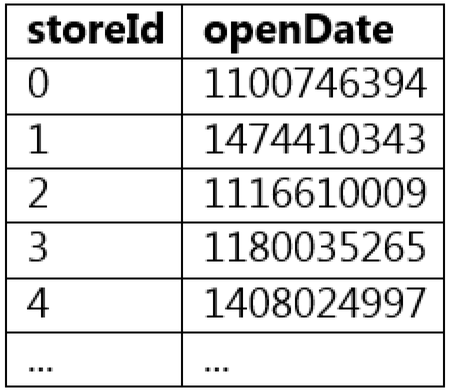
Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns a DataFrame containing a column dayOfYear, an integer representation of the day of the year from column openDate from DataFrame storesDF?

Note that column openDate is of type integer and represents a date in the UNIX epoch format – the number of seconds since midnight on January 1st, 1970.

A sample of storesDF is displayed below:



storeId	openDate
0	1100746394
1	1474410343
2	1116610009
3	1180035265
4	1408024997
...	...

A. (storesDF.withColumn("openTimestamp", col("openDate").cast("Timestamp"))
. withColumn("dayOfYear", dayofyear(col("openTimestamp"))))

B. storesDF.withColumn("dayOfYear", get dayofyear(col("openDate")))

C. storesDF.withColumn("dayOfYear", dayofyear(col("openDate")))

D. (storesDF.withColumn("openDateFormat", col("openDate").cast("Date"))
. withColumn("dayOfYear", dayofyear(col("openDateFormat"))))

E. storesDF.withColumn("dayOfYear", substr(col("openDate"), 4, 6))

Show Suggested Answer

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 51

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block intended to return a new DataFrame that is the result of an inner join between DataFrame storesDF and DataFrame employeesDF on column storeId. Identify the error.

Code block:

```
StoresDF.join(employeesDF, "inner", "storeID")
```

A. The key column storeID needs to be wrapped in the col() operation.

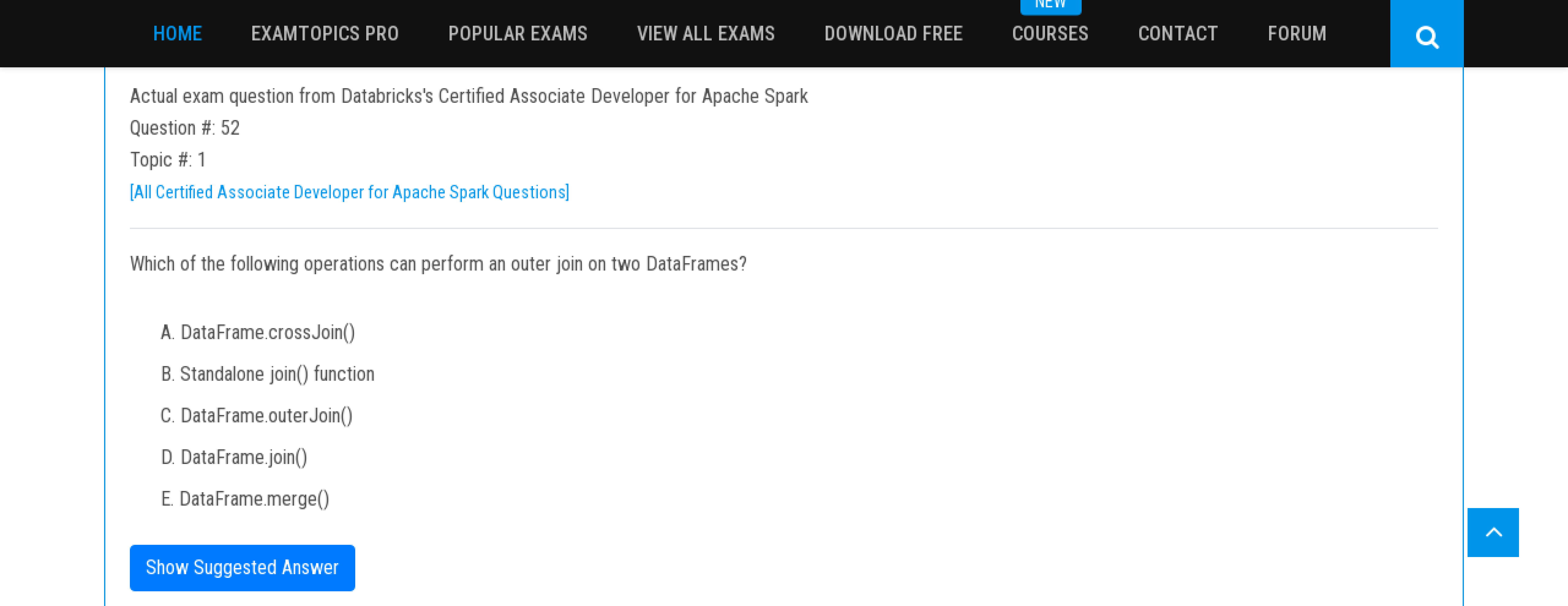
B. The key column storeID needs to be in a list like ["storeID"].

C. The key column storeID needs to be specified in an expression of both DataFrame columns like storesDF.storeId == employeesDF.storeId.

D. There is no DataFrame.join() operation – DataFrame.merge() should be used instead.

E. The column key is the second parameter to join() and the type of join in the third parameter to join() – the second and third arguments should be switched.

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 52

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following operations can perform an outer join on two DataFrames?

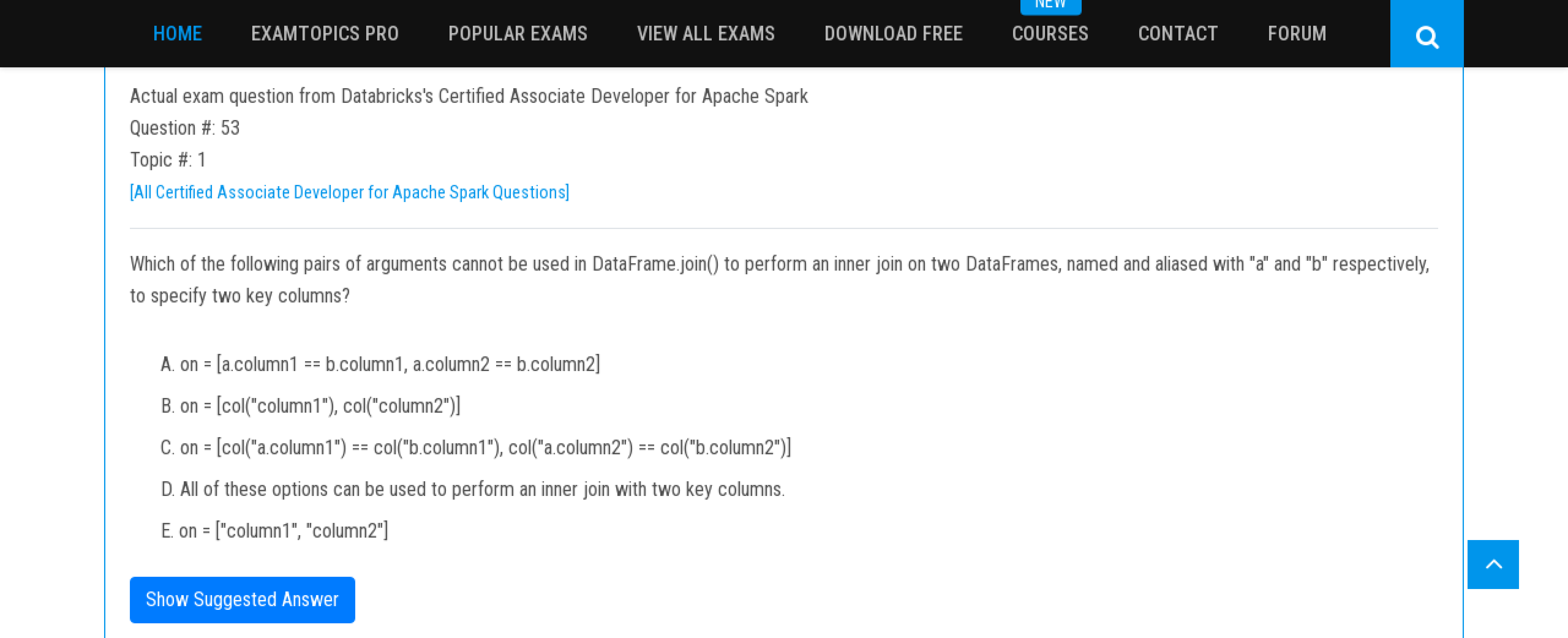
A. DataFrame.crossJoin()

B. Standalone join() function

C. DataFrame.outerJoin()

D. DataFrame.join()

E. DataFrame.merge()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 53

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following pairs of arguments cannot be used in DataFrame.join() to perform an inner join on two DataFrames, named and aliased with "a" and "b" respectively, to specify two key columns?

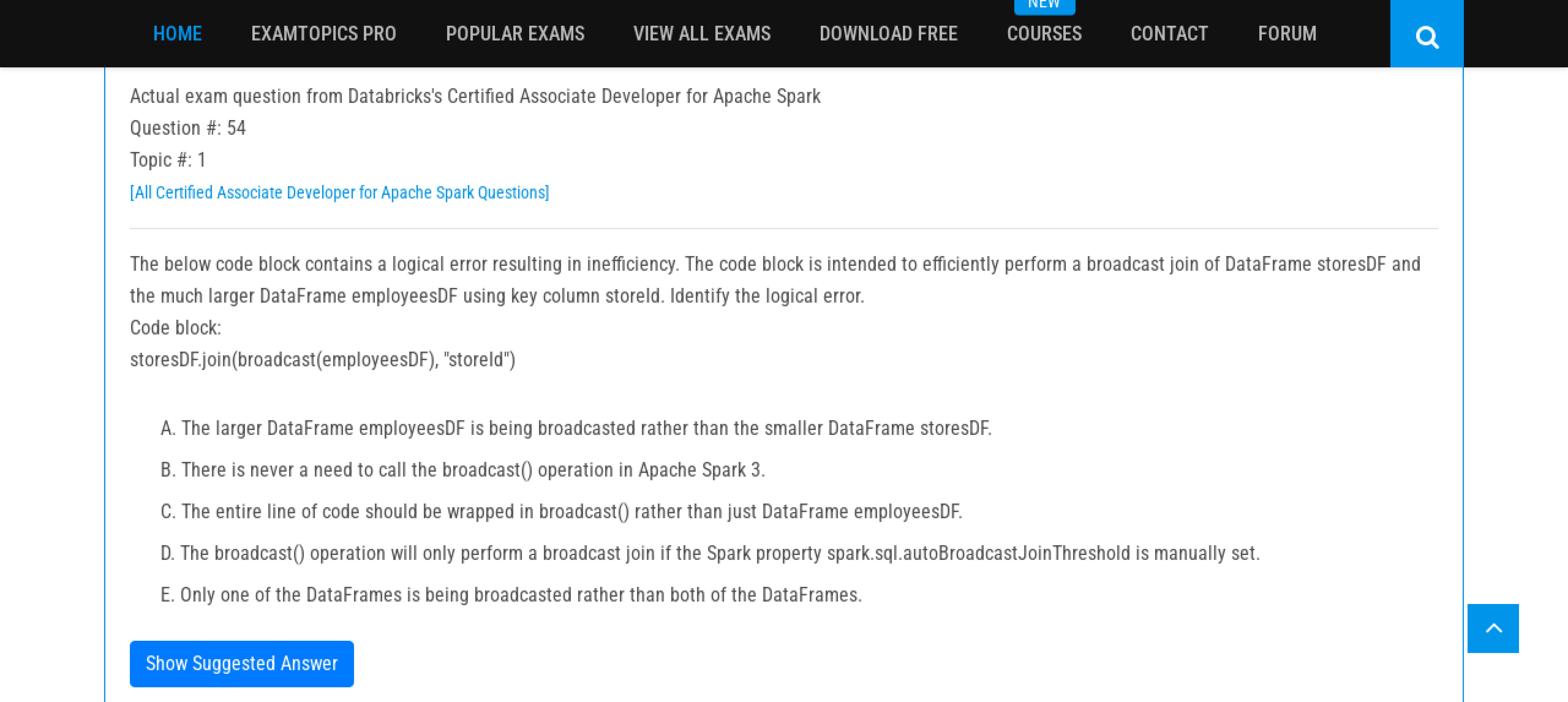
A. on = [a.column1 == b.column1, a.column2 == b.column2]

B. on = [col("column1"), col("column2")]

C. on = [col("a.column1") == col("b.column1"), col("a.column2") == col("b.column2")]

D. All of these options can be used to perform an inner join with two key columns.

E. on = ["column1", "column2"]



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 54

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The below code block contains a logical error resulting in inefficiency. The code block is intended to efficiently perform a broadcast join of DataFrame storesDF and the much larger DataFrame employeesDF using key column storeId. Identify the logical error.

Code block:

```
storesDF.join(broadcast(employeesDF), "storeId")
```

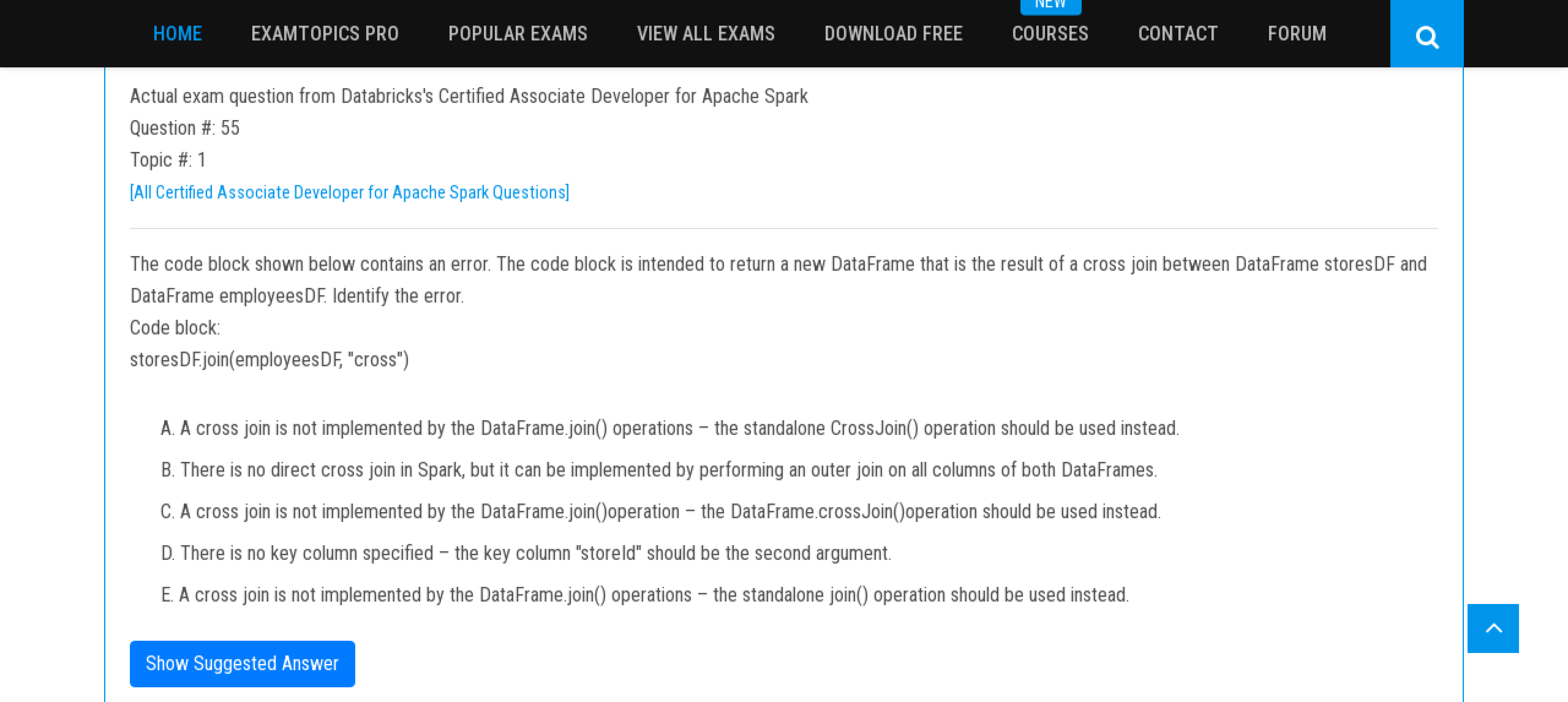
A. The larger DataFrame employeesDF is being broadcasted rather than the smaller DataFrame storesDF.

B. There is never a need to call the broadcast() operation in Apache Spark 3.

C. The entire line of code should be wrapped in broadcast() rather than just DataFrame employeesDF.

D. The broadcast() operation will only perform a broadcast join if the Spark property spark.sql.autoBroadcastJoinThreshold is manually set.

E. Only one of the DataFrames is being broadcasted rather than both of the DataFrames.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 55

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to return a new DataFrame that is the result of a cross join between DataFrame storesDF and DataFrame employeesDF. Identify the error.

Code block:

```
storesDF.join(employeesDF, "cross")
```

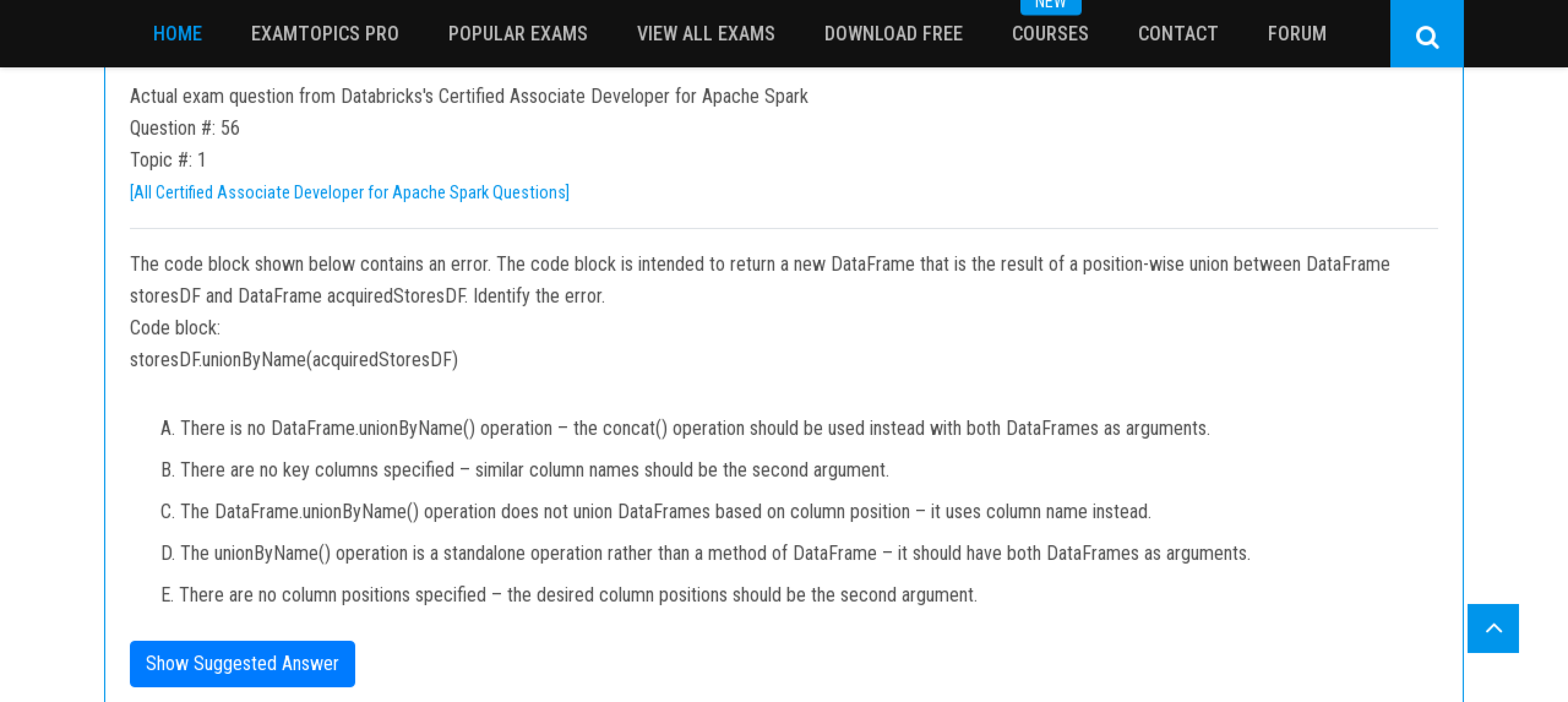
A. A cross join is not implemented by the DataFrame.join() operations – the standalone CrossJoin() operation should be used instead.

B. There is no direct cross join in Spark, but it can be implemented by performing an outer join on all columns of both DataFrames.

C. A cross join is not implemented by the DataFrame.join()operation – the DataFrame.crossJoin()operation should be used instead.

D. There is no key column specified – the key column "storeId" should be the second argument.

E. A cross join is not implemented by the DataFrame.join() operations – the standalone join() operation should be used instead.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 56

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to return a new DataFrame that is the result of a position-wise union between DataFrame storesDF and DataFrame acquiredStoresDF. Identify the error.

Code block:

```
storesDF.unionByName(acquiredStoresDF)
```

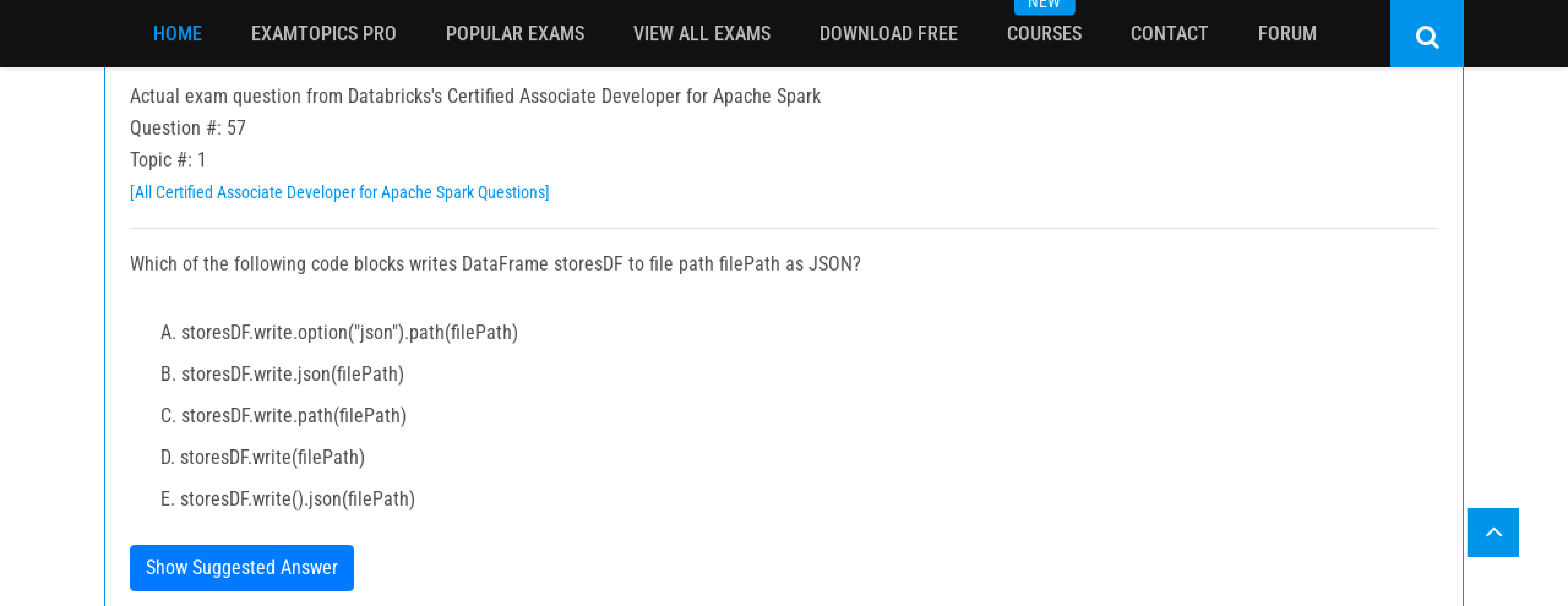
A. There is no DataFrame.unionByName() operation – the concat() operation should be used instead with both DataFrames as arguments.

B. There are no key columns specified – similar column names should be the second argument.

C. The DataFrame.unionByName() operation does not union DataFrames based on column position – it uses column name instead.

D. The unionByName() operation is a standalone operation rather than a method of DataFrame – it should have both DataFrames as arguments.

E. There are no column positions specified – the desired column positions should be the second argument.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 57

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks writes DataFrame storesDF to file path filePath as JSON?

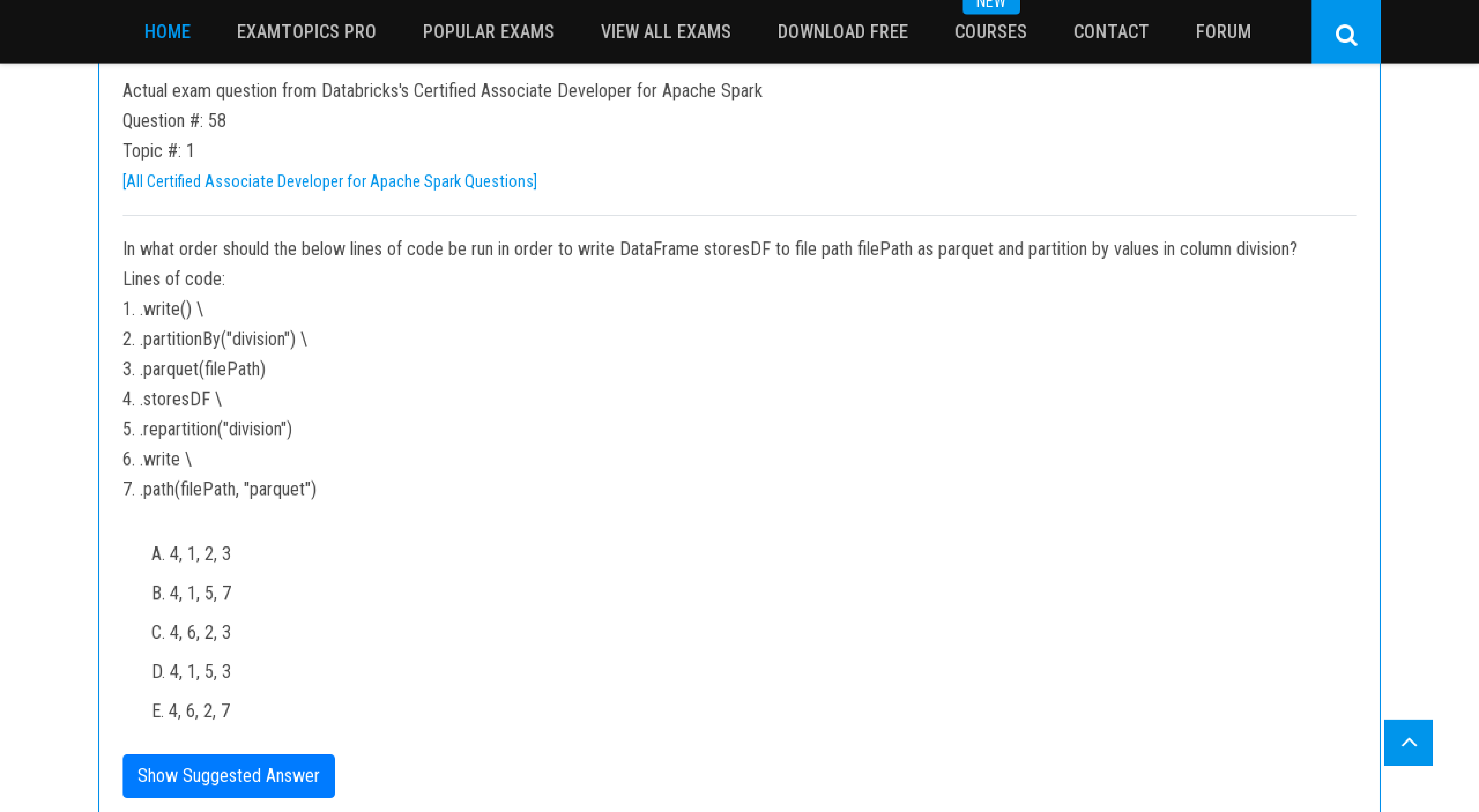
A. storesDF.write.option("json").path(filePath)

B. storesDF.write.json(filePath)

C. storesDF.write.path(filePath)

D. storesDF.write(filePath)

E. storesDF.write().json(filePath)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 58

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

In what order should the below lines of code be run in order to write DataFrame storesDF to file path filePath as parquet and partition by values in column division?

Lines of code:

1. .write() \
2. .partitionBy("division") \
3. .parquet(filePath)
4. .storesDF \
5. .repartition("division")
6. .write \
7. .path(filePath, "parquet")

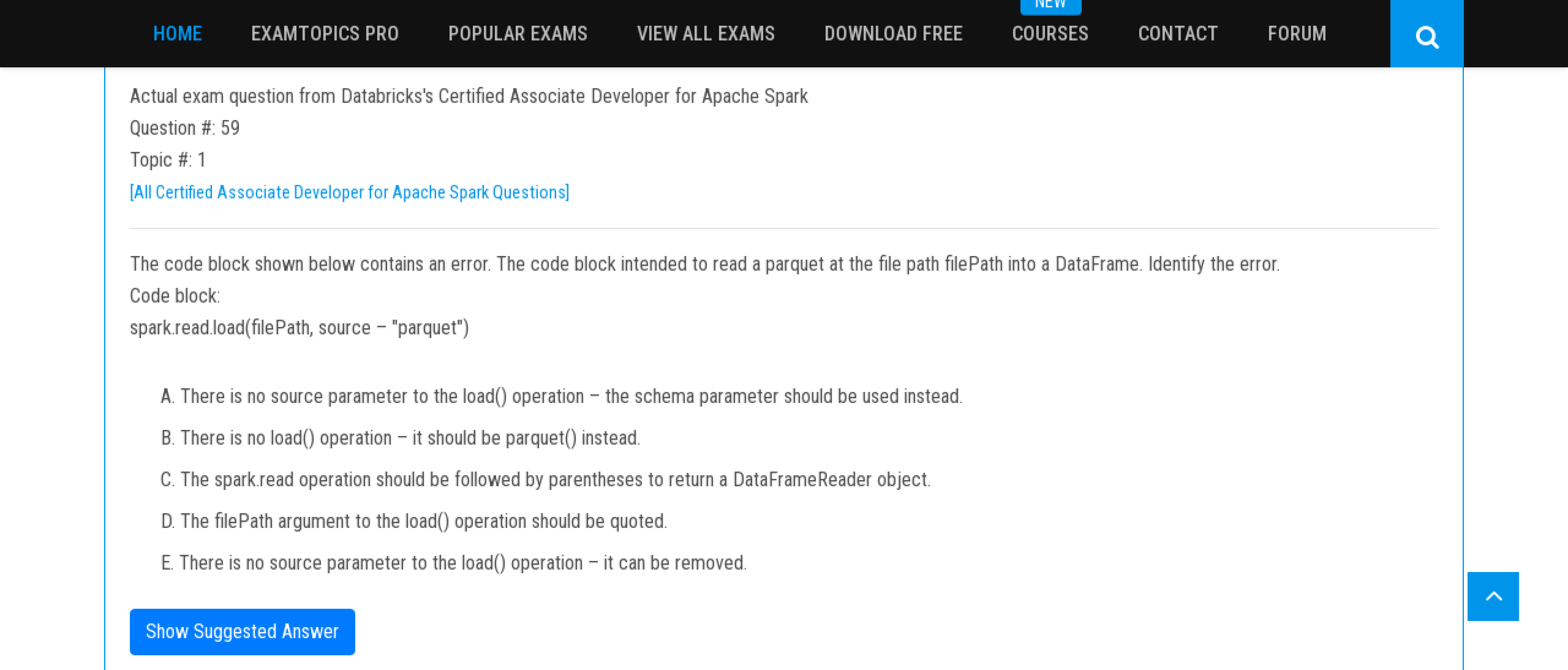
A. 4, 1, 2, 3

B. 4, 1, 5, 7

C. 4, 6, 2, 3

D. 4, 1, 5, 3

E. 4, 6, 2, 7



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 59

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block intended to read a parquet at the file path filePath into a DataFrame. Identify the error.

Code block:

```
spark.read.load(filePath, source – "parquet")
```

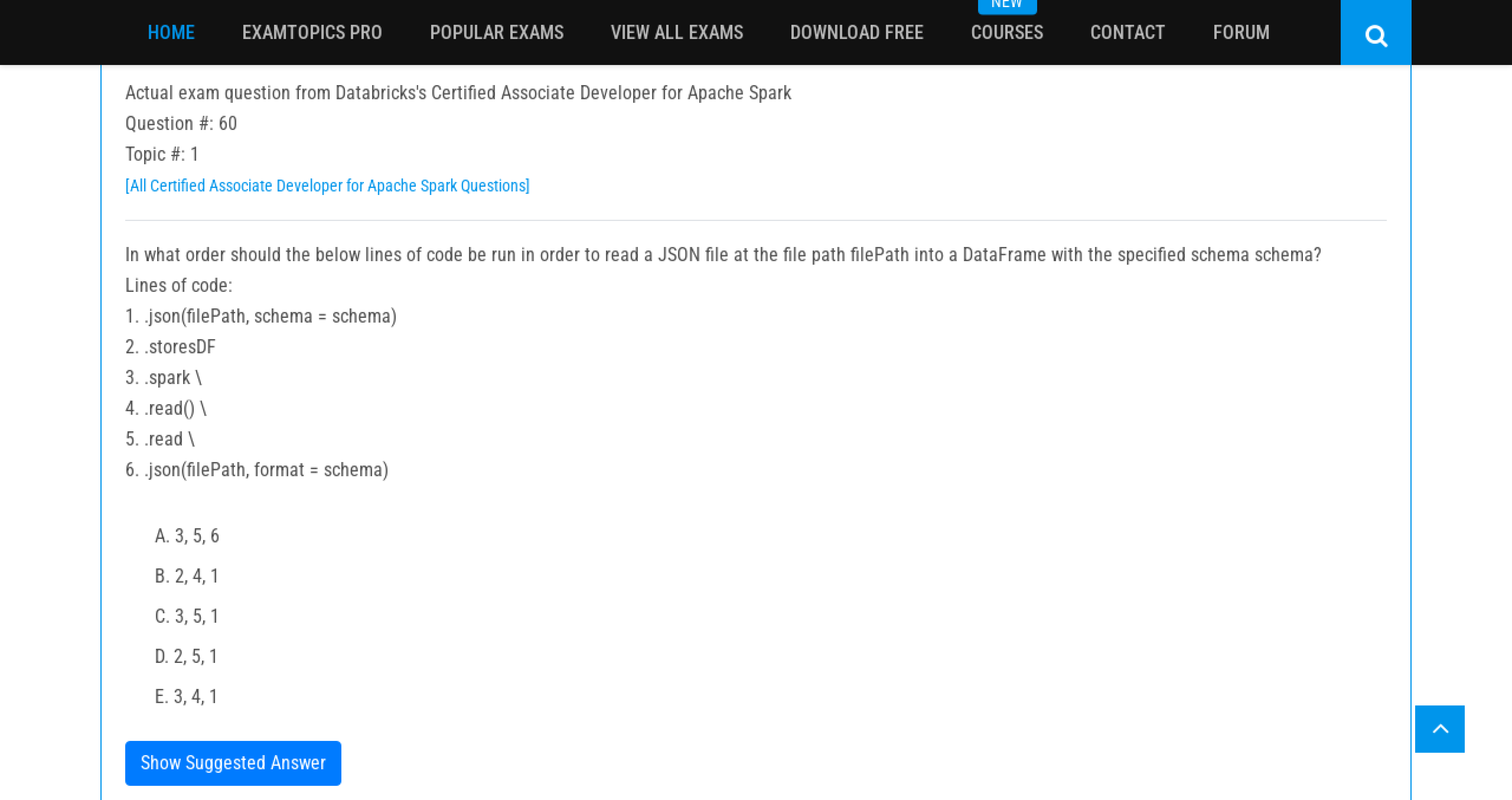
A. There is no source parameter to the load() operation – the schema parameter should be used instead.

B. There is no load() operation – it should be parquet() instead.

C. The spark.read operation should be followed by parentheses to return a DataFrameReader object.

D. The filePath argument to the load() operation should be quoted.

E. There is no source parameter to the load() operation – it can be removed.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 60

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

In what order should the below lines of code be run in order to read a JSON file at the file path filePath into a DataFrame with the specified schema schema?

Lines of code:

1. .json(filePath, schema = schema)
2. .storesDF
3. .spark \
4. .read() \
5. .read \
6. .json(filePath, format = schema)

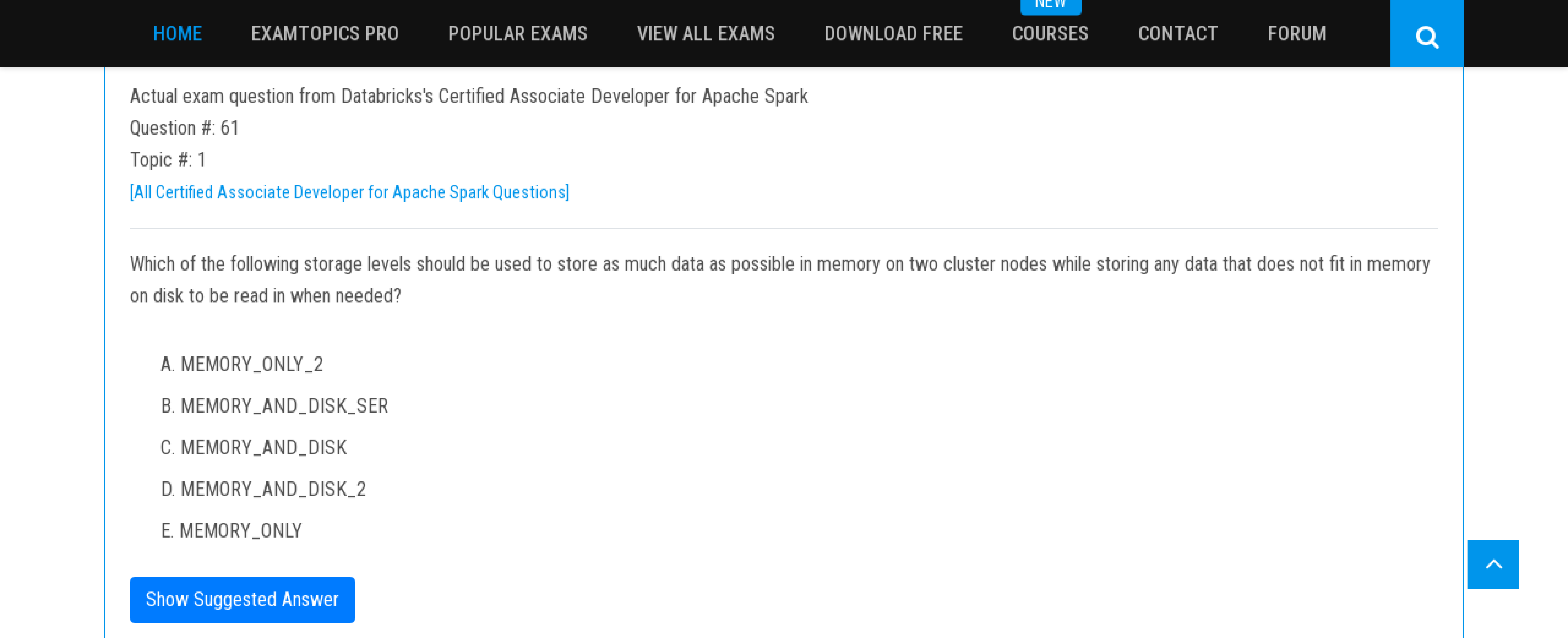
A. 3, 5, 6

B. 2, 4, 1

C. 3, 5, 1

D. 2, 5, 1

E. 3, 4, 1



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 61

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following storage levels should be used to store as much data as possible in memory on two cluster nodes while storing any data that does not fit in memory on disk to be read in when needed?

A. MEMORY_ONLY_2

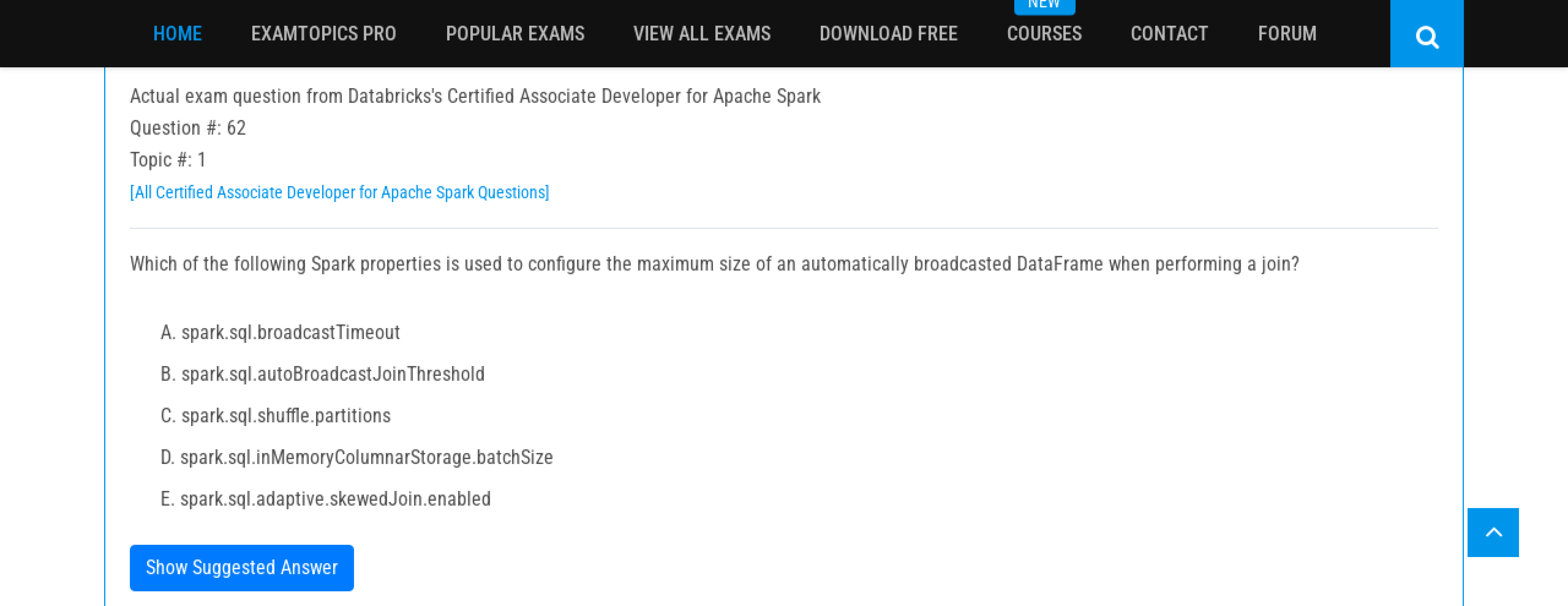
B. MEMORY_AND_DISK_SER

C. MEMORY_AND_DISK

D. MEMORY_AND_DISK_2

E. MEMORY_ONLY

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 62

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following Spark properties is used to configure the maximum size of an automatically broadcasted DataFrame when performing a join?

A. spark.sql.broadcastTimeout

B. spark.sql.autoBroadcastJoinThreshold

C. spark.sql.shuffle.partitions

D. spark.sql.inMemoryColumnarStorage.batchSize

E. spark.sql.adaptive.skewedJoin.enabled

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 63

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following Spark properties is used to configure whether skewed partitions are automatically detected and subdivided into smaller partitions when joining two DataFrames together?

A. spark.sql.adaptive.skewedJoin.enabled

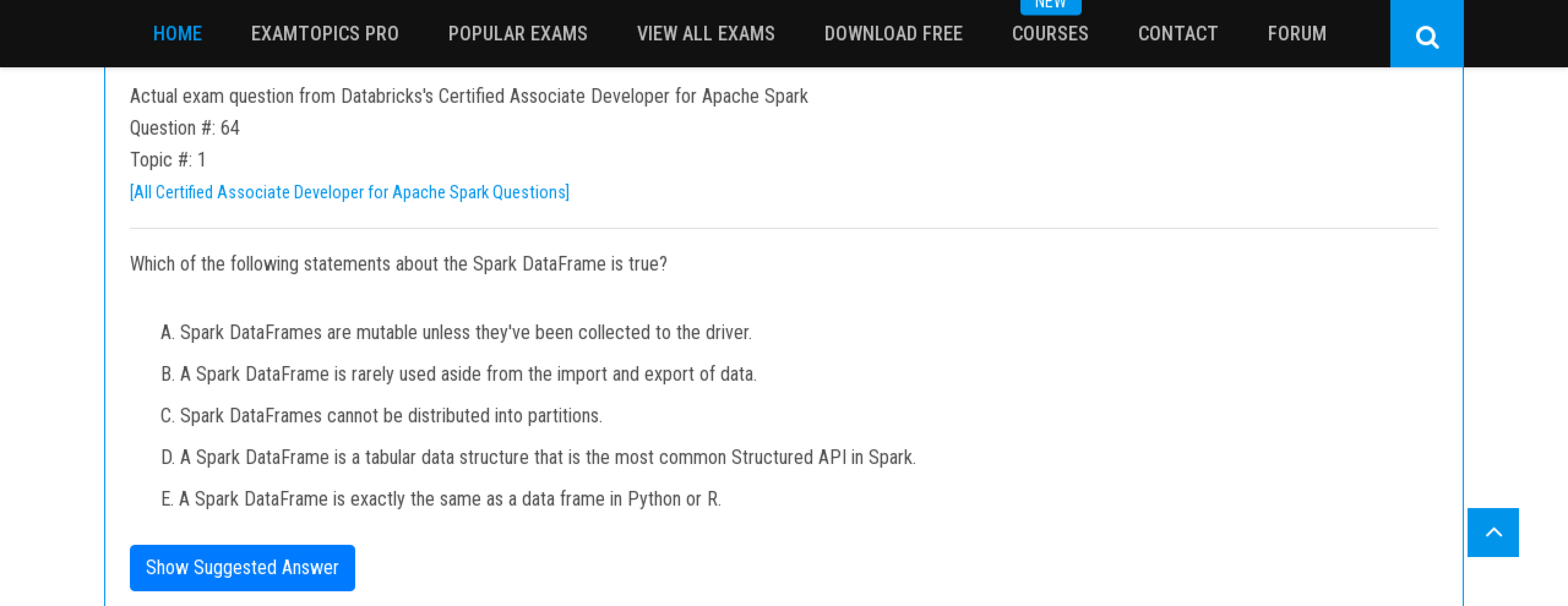
B. spark.sql.adaptive.coalescePartitions.enable

C. spark.sql.adaptive.skewHints.enabled

D. spark.sql.shuffle.partitions

E. spark.sql.shuffle.skewHints.enabled

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 64

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following statements about the Spark DataFrame is true?

A. Spark DataFrames are mutable unless they've been collected to the driver.

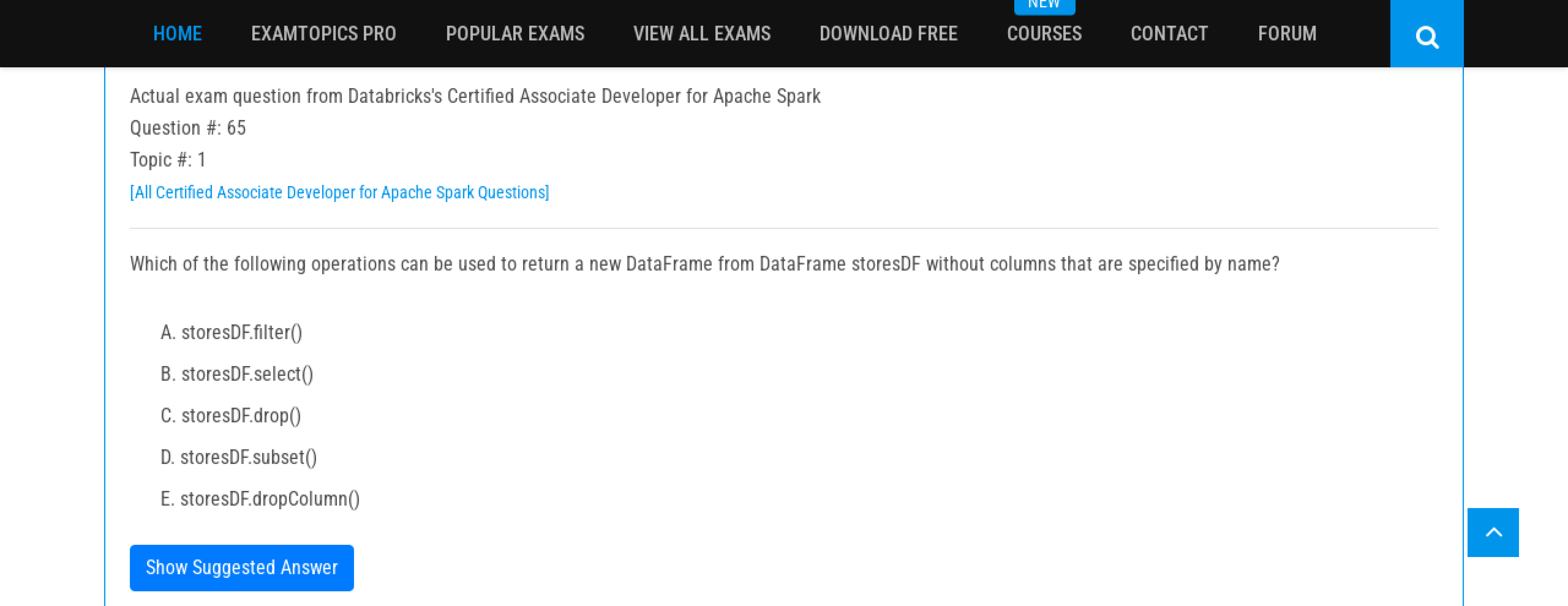
B. A Spark DataFrame is rarely used aside from the import and export of data.

C. Spark DataFrames cannot be distributed into partitions.

D. A Spark DataFrame is a tabular data structure that is the most common Structured API in Spark.

E. A Spark DataFrame is exactly the same as a data frame in Python or R.

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 65

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following operations can be used to return a new DataFrame from DataFrame storesDF without columns that are specified by name?

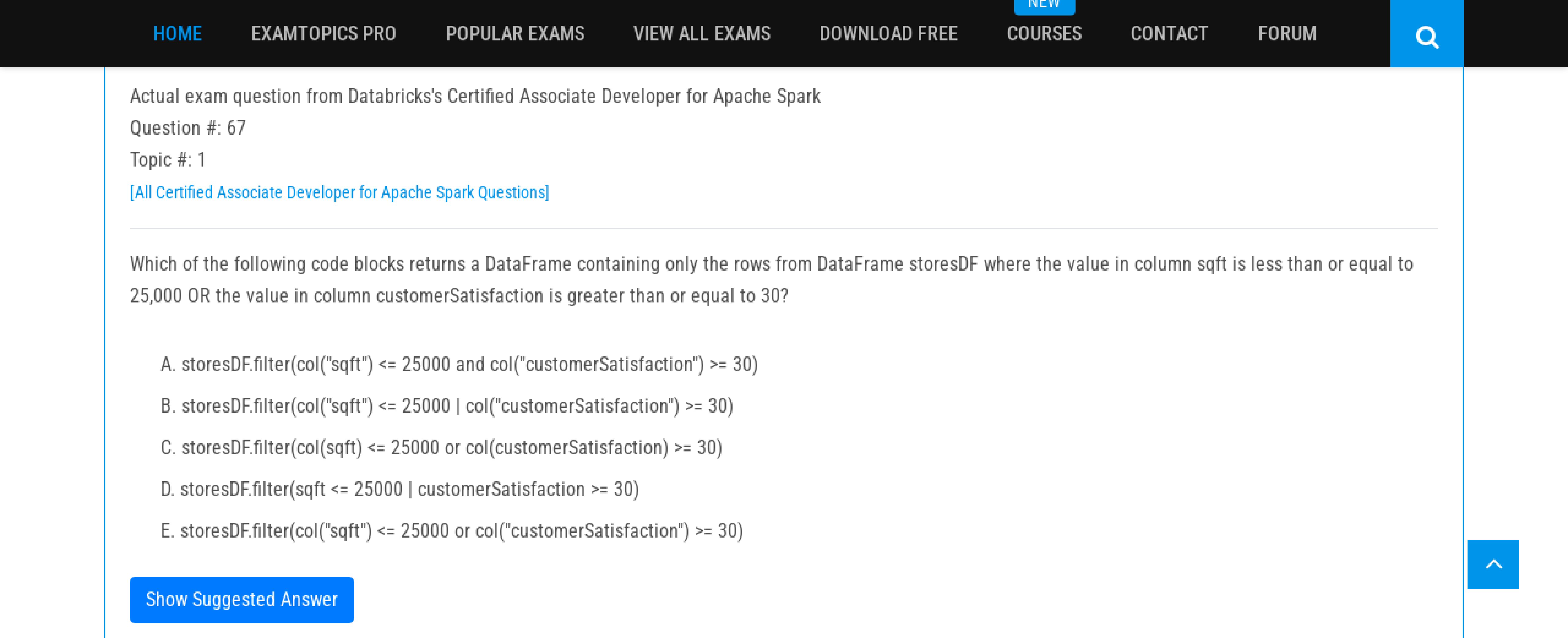
A. storesDF.filter()

B. storesDF.select()

C. storesDF.drop()

D. storesDF.subset()

E. storesDF.dropColumn()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 67

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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?

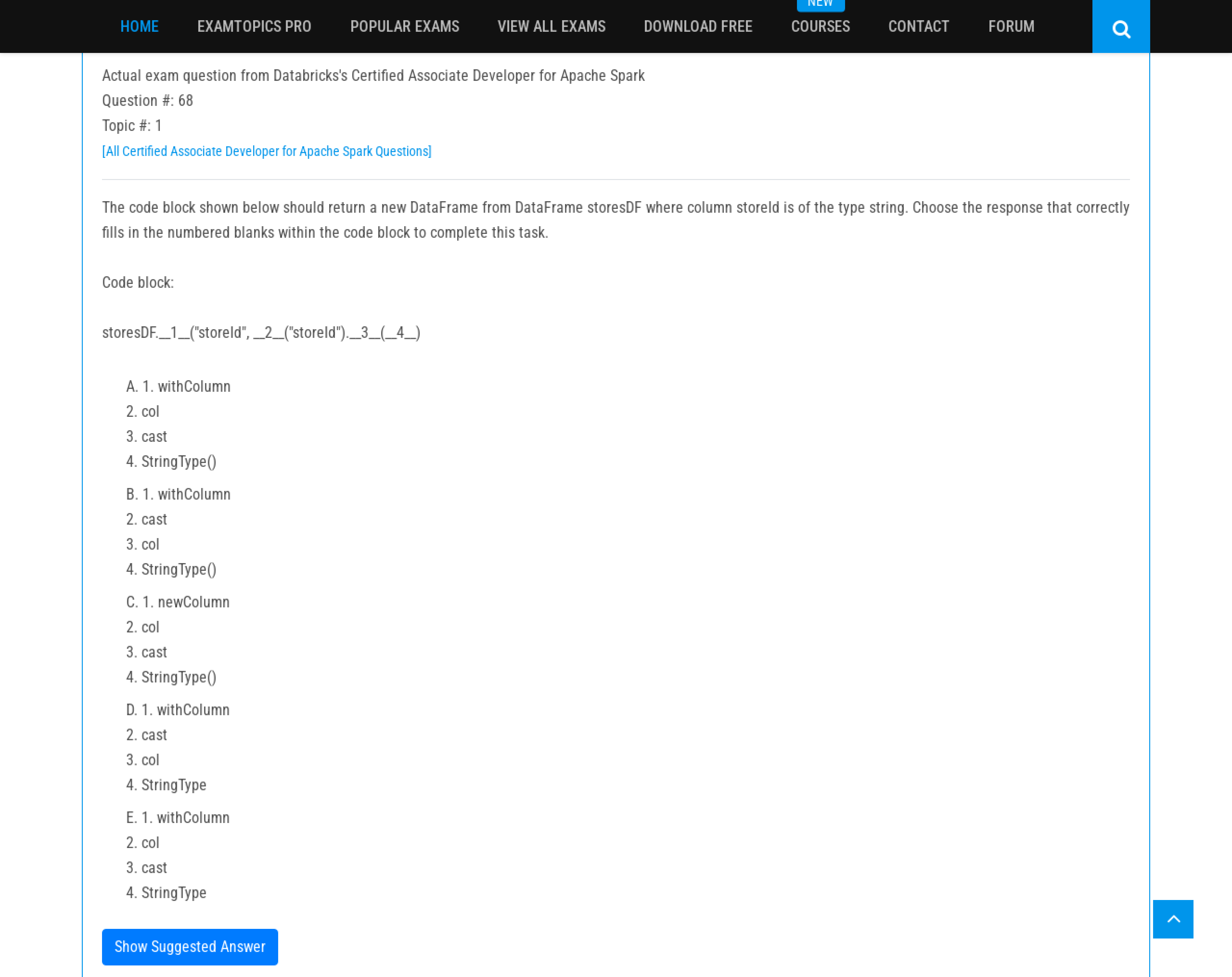
A. storesDF.filter(col("sqft") <= 25000 and col("customerSatisfaction") >= 30)

B. storesDF.filter(col("sqft") <= 25000 | col("customerSatisfaction") >= 30)

C. storesDF.filter(col(sqft) <= 25000 or col(customerSatisfaction) >= 30)

D. storesDF.filter(sqft <= 25000 | customerSatisfaction >= 30)

E. storesDF.filter(col("sqft") <= 25000 or col("customerSatisfaction") >= 30)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 68

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should return a new DataFrame from DataFrame storesDF where column storeId is of the type string. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

storesDF.__1__("storeId", __2__("storeId").__3__(__4__)

A. 1. withColumn
2. col
3. cast
4. StringType()

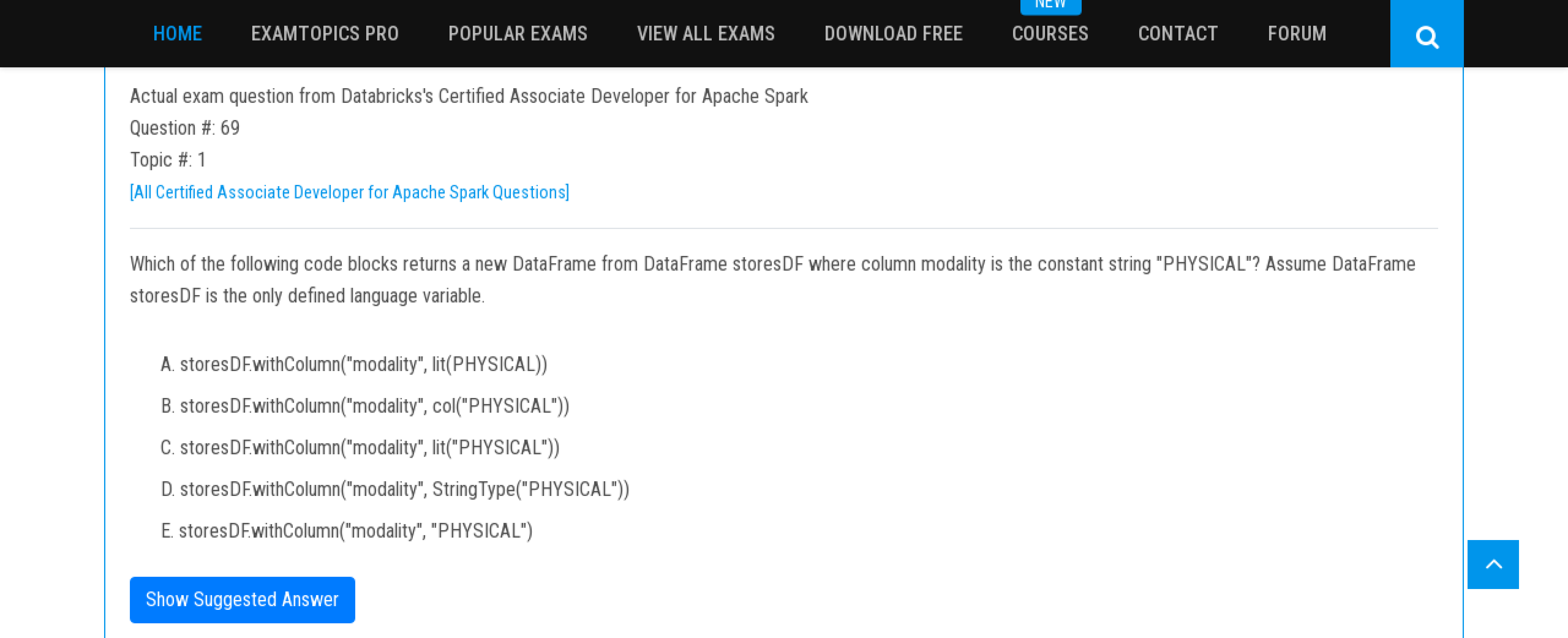
B. 1. withColumn
2. cast
3. col
4. StringType()

C. 1. newColumn
2. col
3. cast
4. StringType()

D. 1. withColumn
2. cast
3. col
4. StringType

E. 1. withColumn
2. col
3. cast
4. StringType

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 69

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns a new DataFrame from DataFrame storesDF where column modality is the constant string "PHYSICAL"? Assume DataFrame storesDF is the only defined language variable.

A. storesDF.withColumn("modality", lit(PHYSICAL))

B. storesDF.withColumn("modality", col("PHYSICAL"))

C. storesDF.withColumn("modality", lit("PHYSICAL"))

D. storesDF.withColumn("modality", StringType("PHYSICAL"))

E. storesDF.withColumn("modality", "PHYSICAL")

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

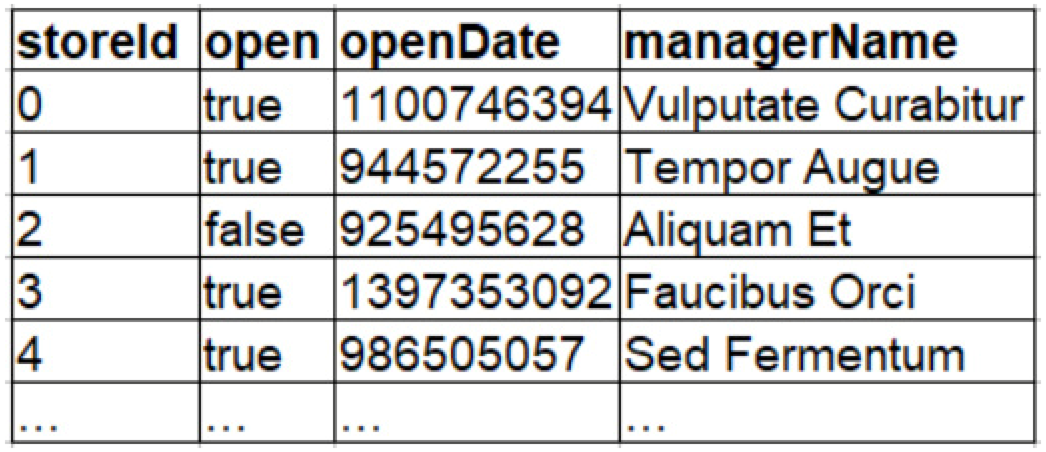
Question #: 70

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to return a new DataFrame where column managerName from DataFrame storesDF is split at the space character into column managerFirstName and column managerLastName. Identify the error.

A sample of DataFrame storesDF is displayed below:



storeId	open	openDate	managerName
0	true	1100746394	Vulputate Curabitur
1	true	944572255	Tempor Augue
2	false	925495628	Aliquam Et
3	true	1397353092	Faucibus Orci
4	true	986505057	Sed Fermentum
...	...	...	...

Code block:

```
storesDF.withColumn("managerFirstName", col("managerName").split(" ").getItem(0))
.withColumn("managerLastName", col("managerName").split(" ").getItem(1))
```

A. The index values of 0 and 1 are not correct – they should be 1 and 2, respectively.

B. The index values of 0 and 1 should be provided as second arguments to the split() operation rather than indexing the result.

C. The split() operation comes from the imported functions object. It accepts a string column name and split character as arguments. It is not a method of a Column object.

D. The split() operation comes from the imported functions object. It accepts a Column object and split character as arguments. It is not a method of a Column object.

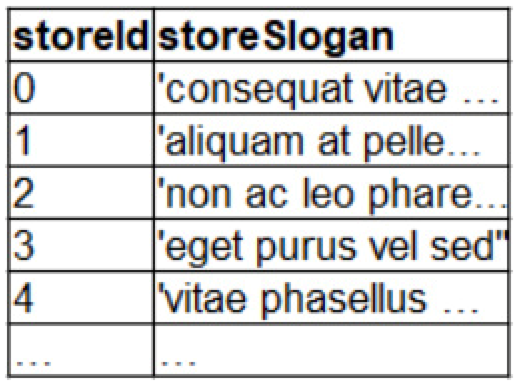
E. The withColumn operation cannot be called twice in a row.

Show Suggested Answer

Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 71
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should return a new DataFrame where single quotes in column storeSlogan have been replaced with double quotes. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

A sample of DataFrame storesDF is below:



storeId	storeSlogan
0	'consequat vitae ...
1	'aliquam at pelle...
2	'non ac leo phare...
3	'eget purus vel sed"
4	'vitae phasellus ...
...	...

Code block:

storesDF.__1__(__2__, __3__(__4__, __5__, __6__))

A. 1. withColumn
2. "storeSlogan"
3. regexp_extract
4. col("storeSlogan")
5. "\""
6. "'"

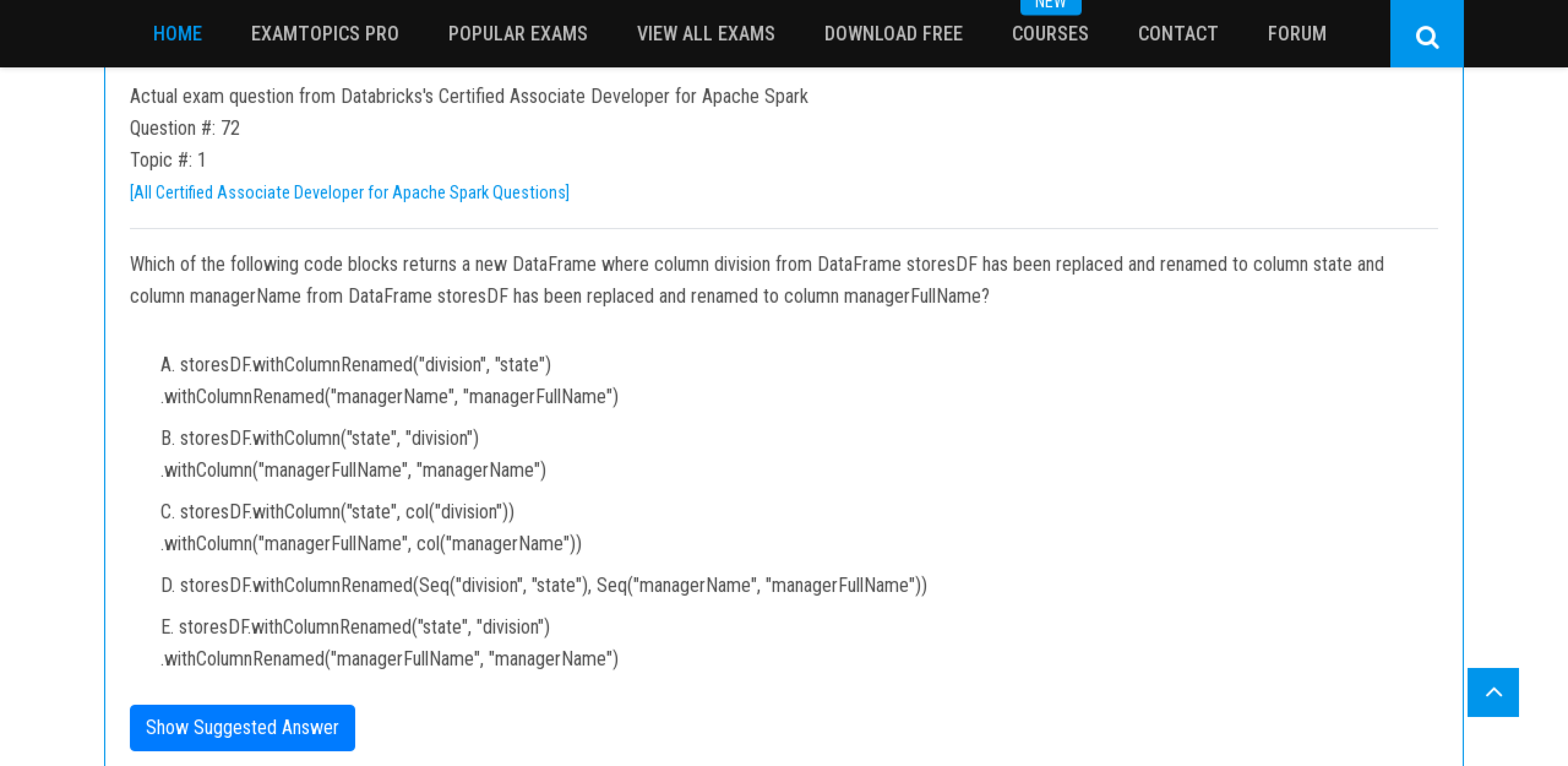
B. 1. newColumn
2. storeSlogan
3. regexp_extract
4. col(storeSlogan)
5. "\""
6. "'"

C. 1. withColumn
2. "storeSlogan"
3. regexp_replace
4. col("storeSlogan")
5. "\""
6. "'"

D. 1. withColumn
2. "storeSlogan"
3. regexp_replace
4. col("storeSlogan")
5. "'"
6. "\""

E. 1. withColumn
2. "storeSlogan"
3. regexp_extract
4. col("storeSlogan")
5. "'"
6. "\""

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 72

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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")
.withColumnRenamed("managerName", "managerFullName")

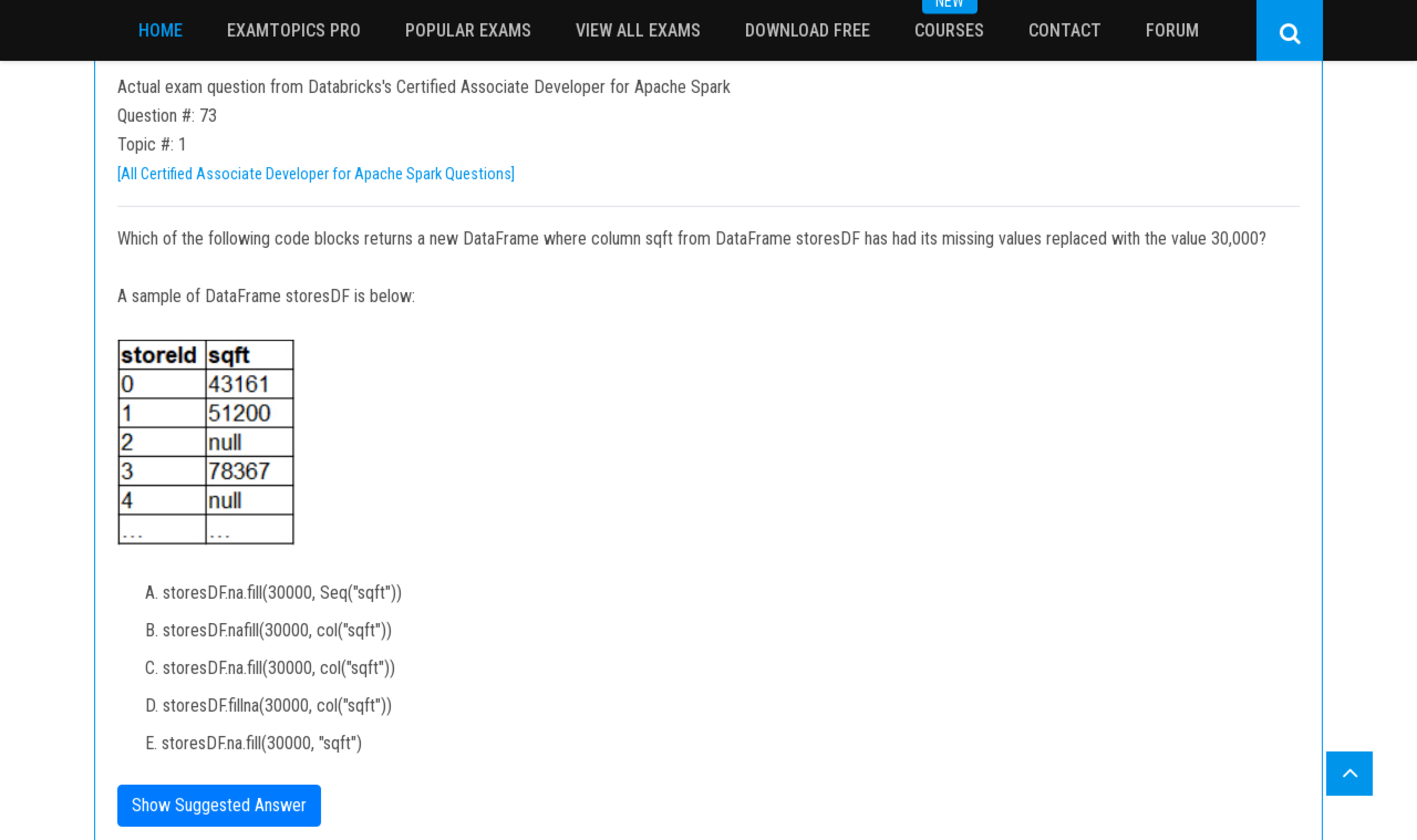
B. storesDF.withColumn("state", "division")
.withColumn("managerFullName", "managerName")

C. storesDF.withColumn("state", col("division"))
.withColumn("managerFullName", col("managerName"))

D. storesDF.withColumnRenamed(Seq("division", "state"), Seq("managerName", "managerFullName"))

E. storesDF.withColumnRenamed("state", "division")
.withColumnRenamed("managerFullName", "managerName")

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 73

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns a new DataFrame where column sqft from DataFrame storesDF has had its missing values replaced with the value 30,000?

A sample of DataFrame storesDF is below:

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

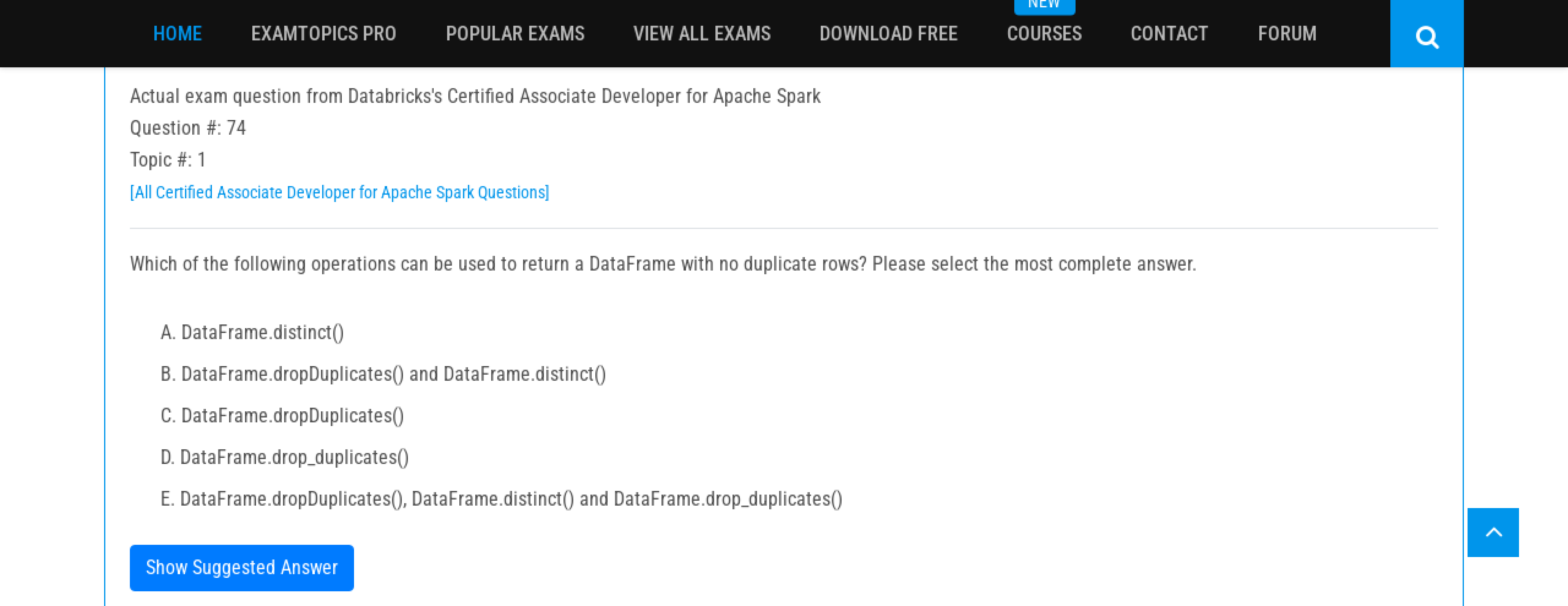
A. storesDF.na.fill(30000, Seq("sqft"))

B. storesDF.nafill(30000, col("sqft"))

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

D. storesDF.fillna(30000, col("sqft"))

E. storesDF.na.fill(30000, "sqft")



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 74

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following operations can be used to return a DataFrame with no duplicate rows? Please select the most complete answer.

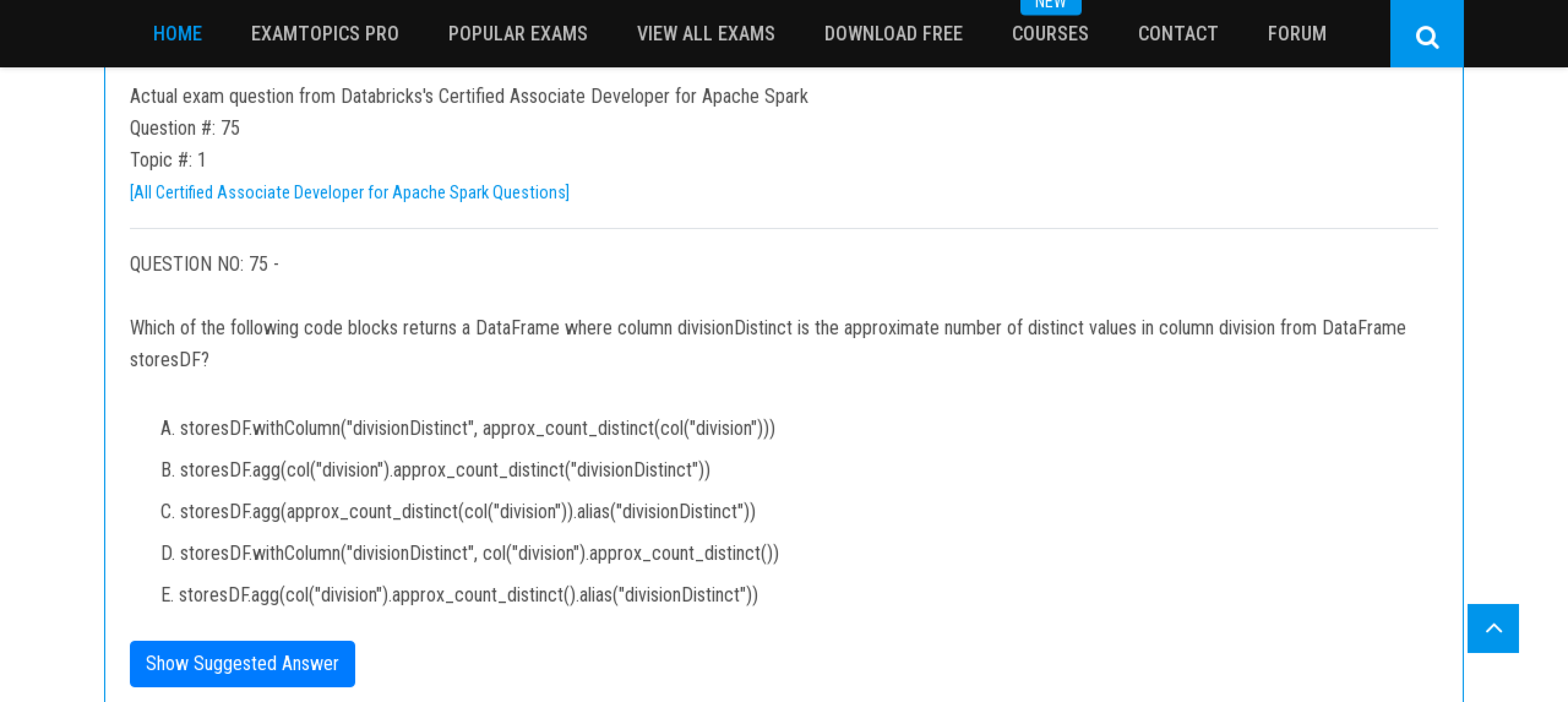
A. DataFrame.distinct()

B. DataFrame.dropDuplicates() and DataFrame.distinct()

C. DataFrame.dropDuplicates()

D. DataFrame.drop_duplicates()

E. DataFrame.dropDuplicates(), DataFrame.distinct() and DataFrame.drop_duplicates()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 75

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

QUESTION NO: 75 -

Which of the following code blocks returns a DataFrame where column divisionDistinct is the approximate number of distinct values in column division from DataFrame storesDF?

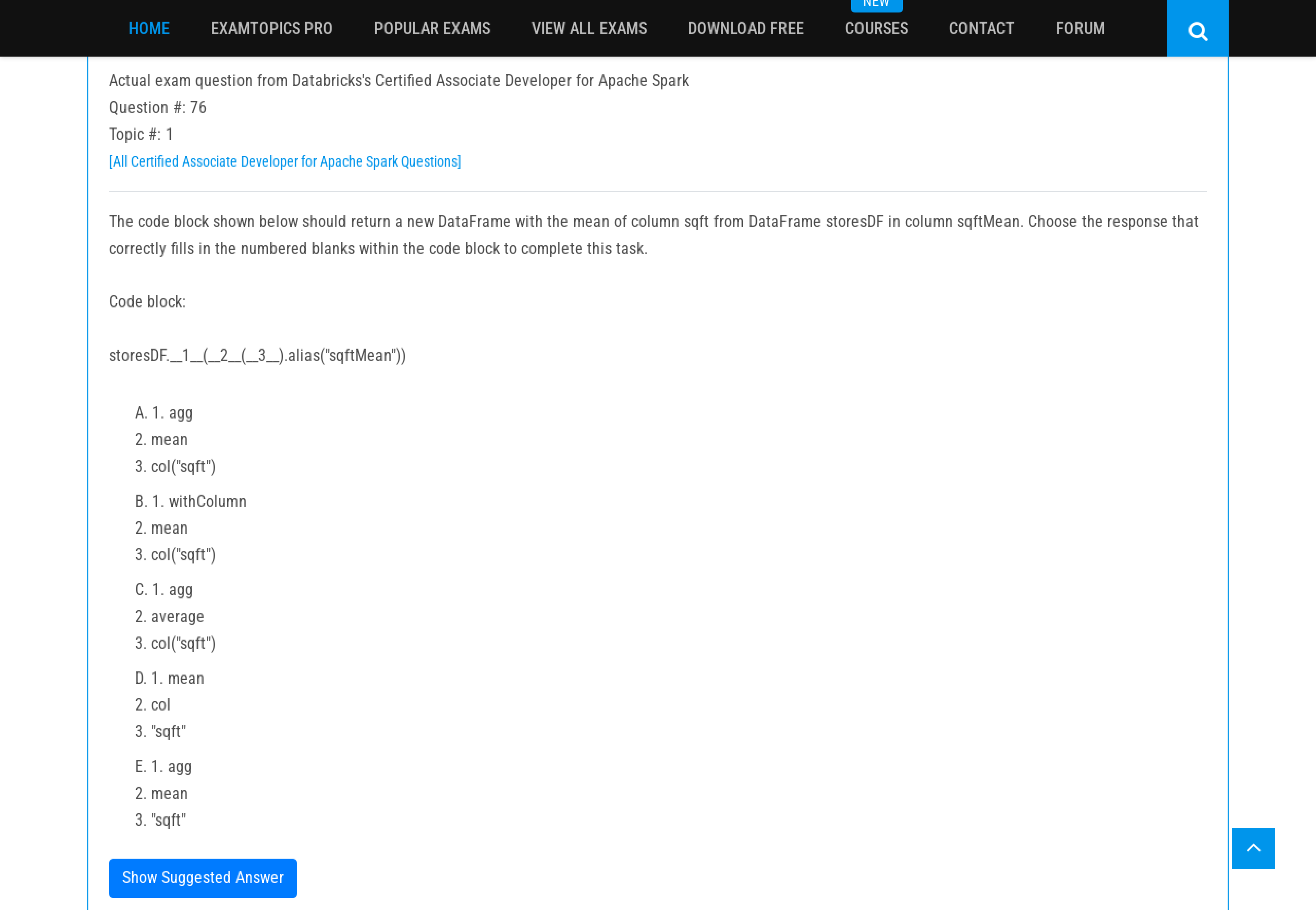
A. storesDF.withColumn("divisionDistinct", approx_count_distinct(col("division")))

B. storesDF.agg(col("division").approx_count_distinct("divisionDistinct"))

C. storesDF.agg(approx_count_distinct(col("division")).alias("divisionDistinct"))

D. storesDF.withColumn("divisionDistinct", col("division").approx_count_distinct())

E. storesDF.agg(col("division").approx_count_distinct().alias("divisionDistinct"))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 76

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should return a new DataFrame with the mean of column sqft from DataFrame storesDF in column sqftMean. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

```
storesDF.__1__(__2__(__3__).alias("sqftMean"))
```

A. 1. agg
2. mean
3. col("sqft")

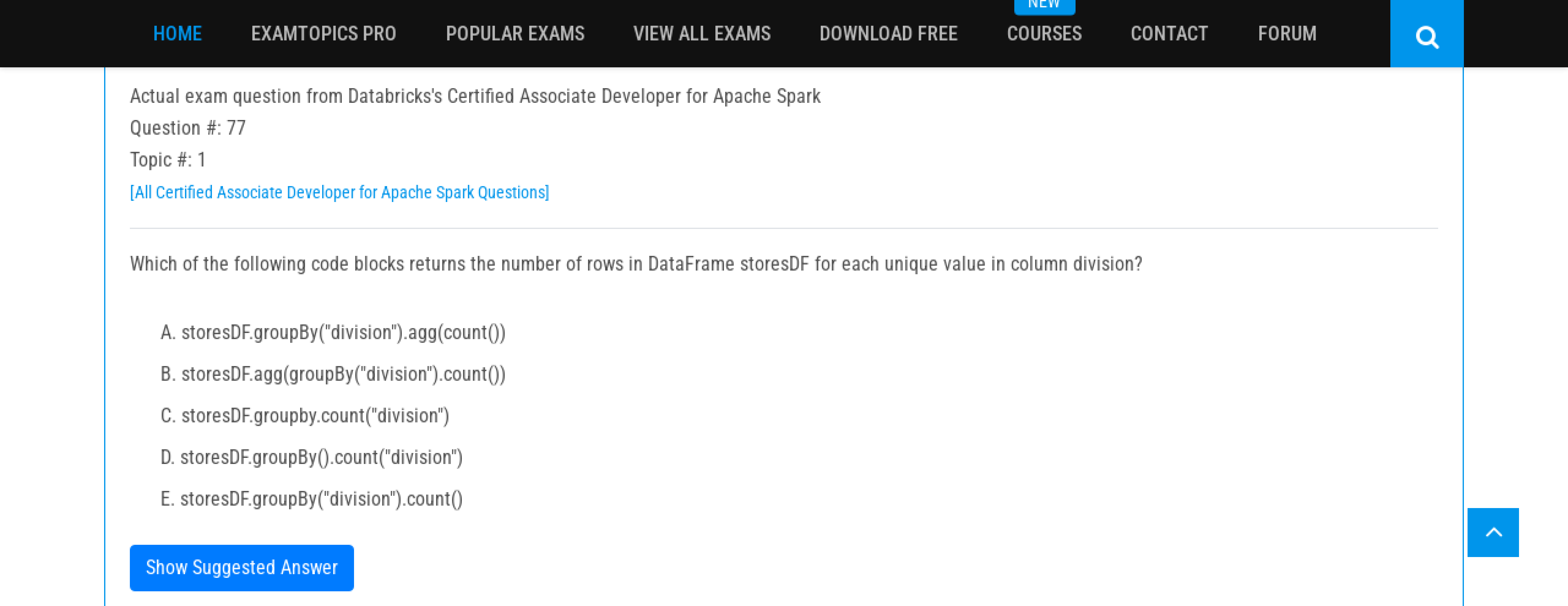
B. 1. withColumn
2. mean
3. col("sqft")

C. 1. agg
2. average
3. col("sqft")

D. 1. mean
2. col
3. "sqft"

E. 1. agg
2. mean
3. "sqft"

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 77

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns the number of rows in DataFrame storesDF for each unique value in column division?

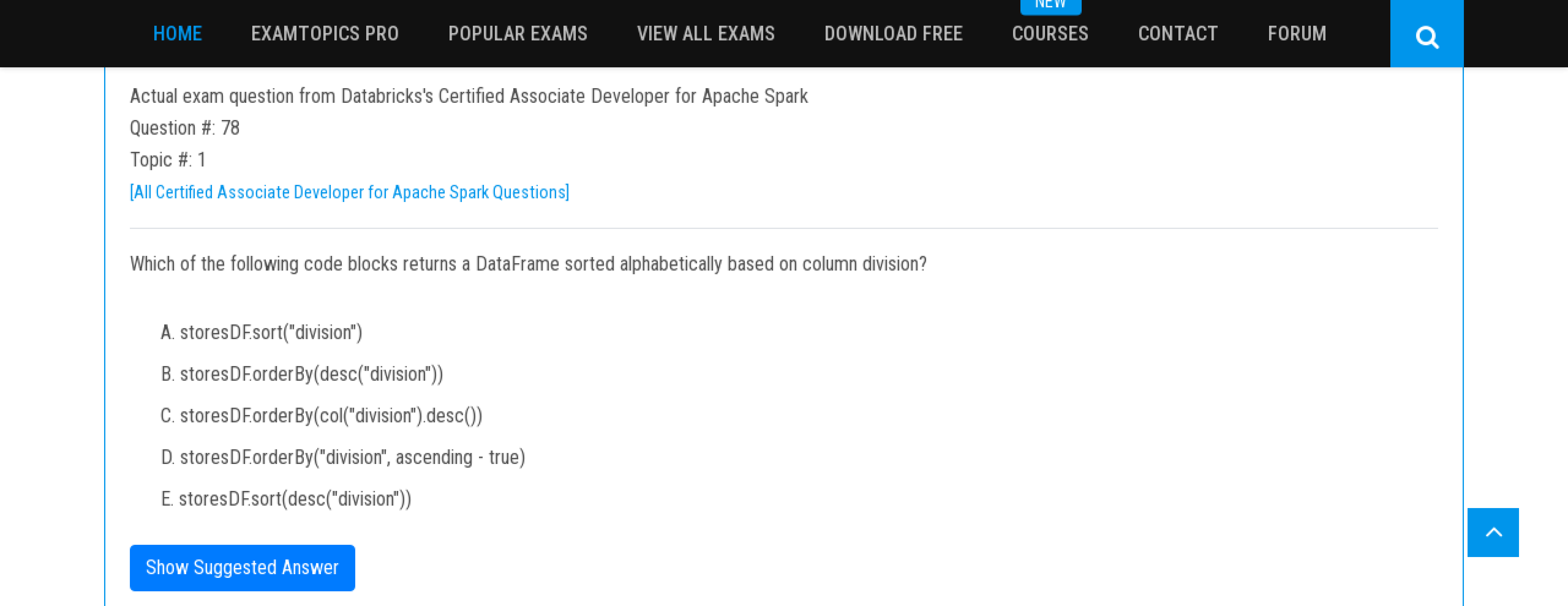
A. storesDF.groupBy("division").agg(count())

B. storesDF.agg(groupBy("division").count())

C. storesDF.groupby.count("division")

D. storesDF.groupBy().count("division")

E. storesDF.groupBy("division").count()



HOME EXAMTOPICS PRO POPULAR EXAMS VIEW ALL EXAMS DOWNLOAD FREE NEW COURSES CONTACT FORUM

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 78

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns a DataFrame sorted alphabetically based on column division?

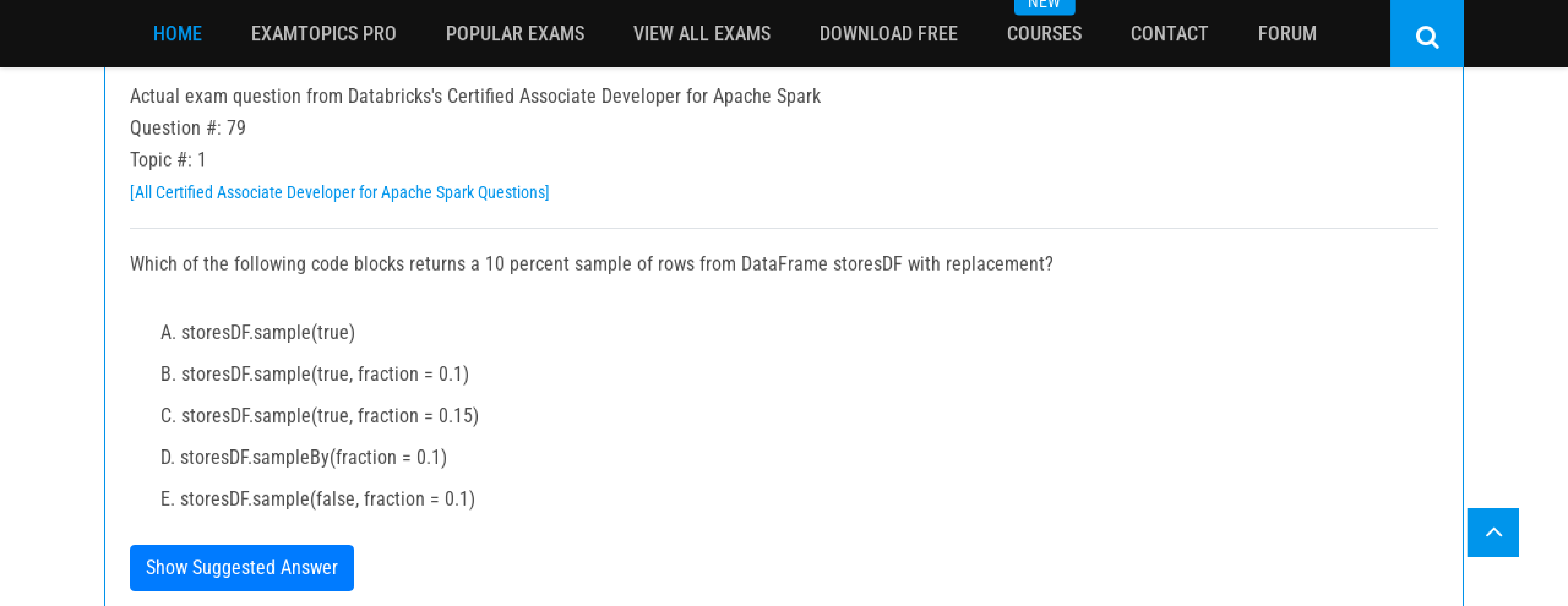
A. storesDF.sort("division")

B. storesDF.orderBy(desc("division"))

C. storesDF.orderBy(col("division").desc())

D. storesDF.orderBy("division", ascending - true)

E. storesDF.sort(desc("division"))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 79

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns a 10 percent sample of rows from DataFrame storesDF with replacement?

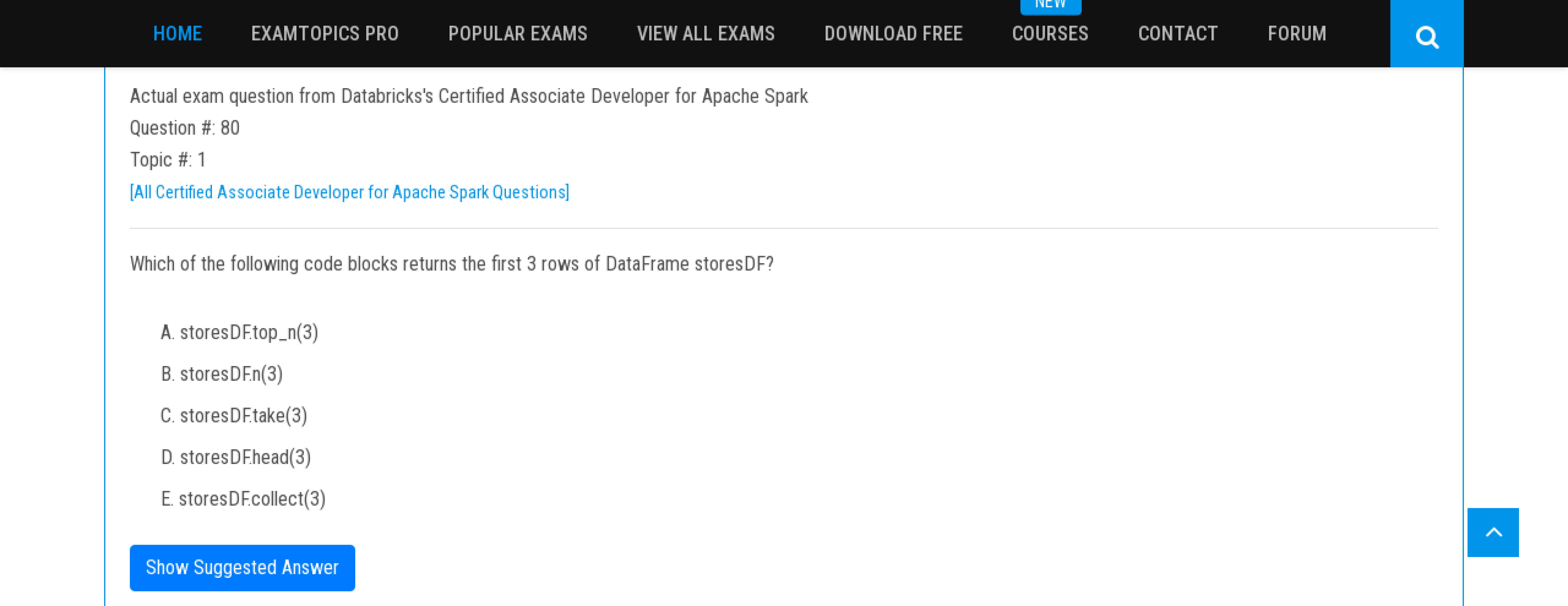
A. storesDF.sample(true)

B. storesDF.sample(true, fraction = 0.1)

C. storesDF.sample(true, fraction = 0.15)

D. storesDF.sampleBy(fraction = 0.1)

E. storesDF.sample(false, fraction = 0.1)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 80

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns the first 3 rows of DataFrame storesDF?

A. storesDF.top_n(3)

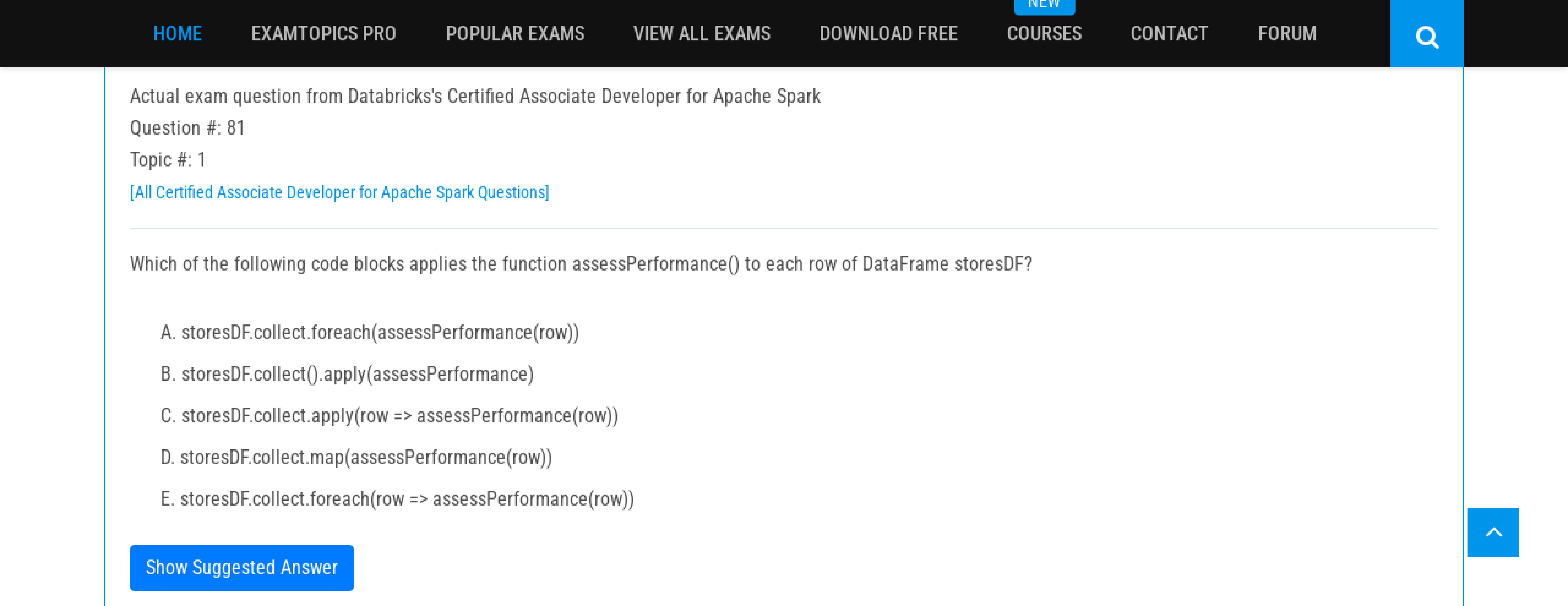
B. storesDF.n(3)

C. storesDF.take(3)

D. storesDF.head(3)

E. storesDF.collect(3)

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 81

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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

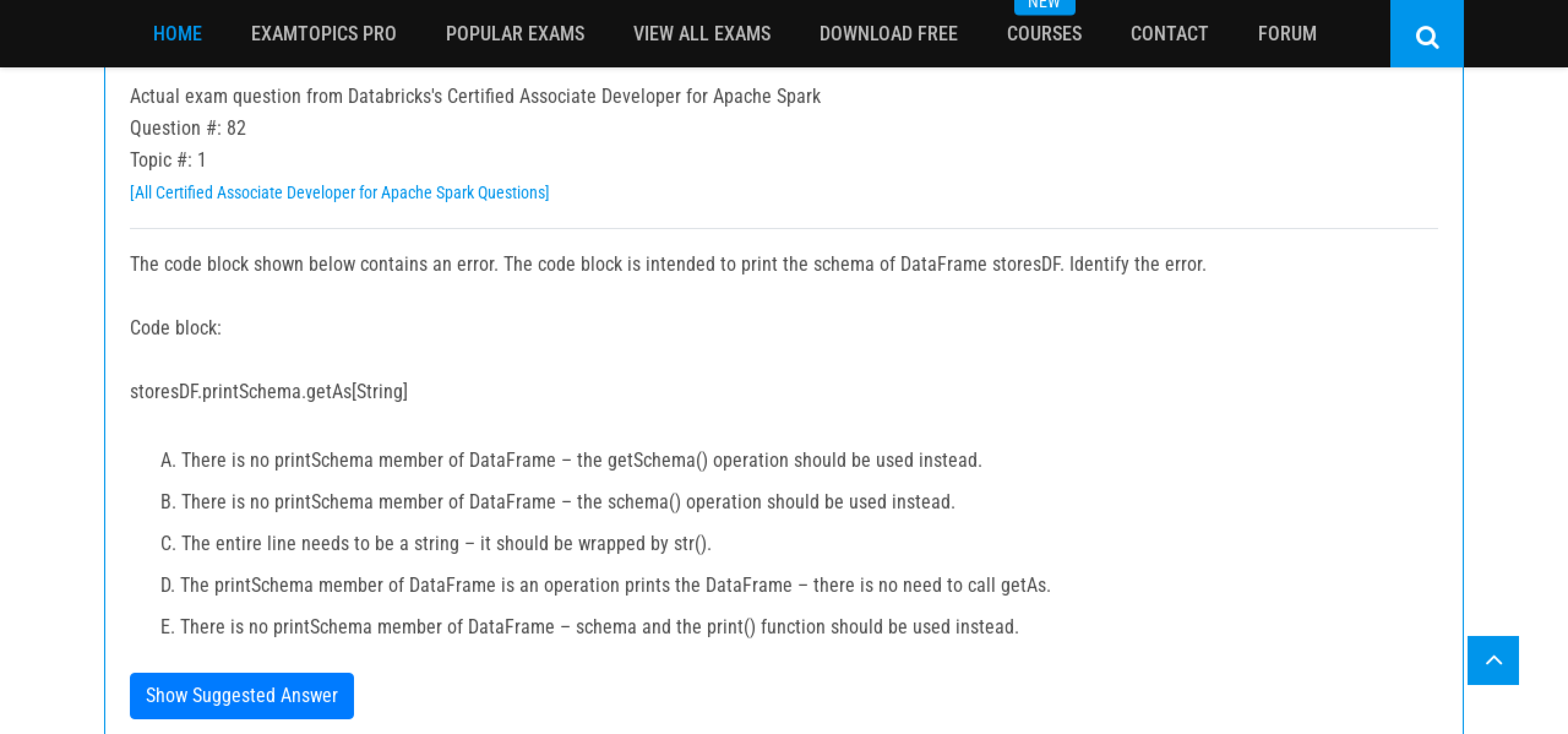
A. storesDF.collect.foreach(assessPerformance(row))

B. storesDF.collect().apply(assessPerformance)

C. storesDF.collect.apply(row => assessPerformance(row))

D. storesDF.collect.map(assessPerformance(row))

E. storesDF.collect.foreach(row => assessPerformance(row))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 82

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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.getAs[String]
```

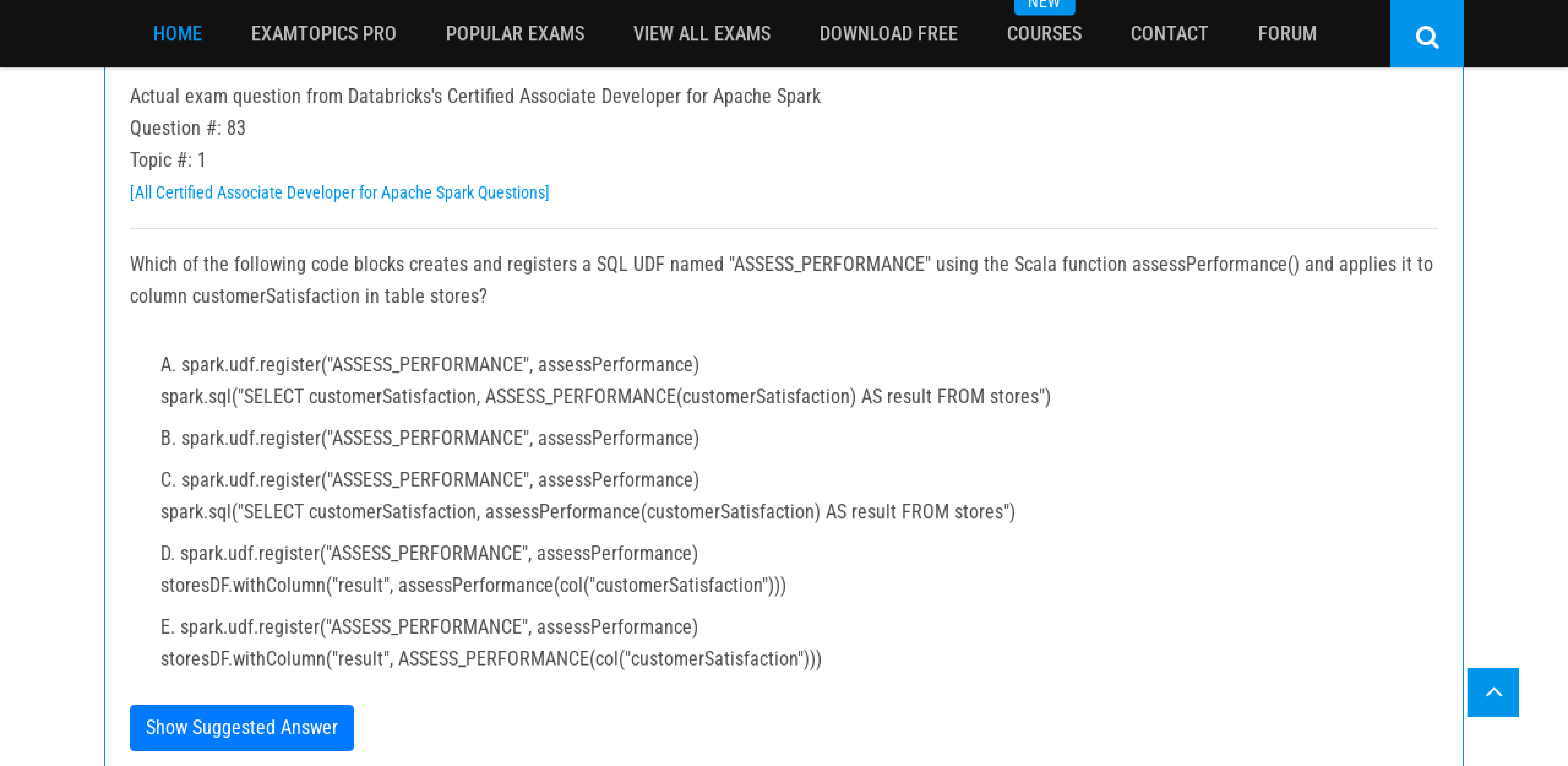
A. There is no printSchema member of DataFrame – the getSchema() operation should be used instead.

B. There is no printSchema member of DataFrame – the schema() operation should be used instead.

C. The entire line needs to be a string – it should be wrapped by str().

D. The printSchema member of DataFrame is an operation prints the DataFrame – there is no need to call getAs.

E. There is no printSchema member of DataFrame – schema and the print() function should be used instead.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 83
Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks creates and registers a SQL UDF named "ASSESS_PERFORMANCE" using the Scala function assessPerformance() and applies it to column customerSatisfaction in table stores?

A. spark.udf.register("ASSESS_PERFORMANCE", assessPerformance)
spark.sql("SELECT customerSatisfaction, ASSESS_PERFORMANCE(customerSatisfaction) AS result FROM stores")

B. spark.udf.register("ASSESS_PERFORMANCE", assessPerformance)

C. spark.udf.register("ASSESS_PERFORMANCE", assessPerformance)
spark.sql("SELECT customerSatisfaction, assessPerformance(customerSatisfaction) AS result FROM stores")

D. spark.udf.register("ASSESS_PERFORMANCE", assessPerformance)
storesDF.withColumn("result", assessPerformance(col("customerSatisfaction")))

E. spark.udf.register("ASSESS_PERFORMANCE", assessPerformance)
storesDF.withColumn("result", ASSESS_PERFORMANCE(col("customerSatisfaction")))

Show Suggested Answer

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 84

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should use SQL to return a new DataFrame containing column storeId and column managerName from a table created from DataFrame storesDF. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

```
__1__.__2__("stores")
__3__.__4__("SELECT storeId, managerName FROM stores")
```

A. 1. spark
2. createOrReplaceTempView
3. storesDF
4. query

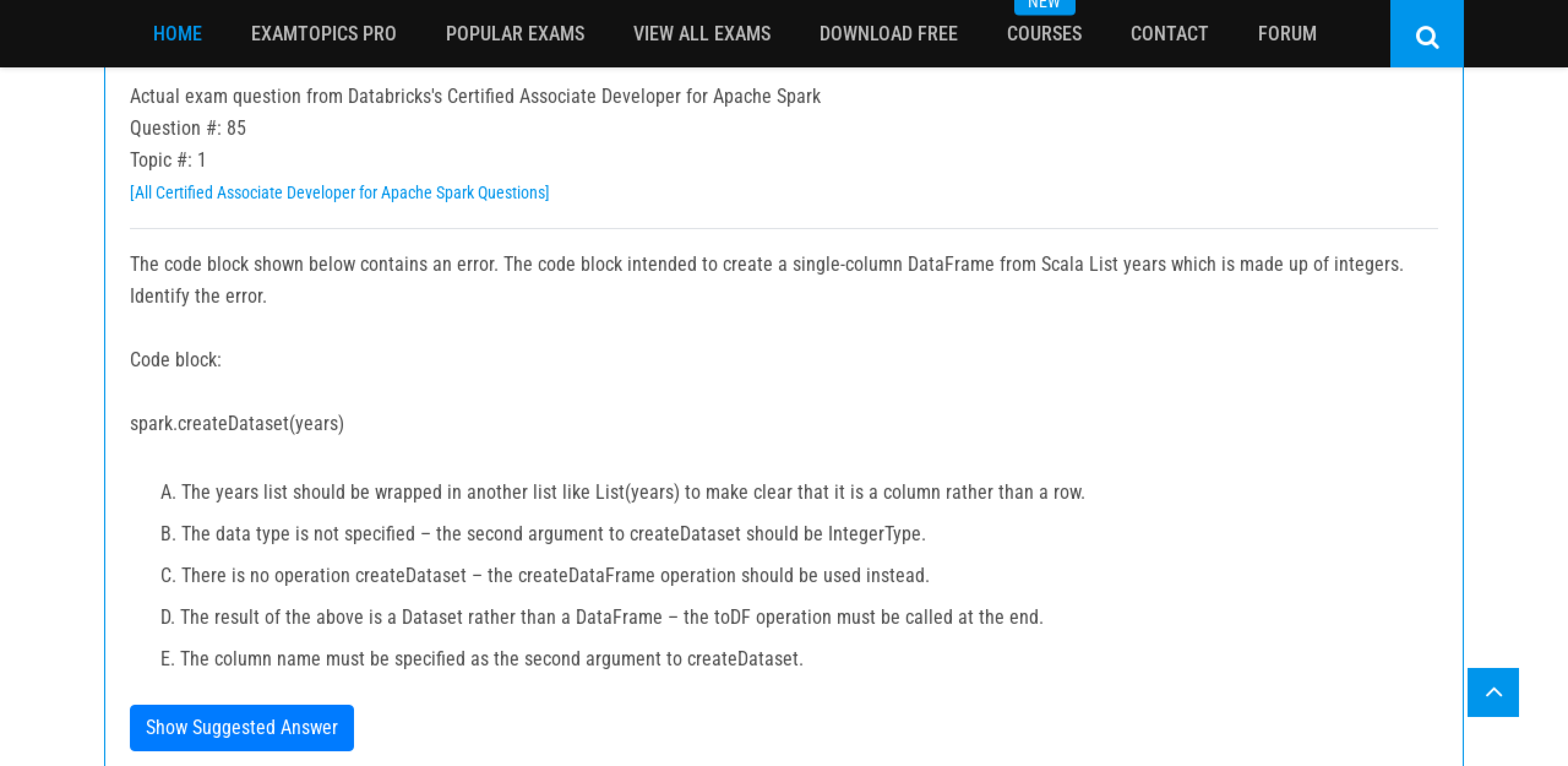
B. 1. spark
2. createTable
3. storesDF
4. sql

C. 1. storesDF
2. createOrReplaceTempView
3. spark
4. query

D. 1. spark
2. createOrReplaceTempView
3. storesDF
4. sql

E. 1. storesDF
2. createOrReplaceTempView
3. spark
4. sql

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 85

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block intended to create a single-column DataFrame from Scala List years which is made up of integers. Identify the error.

Code block:

```
spark.createDataset(years)
```

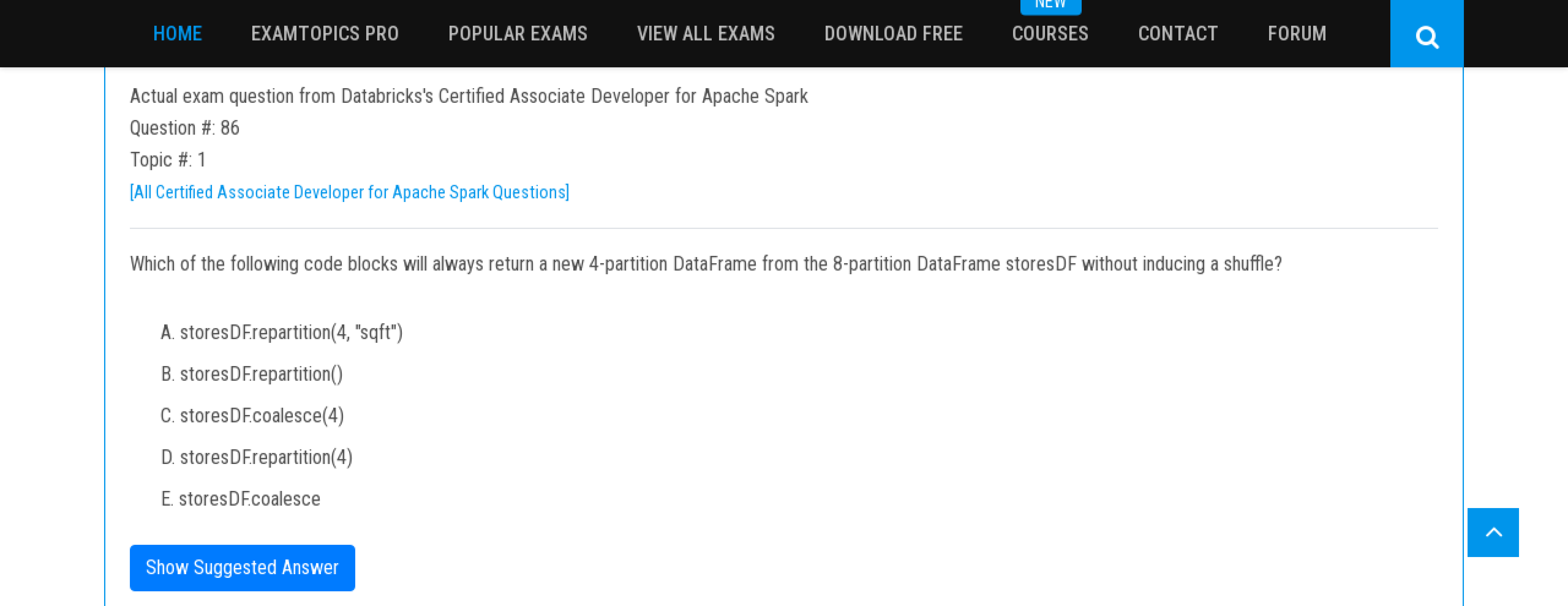
A. The years list should be wrapped in another list like List(years) to make clear that it is a column rather than a row.

B. The data type is not specified – the second argument to createDataset should be IntegerType.

C. There is no operation createDataset – the createDataFrame operation should be used instead.

D. The result of the above is a Dataset rather than a DataFrame – the toDF operation must be called at the end.

E. The column name must be specified as the second argument to createDataset.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 86

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks will always return a new 4-partition DataFrame from the 8-partition DataFrame storesDF without inducing a shuffle?

A. storesDF.repartition(4, "sqft")

B. storesDF.repartition()

C. storesDF.coalesce(4)

D. storesDF.repartition(4)

E. storesDF.coalesce

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 87

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should return a new 12-partition DataFrame from DataFrame storesDF. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

```
__1__.__2__(__3__)
```

A. 1. storesDF
2. coalesce
3. 4

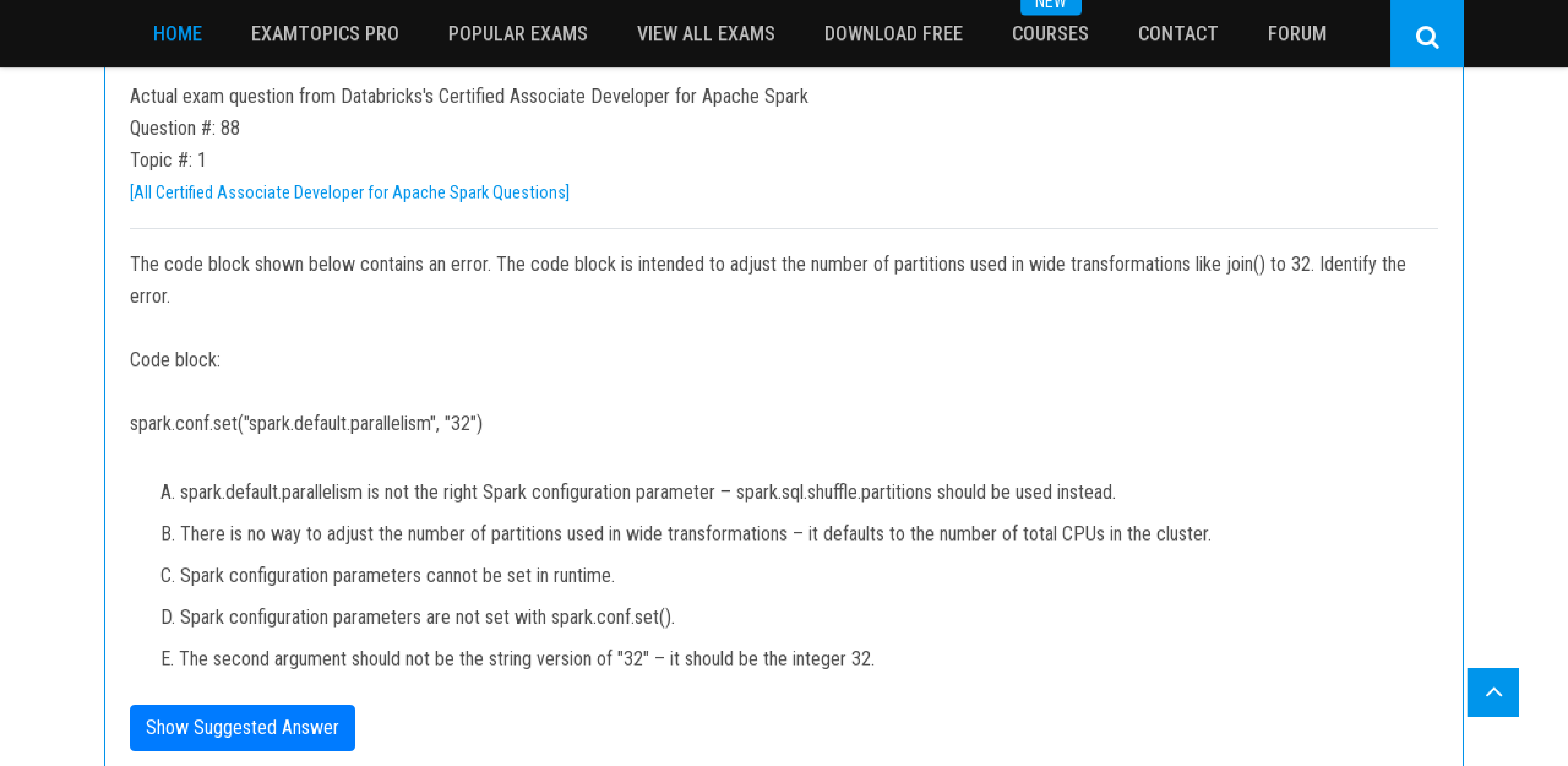
B. 1. storesDF
2. coalesce
3. 4, "storeId"

C. 1. storesDF
2. repartition
3. "storeId"

D. 1. storesDF
2. repartition
3. 12

E. 1. storesDF
2. repartition
3. Nothing

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 88

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to adjust the number of partitions used in wide transformations like join() to 32. Identify the error.

Code block:

```
spark.conf.set("spark.default.parallelism", "32")
```

A. spark.default.parallelism is not the right Spark configuration parameter – spark.sql.shuffle.partitions should be used instead.

B. There is no way to adjust the number of partitions used in wide transformations – it defaults to the number of total CPUs in the cluster.

C. Spark configuration parameters cannot be set in runtime.

D. Spark configuration parameters are not set with spark.conf.set().

E. The second argument should not be the string version of "32" – it should be the integer 32.

Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 89

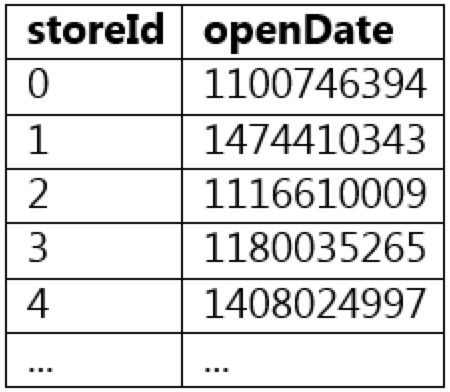
Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block intended to return a DataFrame containing a column dayOfYear, an integer representation of the day of the year from column openDate from DataFrame storesDF. Identify the error.

Note that column openDate is of type integer and represents a date in the UNIX epoch format – the number of seconds since midnight on January 1st, 1970.

A sample of storesDF is displayed below:



storeId	openDate
0	1100746394
1	1474410343
2	1116610009
3	1180035265
4	1408024997
...	...

Code block:

```
storesDF.withColumn("dayOfYear", dayofyear(col("openDate")))
```

A. The dayofyear() operation cannot extract the day of year from a column of type integer – column openDate must first be converted to type Timestamp.

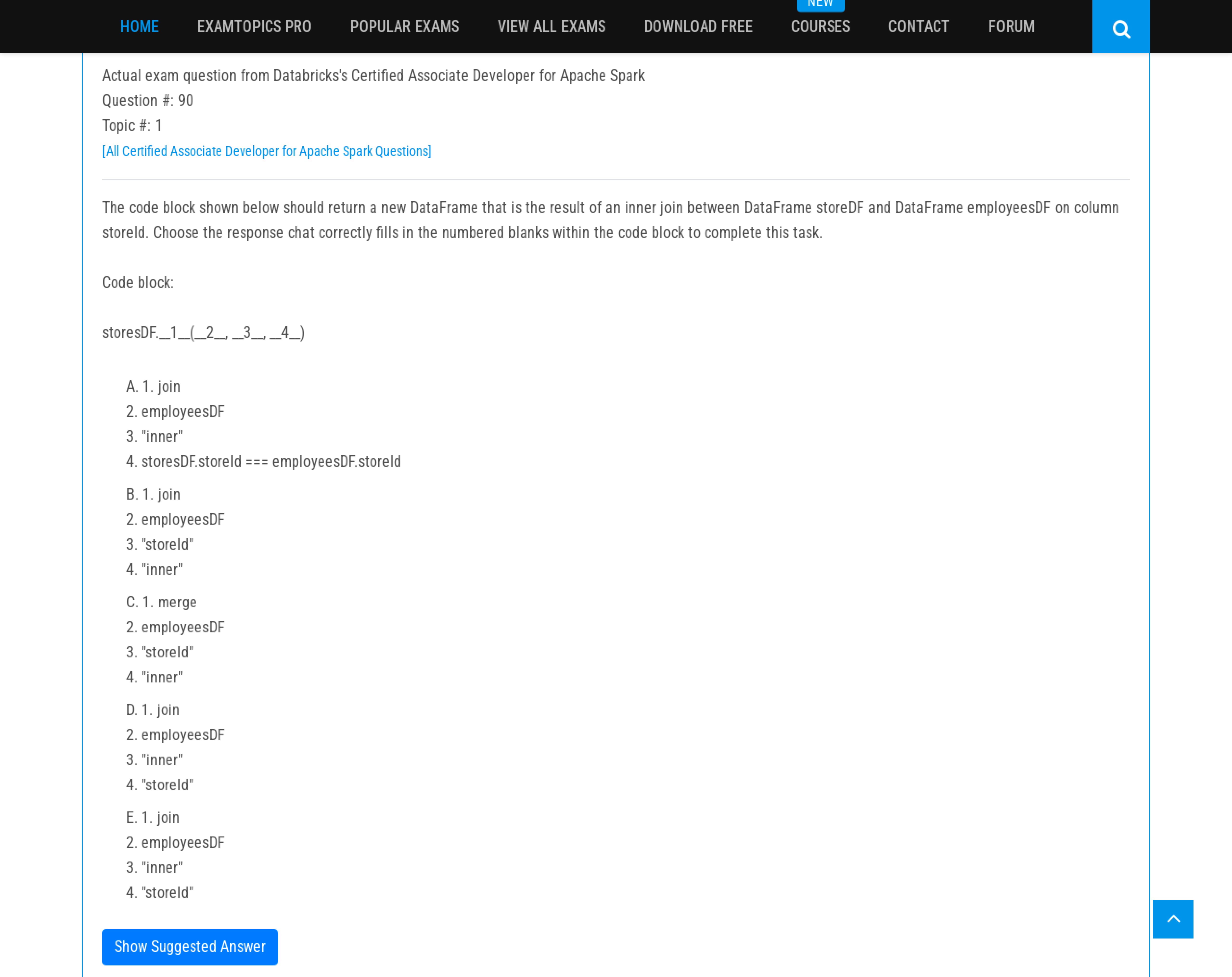
B. The dayofyear() operation takes a quoted column name rather than a Column object as its first argument – the first argument should be "openDate".

C. The dayofyear() operation cannot extract the day of year from a column of type integer – column openDate must first be converted to type Date.

D. The dayofyear() operation is not applicable in a withColumn() call – the newColumn() operation must be used instead.

E. There is no dayofyear() operation – the day of year number must be extracted using substring utilities.

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 90
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should return a new DataFrame that is the result of an inner join between DataFrame storeDF and DataFrame employeesDF on column storeId. Choose the response chat correctly fills in the numbered blanks within the code block to complete this task.

Code block:

```
storesDF.__1__(__2__, __3__, __4__)
```

A. 1. join
2. employeesDF
3. "inner"
4. storesDF.storeId === employeesDF.storeId

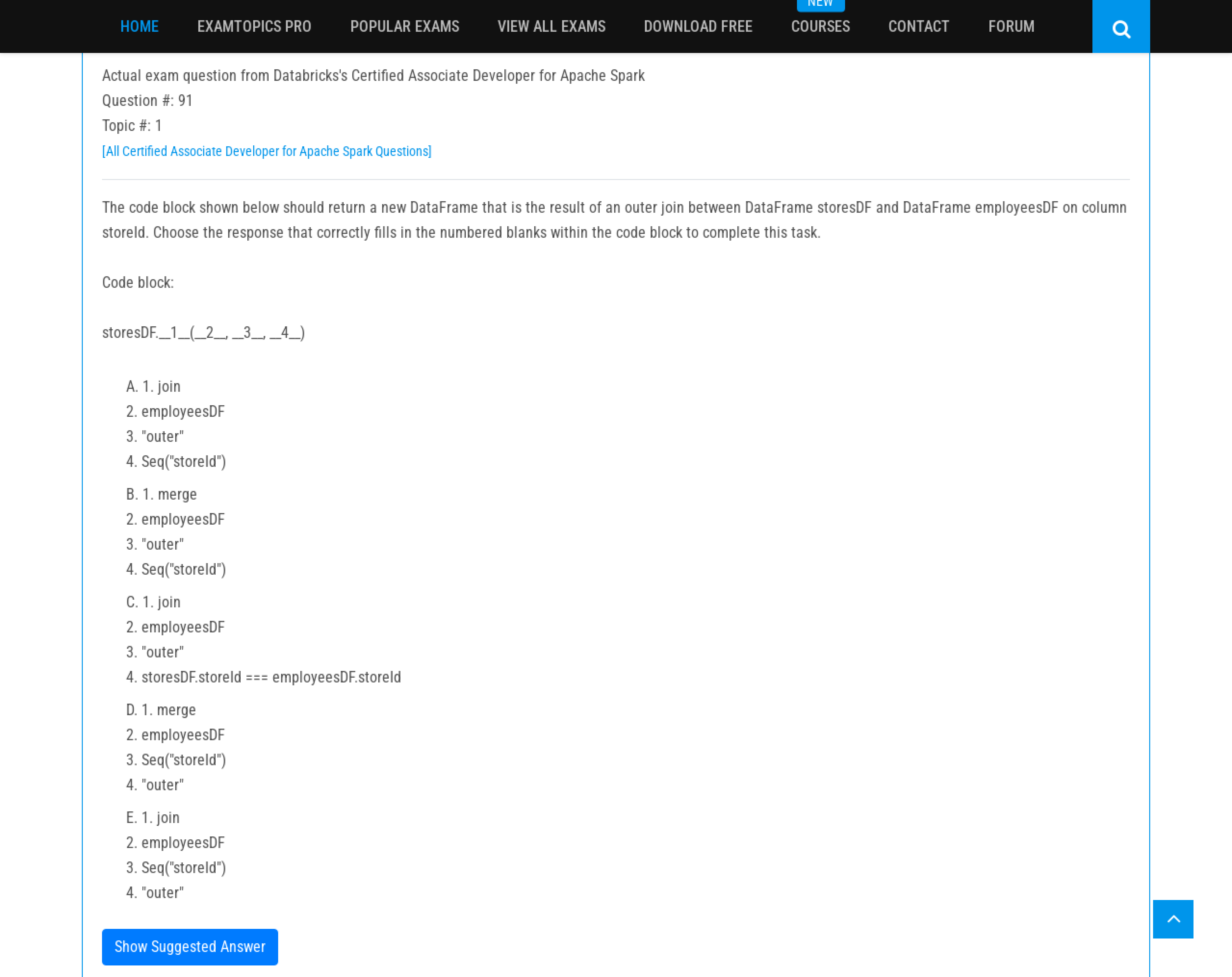
B. 1. join
2. employeesDF
3. "storeId"
4. "inner"

C. 1. merge
2. employeesDF
3. "storeId"
4. "inner"

D. 1. join
2. employeesDF
3. "inner"
4. "storeId"

E. 1. join
2. employeesDF
3. "inner"
4. "storeId"

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 91

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should return a new DataFrame that is the result of an outer join between DataFrame storesDF and DataFrame employeesDF on column storeId. 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__)
```

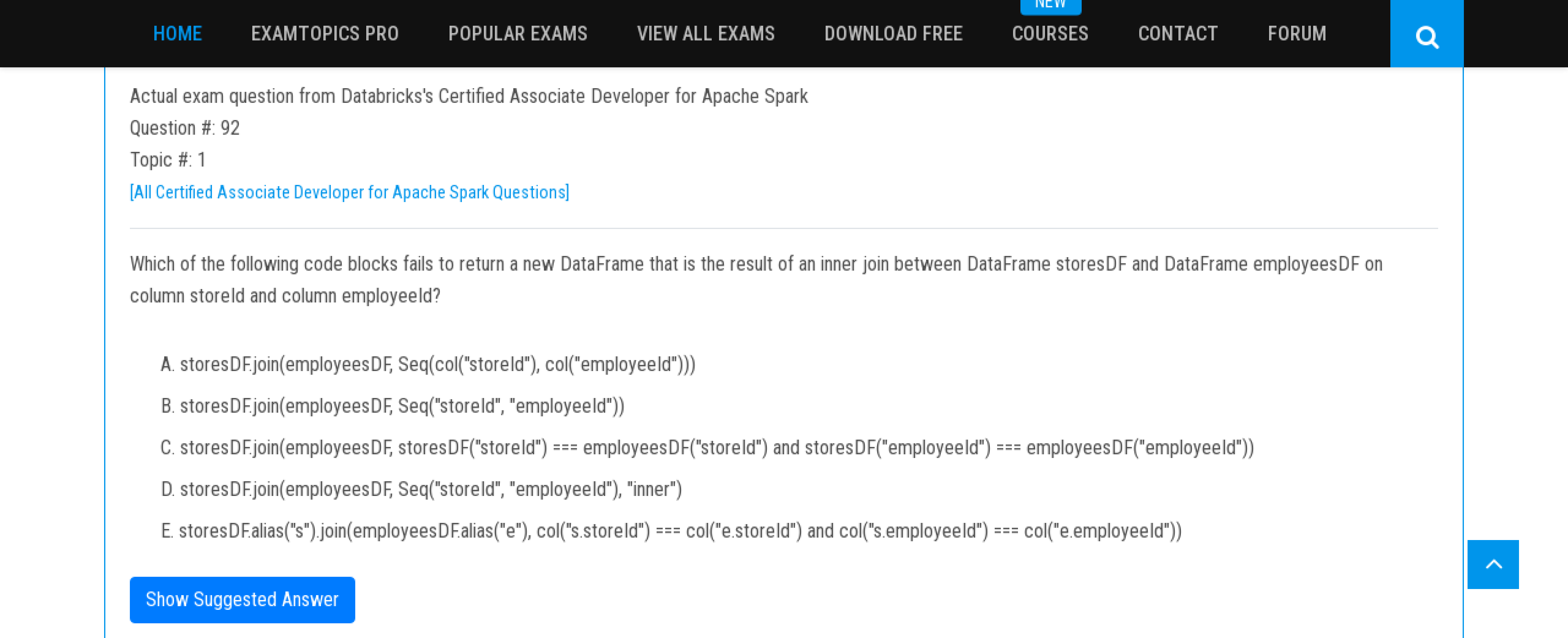
A. 1. join
2. employeesDF
3. "outer"
4. Seq("storeId")

B. 1. merge
2. employeesDF
3. "outer"
4. Seq("storeId")

C. 1. join
2. employeesDF
3. "outer"
4. storesDF.storeId === employeesDF.storeId

D. 1. merge
2. employeesDF
3. Seq("storeId")
4. "outer"

E. 1. join
2. employeesDF
3. Seq("storeId")
4. "outer"



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 92

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks fails to return a new DataFrame that is the result of an inner join between DataFrame storesDF and DataFrame employeesDF on column storeId and column employeeId?

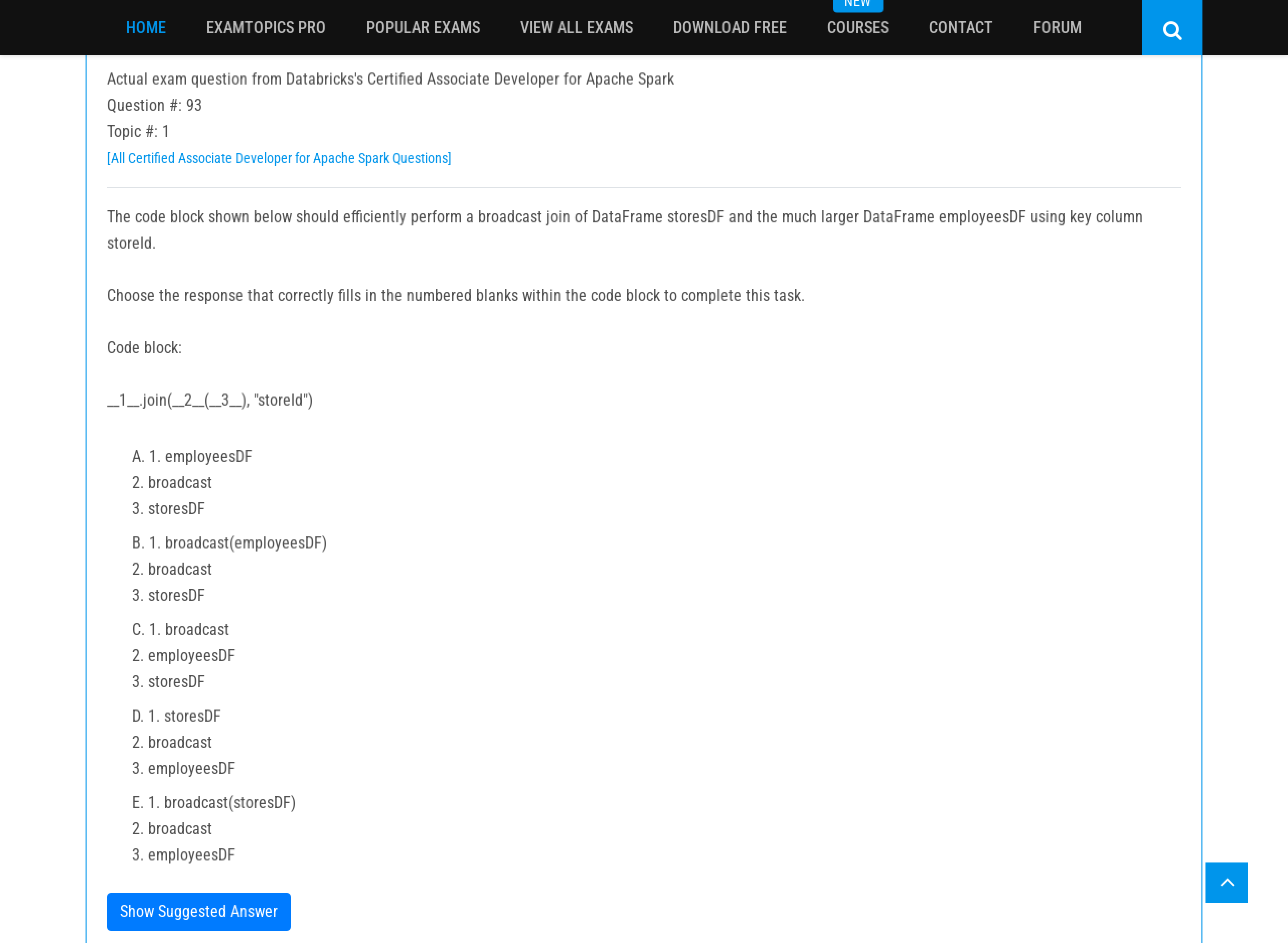
A. storesDF.join(employeesDF, Seq(col("storeId"), col("employeeId")))

B. storesDF.join(employeesDF, Seq("storeId", "employeeId"))

C. storesDF.join(employeesDF, storesDF("storeId") === employeesDF("storeId") and storesDF("employeeId") === employeesDF("employeeId"))

D. storesDF.join(employeesDF, Seq("storeId", "employeeId"), "inner")

E. storesDF.alias("s").join(employeesDF.alias("e"), col("s.storeId") === col("e.storeId") and col("s.employeeId") === col("e.employeeId"))



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 93

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should efficiently perform a broadcast join of DataFrame storesDF and the much larger DataFrame employeesDF using key column storeId.

Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

```
__1__.join(__2__(__3__), "storeId")
```

A. 1. employeesDF
2. broadcast
3. storesDF

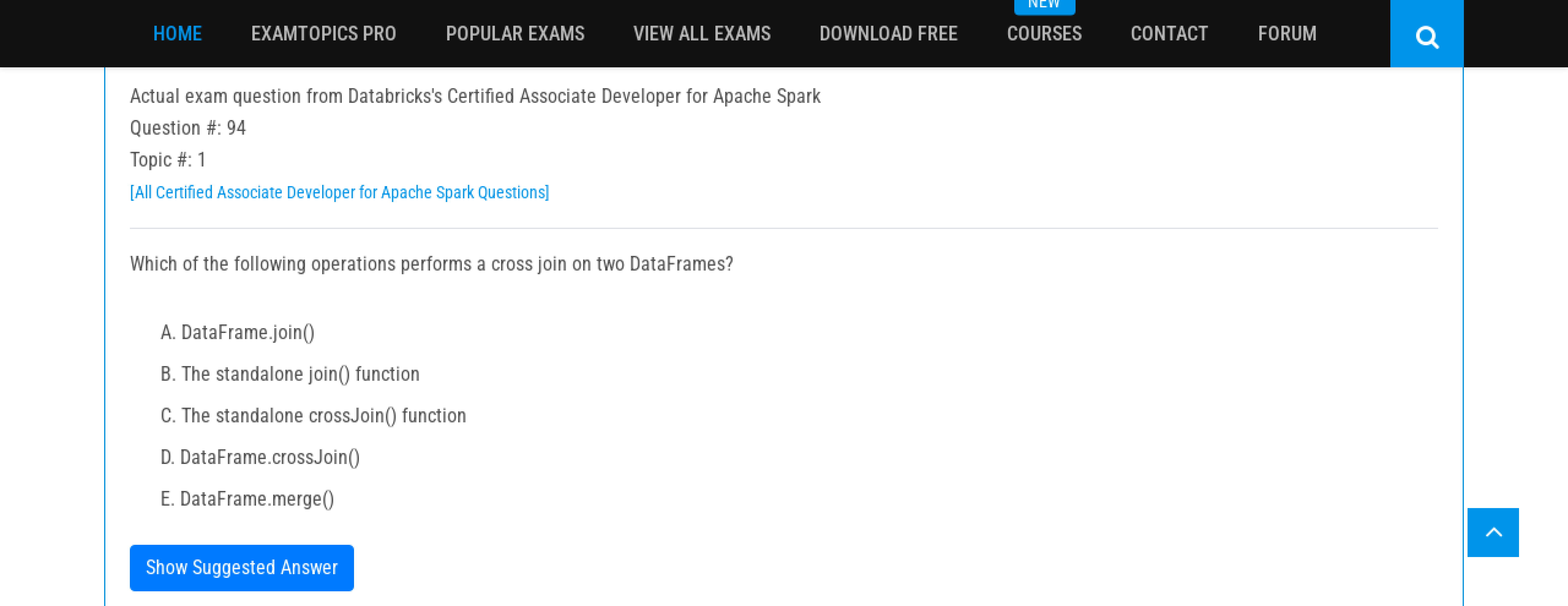
B. 1. broadcast(employeesDF)
2. broadcast
3. storesDF

C. 1. broadcast
2. employeesDF
3. storesDF

D. 1. storesDF
2. broadcast
3. employeesDF

E. 1. broadcast(storesDF)
2. broadcast
3. employeesDF

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 94

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following operations performs a cross join on two DataFrames?

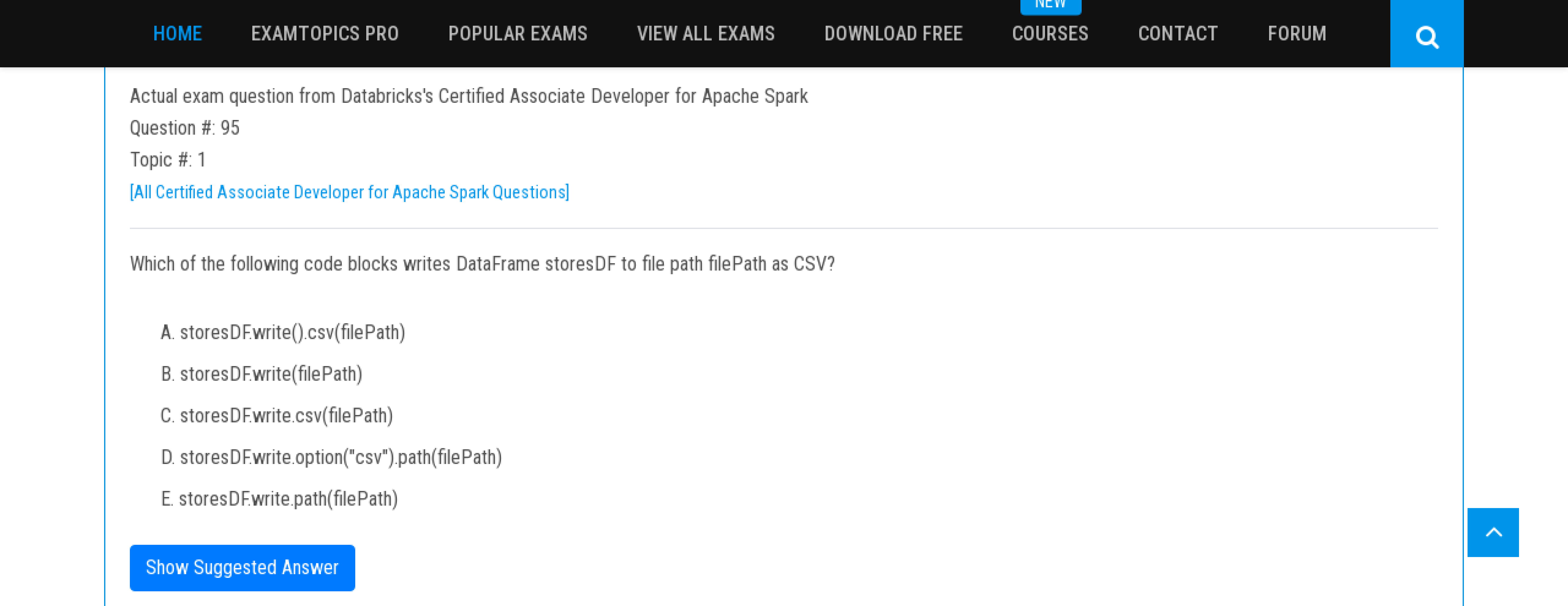
A. DataFrame.join()

B. The standalone join() function

C. The standalone crossJoin() function

D. DataFrame.crossJoin()

E. DataFrame.merge()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 95

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks writes DataFrame storesDF to file path filePath as CSV?

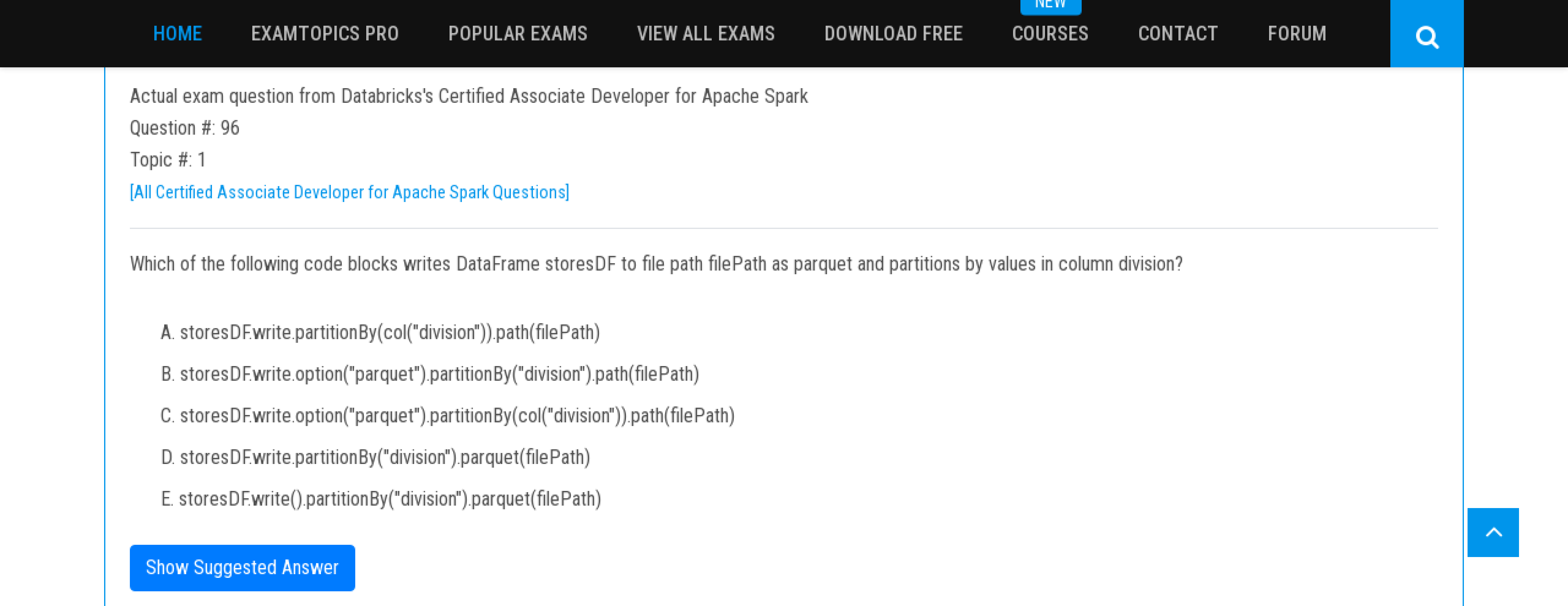
A. storesDF.write().csv(filePath)

B. storesDF.write(filePath)

C. storesDF.write.csv(filePath)

D. storesDF.write.option("csv").path(filePath)

E. storesDF.write.path(filePath)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 96

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks writes DataFrame storesDF to file path filePath as parquet and partitions by values in column division?

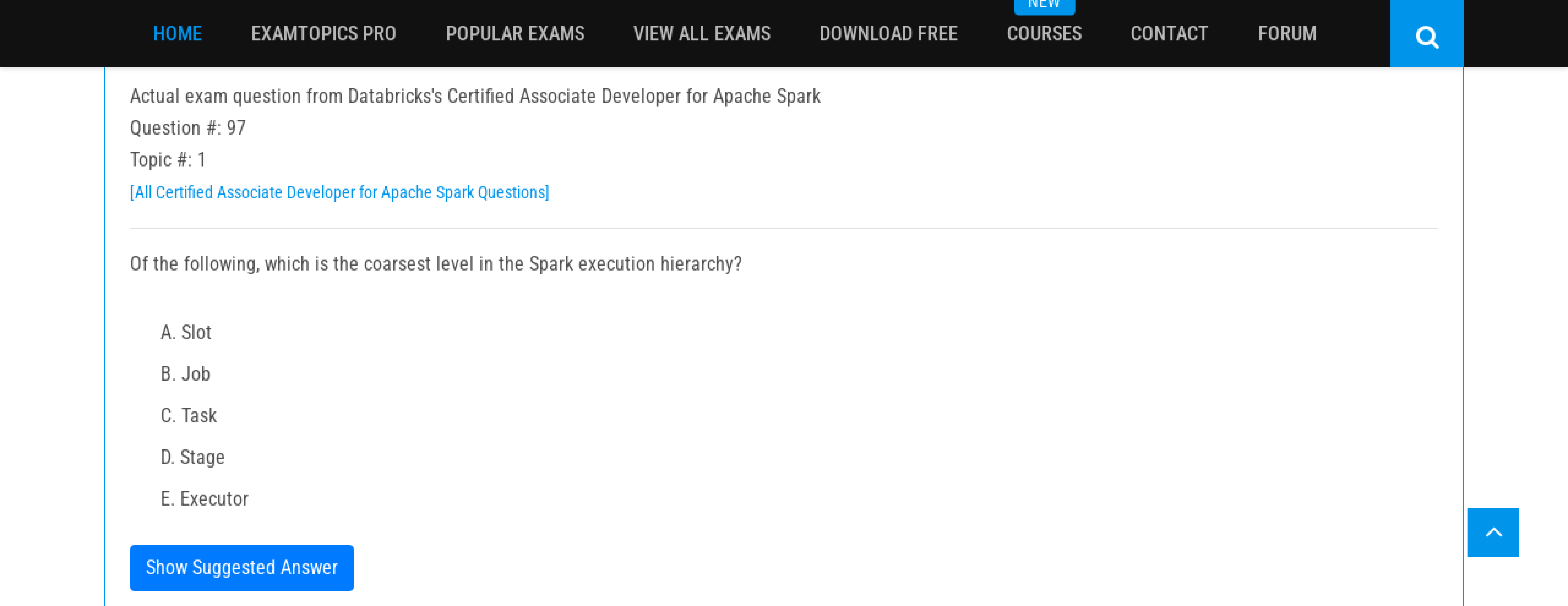
A. storesDF.write.partitionBy(col("division")).path(filePath)

B. storesDF.write.option("parquet").partitionBy("division").path(filePath)

C. storesDF.write.option("parquet").partitionBy(col("division")).path(filePath)

D. storesDF.write.partitionBy("division").parquet(filePath)

E. storesDF.write().partitionBy("division").parquet(filePath)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 97

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Of the following, which is the coarsest level in the Spark execution hierarchy?

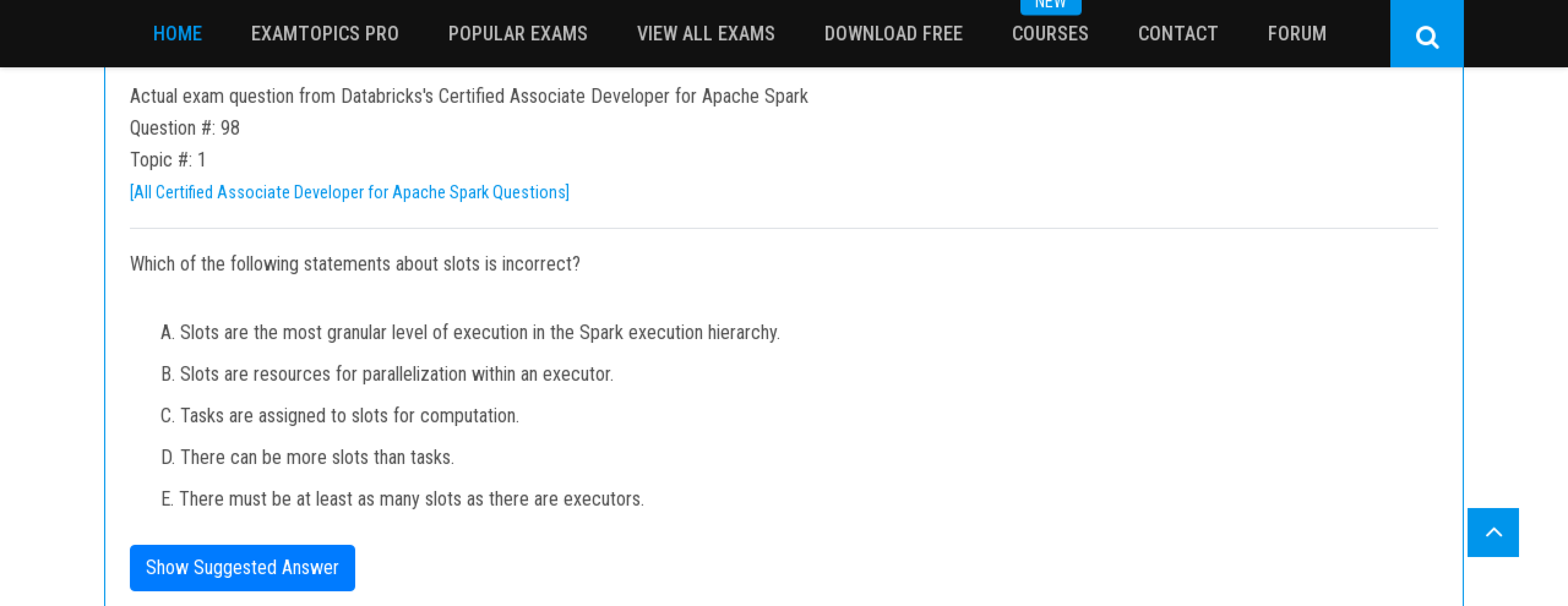
A. Slot

B. Job

C. Task

D. Stage

E. Executor



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 98

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following statements about slots is incorrect?

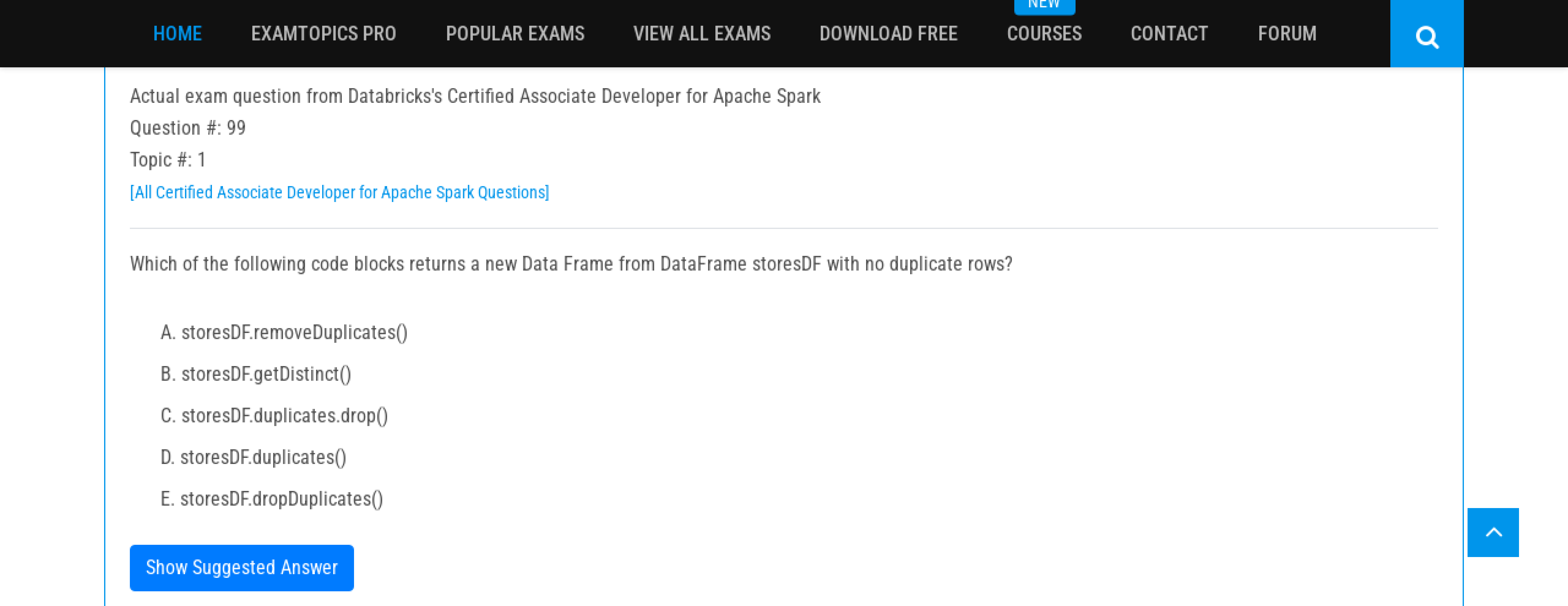
A. Slots are the most granular level of execution in the Spark execution hierarchy.

B. Slots are resources for parallelization within an executor.

C. Tasks are assigned to slots for computation.

D. There can be more slots than tasks.

E. There must be at least as many slots as there are executors.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 99

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns a new Data Frame from DataFrame storesDF with no duplicate rows?

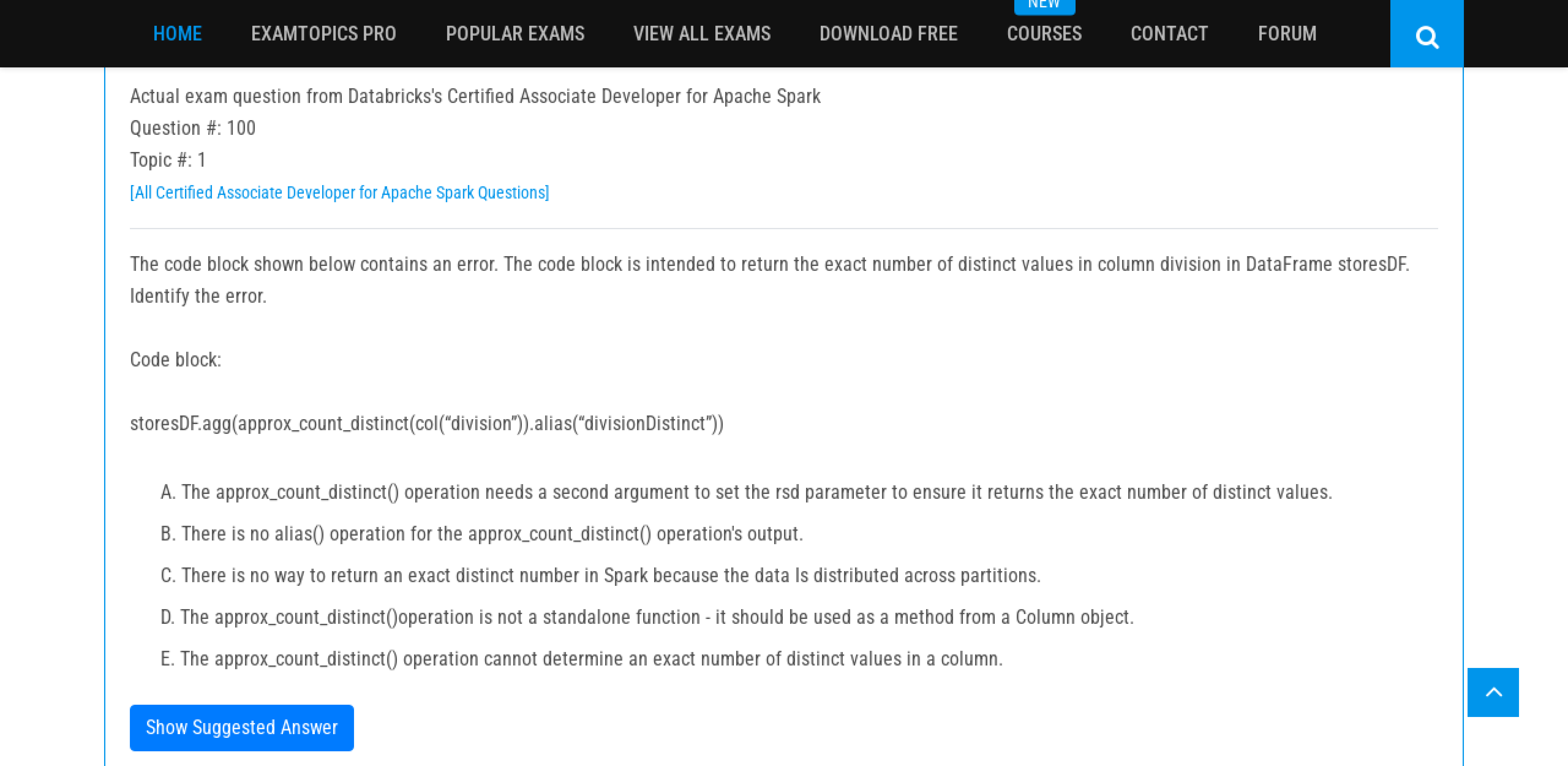
A. storesDF.removeDuplicates()

B. storesDF.getDistinct()

C. storesDF.duplicates.drop()

D. storesDF.duplicates()

E. storesDF.dropDuplicates()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 100

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to return the exact number of distinct values in column division in DataFrame storesDF. Identify the error.

Code block:

```
storesDF.agg(approx_count_distinct(col(“division”)).alias(“divisionDistinct”))
```

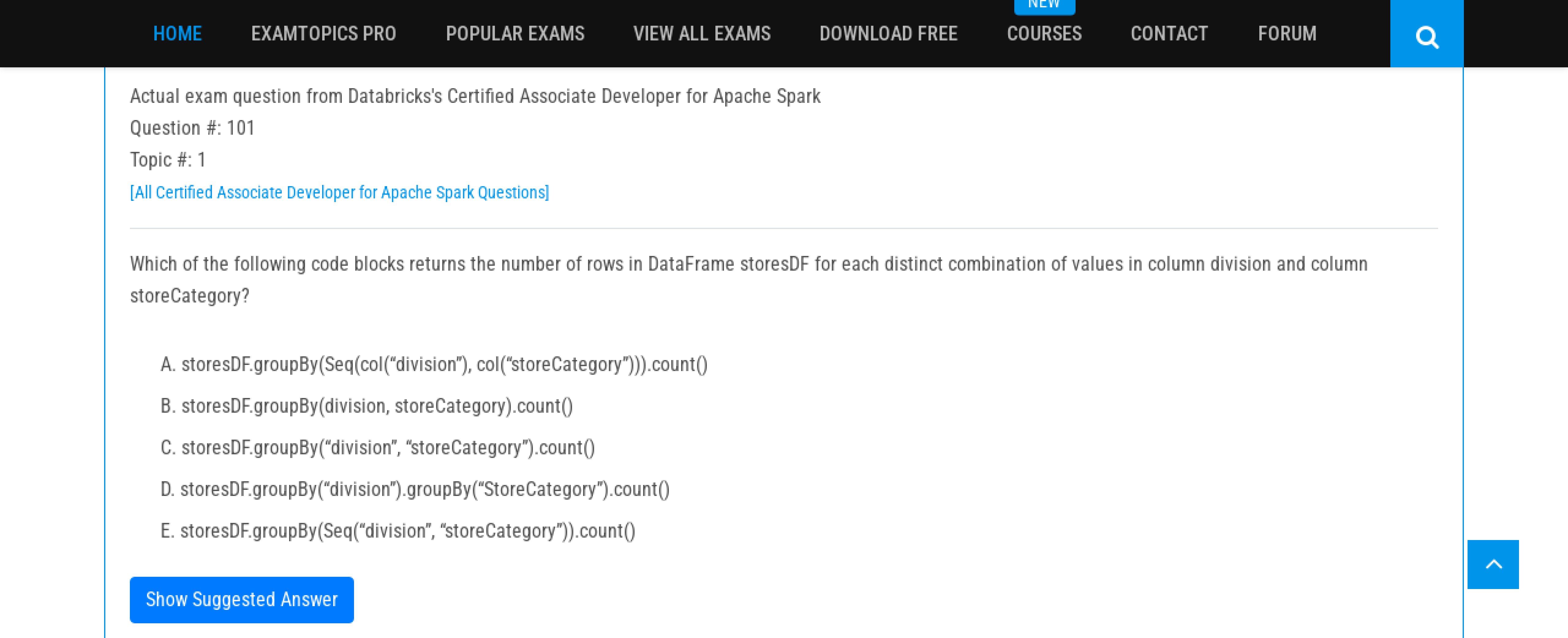
A. The approx_count_distinct() operation needs a second argument to set the rsd parameter to ensure it returns the exact number of distinct values.

B. There is no alias() operation for the approx_count_distinct() operation's output.

C. There is no way to return an exact distinct number in Spark because the data Is distributed across partitions.

D. The approx_count_distinct()operation is not a standalone function - it should be used as a method from a Column object.

E. The approx_count_distinct() operation cannot determine an exact number of distinct values in a column.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 101

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns the number of rows in DataFrame storesDF for each distinct combination of values in column division and column storeCategory?

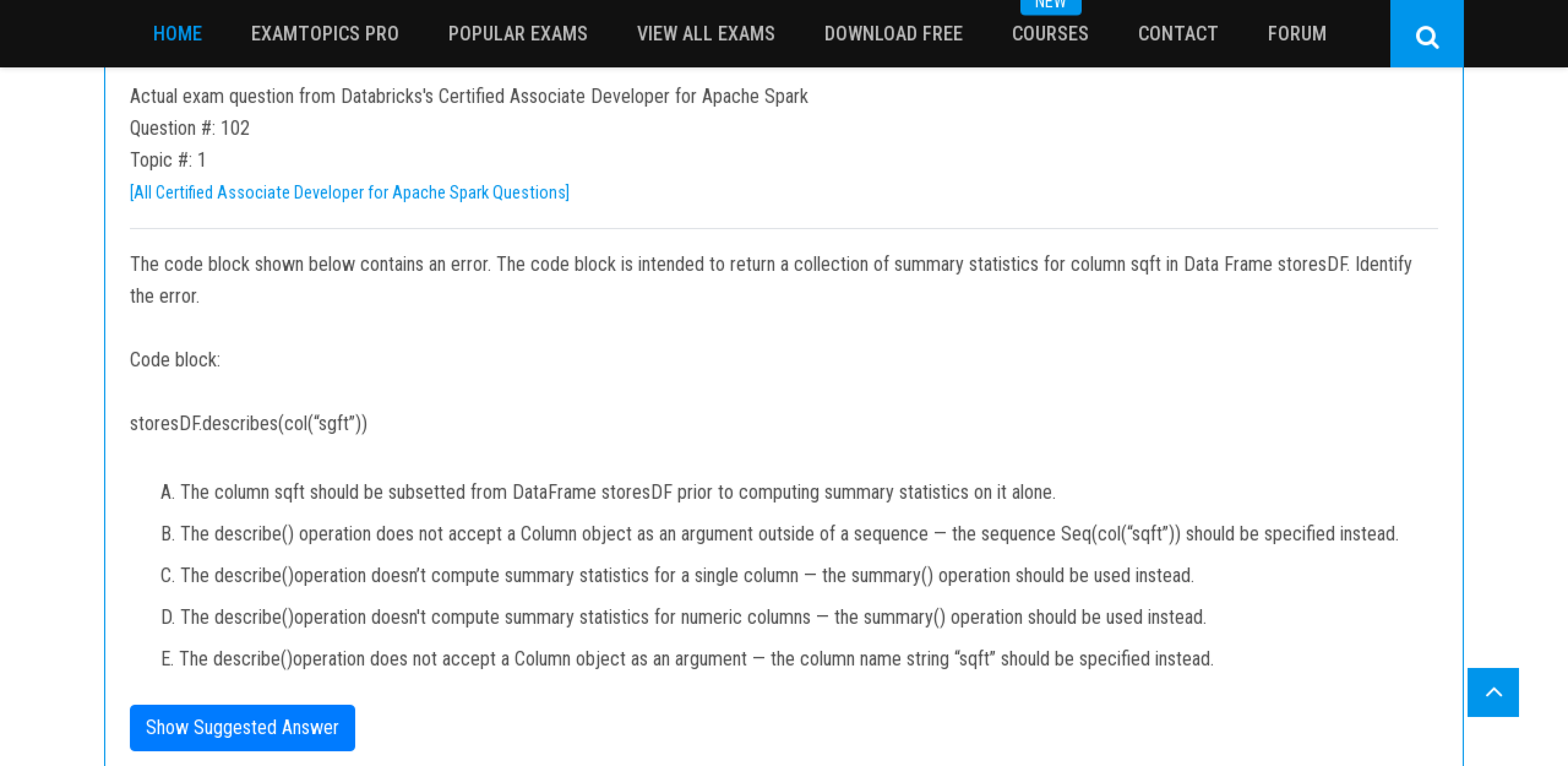
A. storesDF.groupBy(Seq(col(“division”), col(“storeCategory”))).count()

B. storesDF.groupBy(division, storeCategory).count()

C. storesDF.groupBy(“division”, “storeCategory”).count()

D. storesDF.groupBy(“division”).groupBy(“StoreCategory”).count()

E. storesDF.groupBy(Seq(“division”, “storeCategory”)).count()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 102

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to return a collection of summary statistics for column sqft in Data Frame storesDF. Identify the error.

Code block:

```
storesDF.describes(col(“sgft”))
```

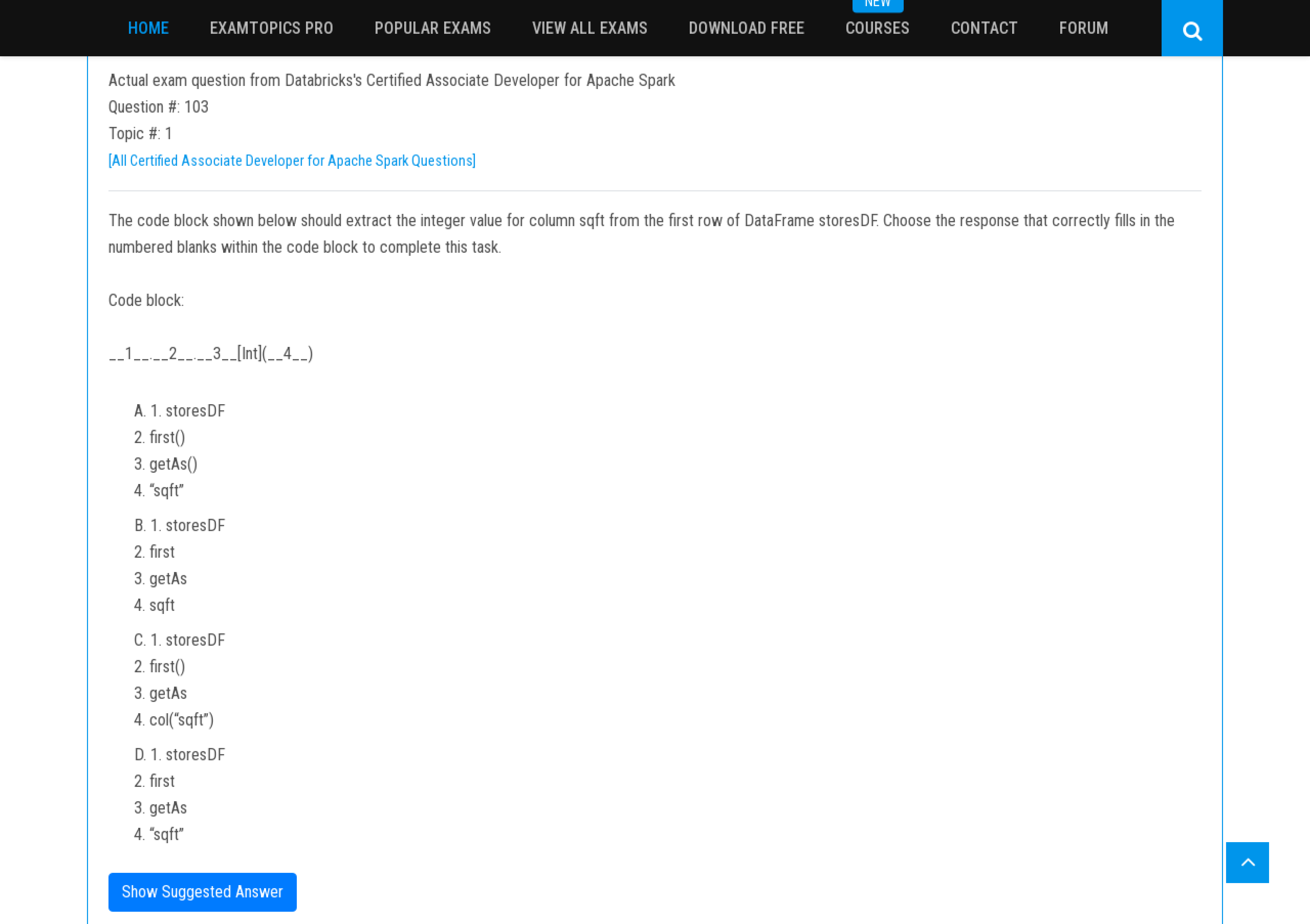
A. The column sqft should be subsetted from DataFrame storesDF prior to computing summary statistics on it alone.

B. The describe() operation does not accept a Column object as an argument outside of a sequence — the sequence Seq(col(“sqft”)) should be specified instead.

C. The describe()operation doesn’t compute summary statistics for a single column — the summary() operation should be used instead.

D. The describe()operation doesn't compute summary statistics for numeric columns — the summary() operation should be used instead.

E. The describe()operation does not accept a Column object as an argument — the column name string “sqft” should be specified instead.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 103
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should extract the integer value for column sqft from the first row of DataFrame storesDF. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

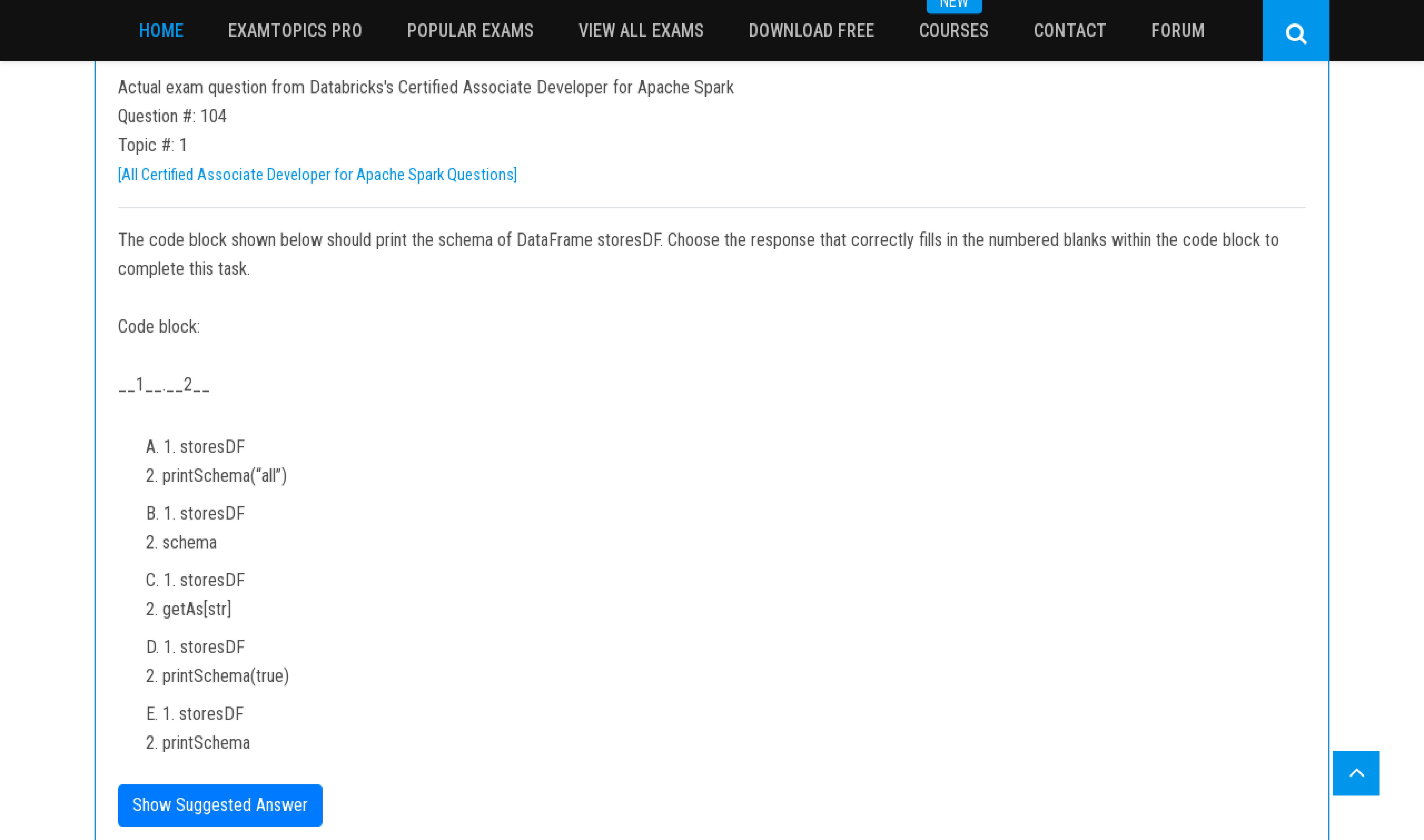
```
__1__.__2__.__3__[Int](__4__)
```

A. 1. storesDF
2. first()
3. getAs()
4. “sqft”

B. 1. storesDF
2. first
3. getAs
4. sqft

C. 1. storesDF
2. first()
3. getAs
4. col(“sqft”)

D. 1. storesDF
2. first
3. getAs
4. “sqft”



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 104

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should print the schema of DataFrame storesDF. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

```
__1__.__2__
```

A. 1. storesDF
2. printSchema(“all”)

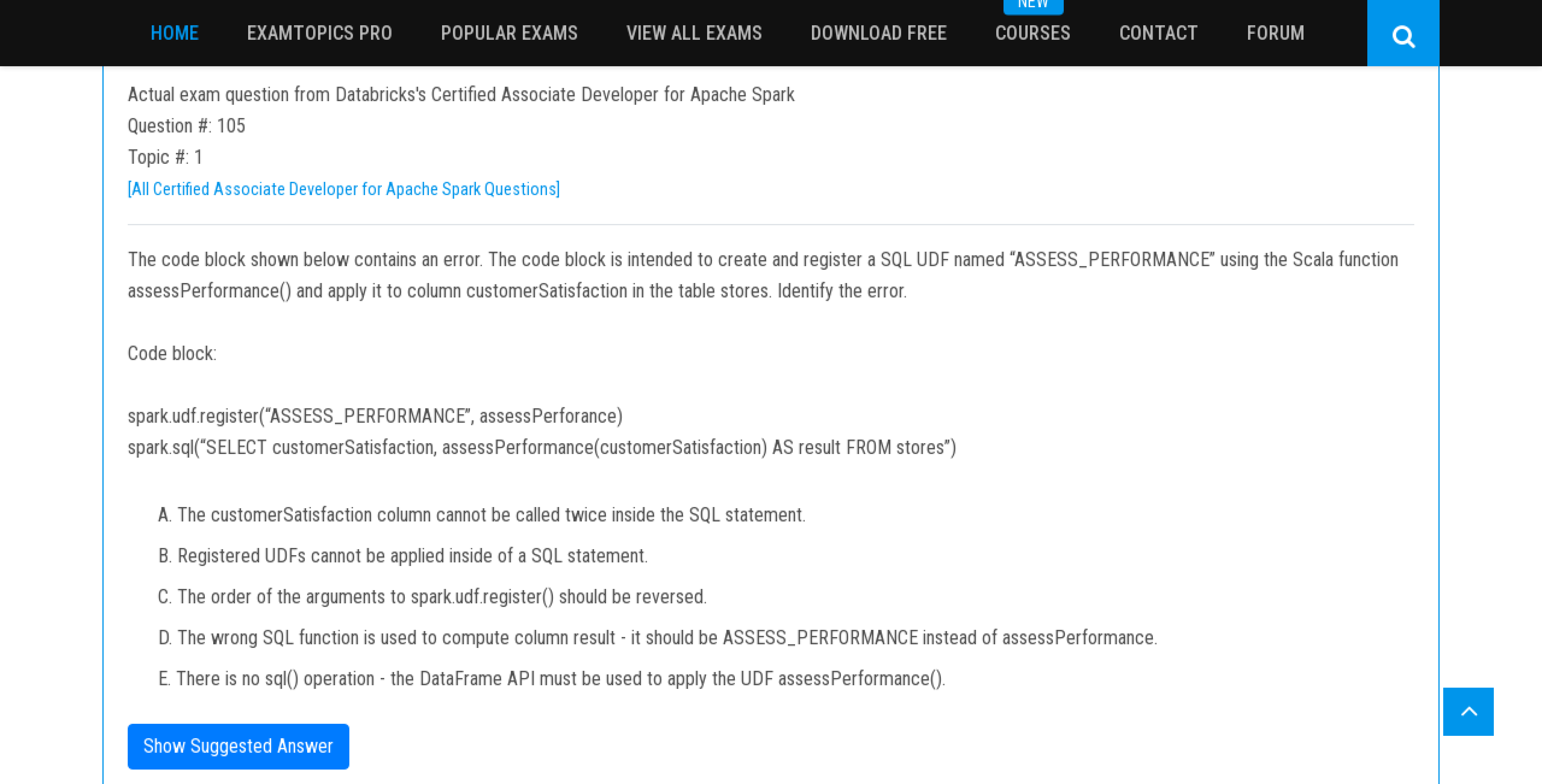
B. 1. storesDF
2. schema

C. 1. storesDF
2. getAs[str]

D. 1. storesDF
2. printSchema(true)

E. 1. storesDF
2. printSchema

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 105

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to create and register a SQL UDF named “ASSESS_PERFORMANCE” using the Scala function assessPerformance() and apply it to column customerSatisfaction in the table stores. Identify the error.

Code block:

```
spark.udf.register(“ASSESS_PERFORMANCE”, assessPerforance)
spark.sql(“SELECT customerSatisfaction, assessPerformance(customerSatisfaction) AS result FROM stores”)
```

A. The customerSatisfaction column cannot be called twice inside the SQL statement.

B. Registered UDFs cannot be applied inside of a SQL statement.

C. The order of the arguments to spark.udf.register() should be reversed.

D. The wrong SQL function is used to compute column result - it should be ASSESS_PERFORMANCE instead of assessPerformance.

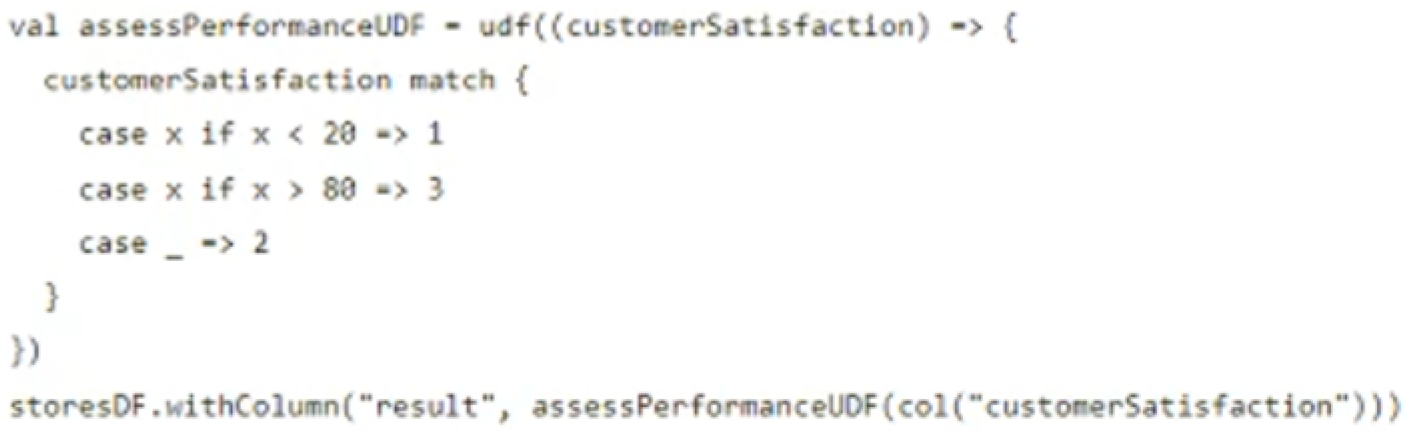
E. There is no sql() operation - the DataFrame API must be used to apply the UDF assessPerformance().

Show Suggested Answer

Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 106
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to create the Scala UDF assessPerformanceUDF() and apply it to the integer column customers1t1sfaction in Data Frame storesDF. Identify the error.

Code block:



```
val assessPerformanceUDF = udf((customerSatisfaction) => {
  customerSatisfaction match {
    case x if x < 20 => 1
    case x if x > 80 => 3
    case _ => 2
  }
})
storesDF.withColumn("result", assessPerformanceUDF(col("customerSatisfaction")))
```

A. The input type of customerSatisfaction is not specified in the udf() operation.

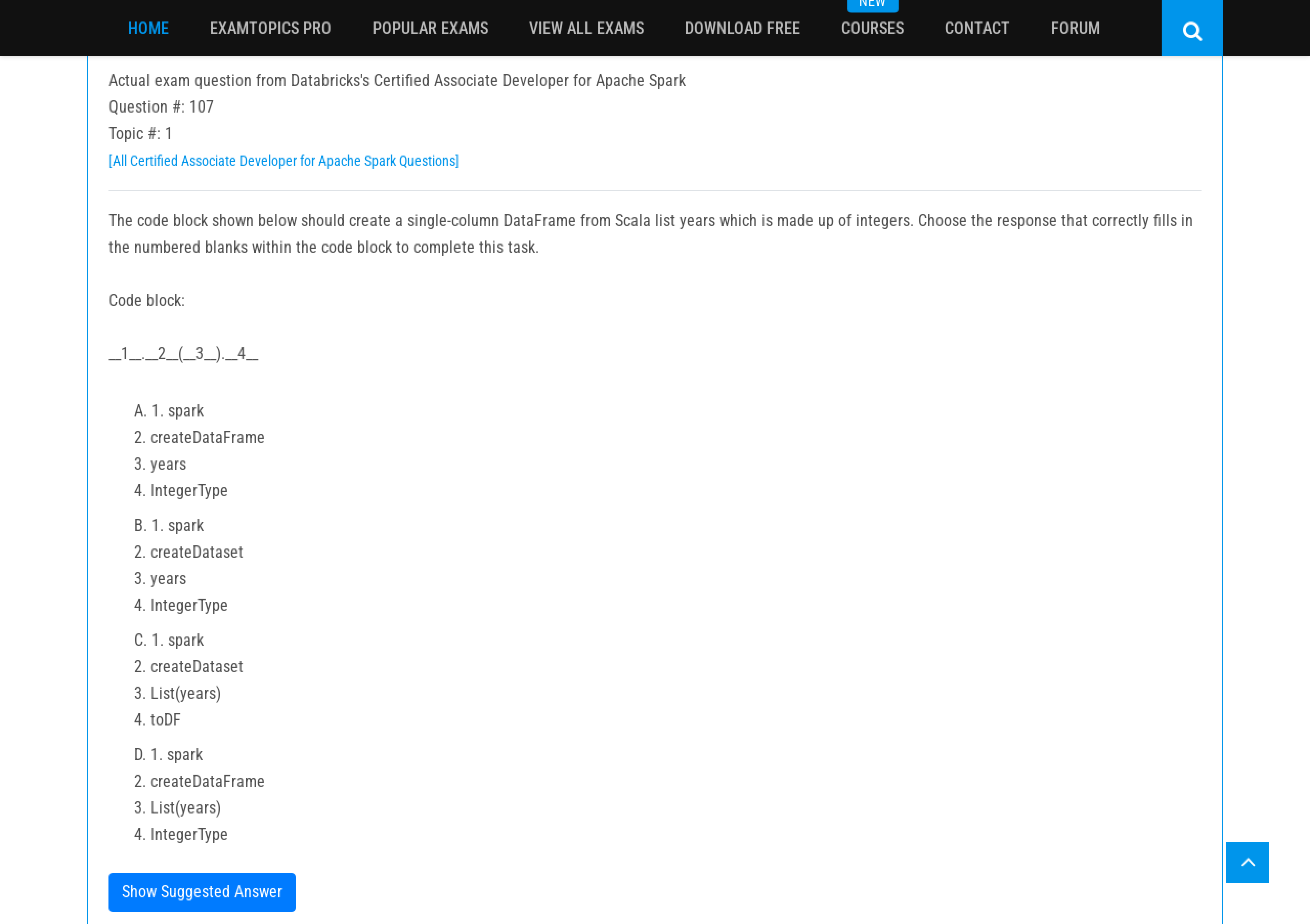
B. The return type of assessPerformanceUDF() must be specified.

C. The withColumn() operation is not appropriate here - UDFs should be applied by iterating over rows instead.

D. The assessPerformanceUDF() must first be defined as a Scala function and then converted to a UDF.

E. UDFs can only be applied via SQL and not through the Data Frame API.

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 107
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should create a single-column DataFrame from Scala list years which is made up of integers. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

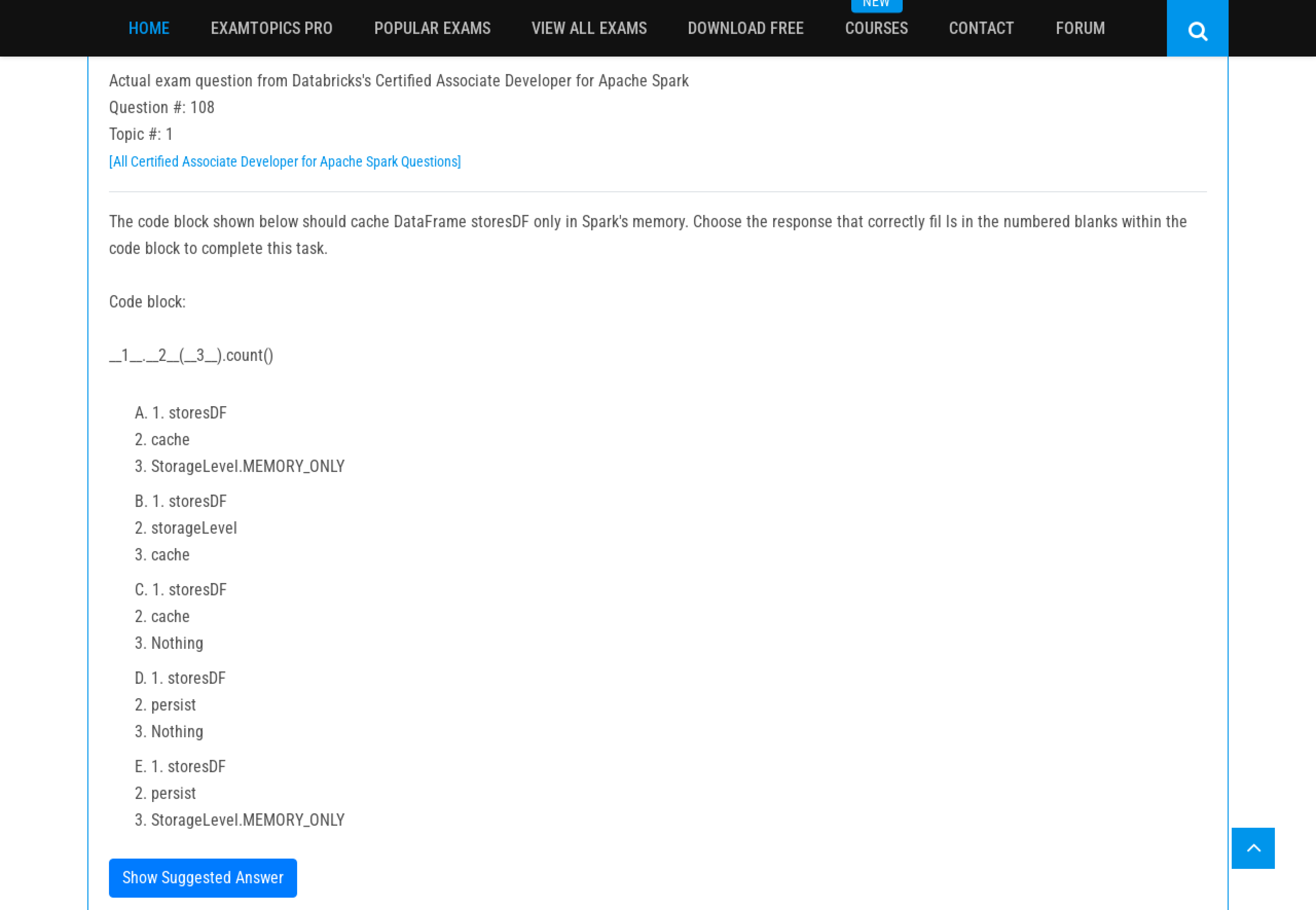
```
__1__.__2__(__3__).__4__
```

A. 1. spark
2. createDataFrame
3. years
4. IntegerType

B. 1. spark
2. createDataset
3. years
4. IntegerType

C. 1. spark
2. createDataset
3. List(years)
4. toDF

D. 1. spark
2. createDataFrame
3. List(years)
4. IntegerType



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 108
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should cache DataFrame storesDF only in Spark's memory. Choose the response that correctly fil ls in the numbered blanks within the code block to complete this task.

Code block:

```
__1__.__2__(__3__).count()
```

A. 1. storesDF
2. cache
3. StorageLevel.MEMORY_ONLY

B. 1. storesDF
2. storageLevel
3. cache

C. 1. storesDF
2. cache
3. Nothing

D. 1. storesDF
2. persist
3. Nothing

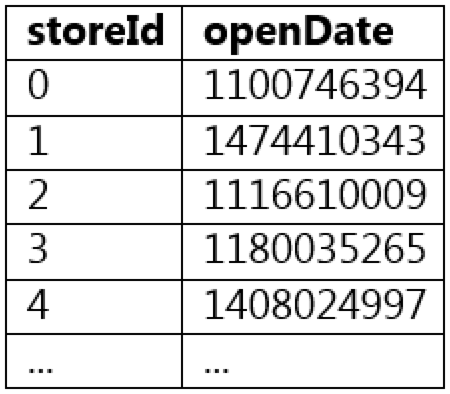
E. 1. storesDF
2. persist
3. StorageLevel.MEMORY_ONLY

Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 109
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks returns a DataFrame containing a column month, an integer representation of the day of the year from column openDate from DataFrame storesDF.

Note that column openDate is of type integer and represents a date in the UNIX epoch format – the number of seconds since midnight on January 1st, 1970.

A sample of storesDF is displayed below:



storeId	openDate
0	1100746394
1	1474410343
2	1116610009
3	1180035265
4	1408024997
...	...

Code block:

```
stored.withColumn("openTimestamp", col("openDate").cast(__1__))
.withColumn(__2__, __3__(__4__))
```

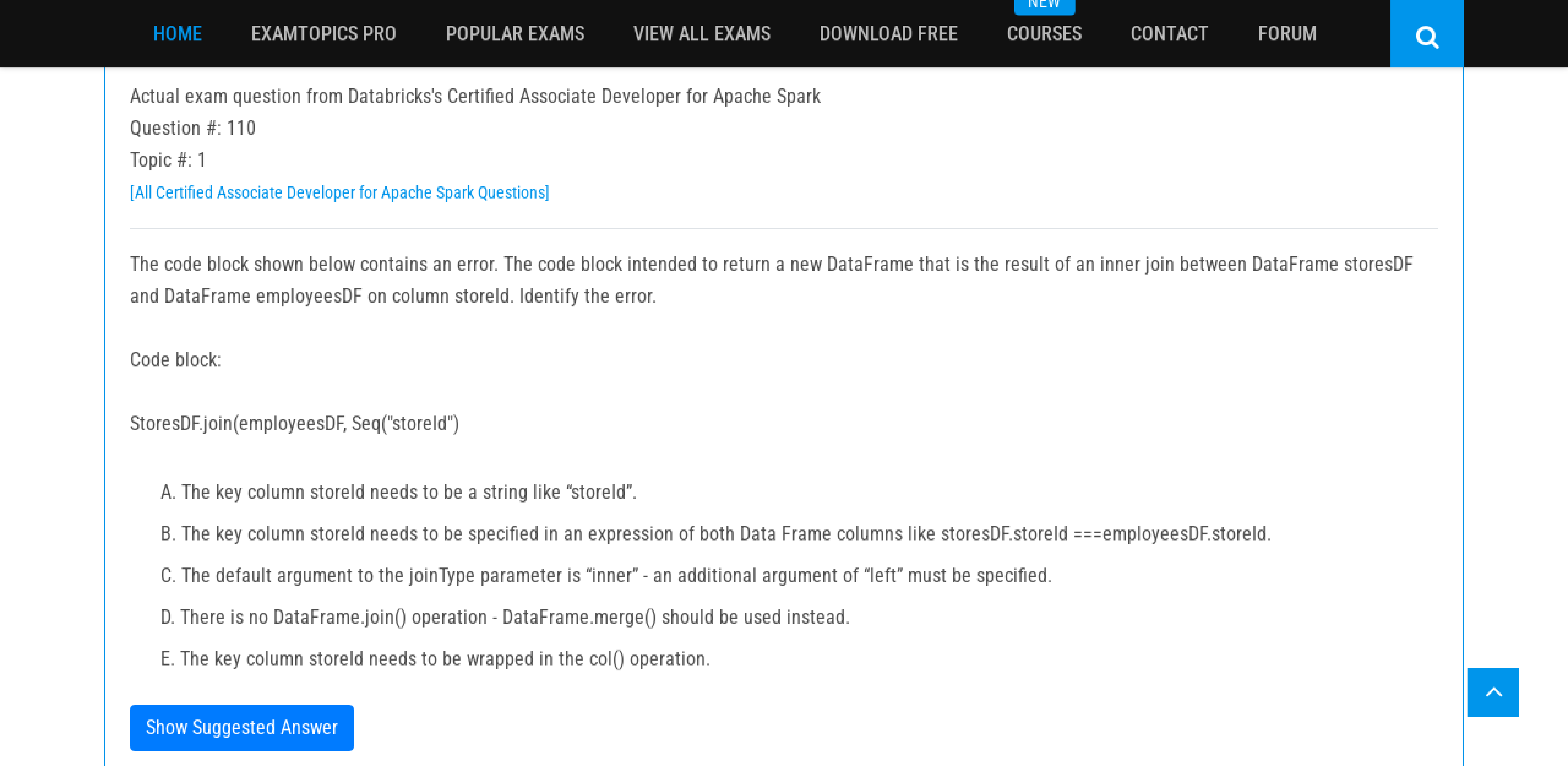
A. 1. "Data"
2. month
3. "month"
4. "openTimestamp"

B. 1. "Timestamp"
2. month
3. "month"
4. col("openTimestamp")

C. 1. "Timestamp"
2. month
3. getMonth
4. col("openTimestamp")

D. 1. "Timestamp"
2. "month"
3. month
4. col("openTimestamp")

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 110
Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block intended to return a new DataFrame that is the result of an inner join between DataFrame storesDF and DataFrame employeesDF on column storeId. Identify the error.

Code block:

```
StoresDF.join(employeesDF, Seq("storeId")
```

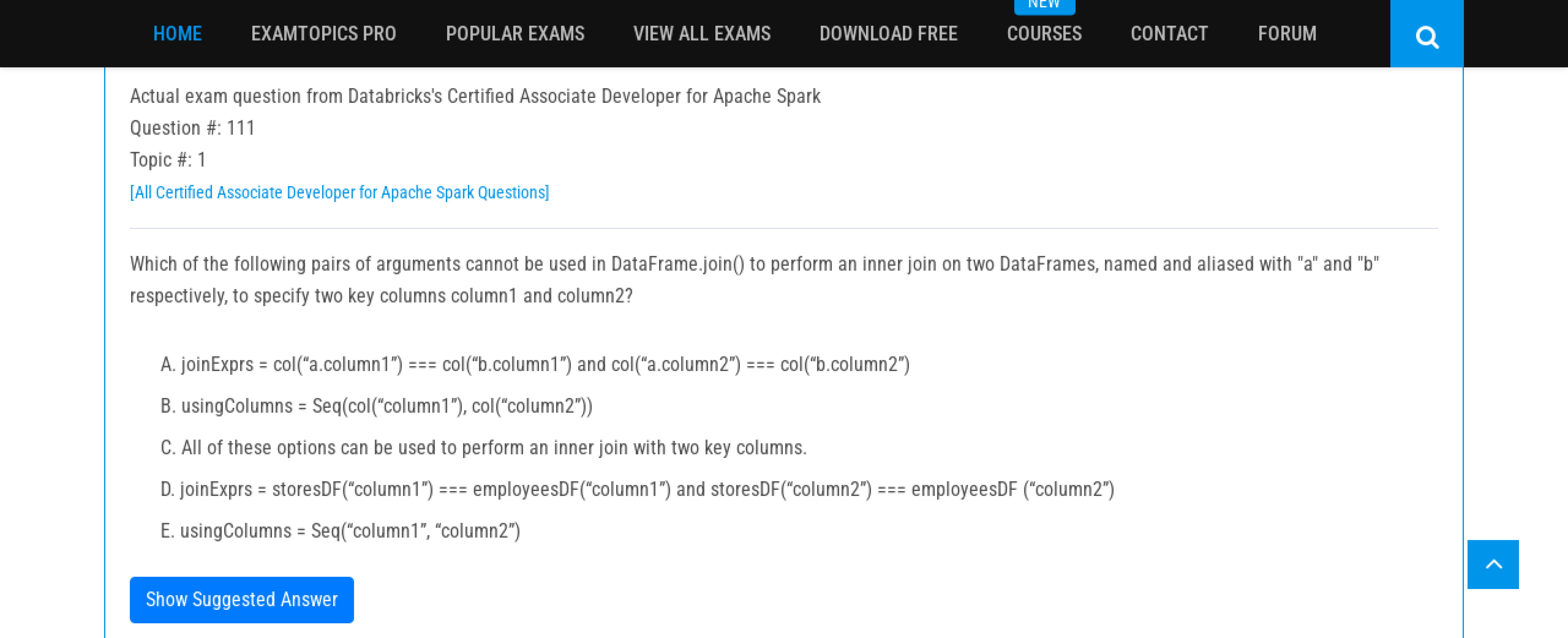
A. The key column storeId needs to be a string like “storeId”.

B. The key column storeId needs to be specified in an expression of both Data Frame columns like storesDF.storeId ===employeesDF.storeId.

C. The default argument to the joinType parameter is “inner” - an additional argument of “left” must be specified.

D. There is no DataFrame.join() operation - DataFrame.merge() should be used instead.

E. The key column storeId needs to be wrapped in the col() operation.



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 111
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

Which of the following pairs of arguments cannot be used in DataFrame.join() to perform an inner join on two DataFrames, named and aliased with "a" and "b" respectively, to specify two key columns column1 and column2?

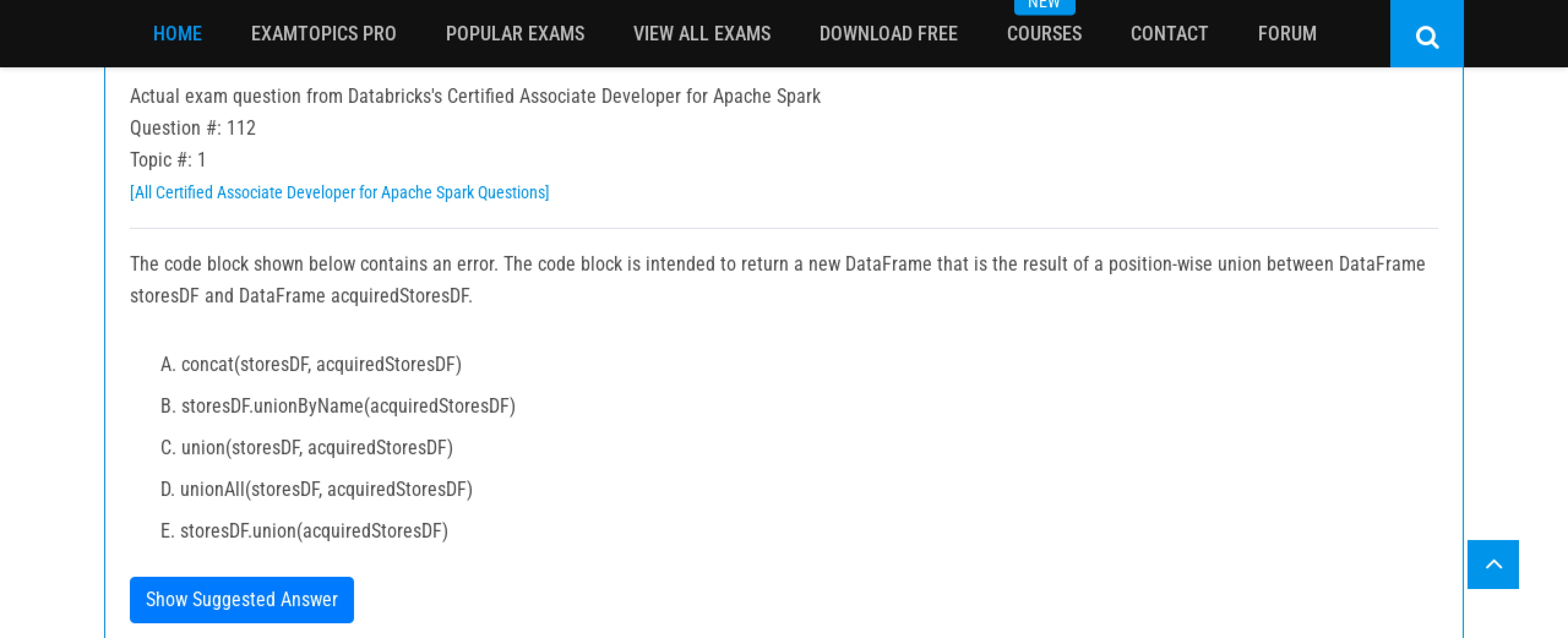
A. joinExprs = col(“a.column1”) === col(“b.column1”) and col(“a.column2”) === col(“b.column2”)

B. usingColumns = Seq(col(“column1”), col(“column2”))

C. All of these options can be used to perform an inner join with two key columns.

D. joinExprs = storesDF(“column1”) === employeesDF(“column1”) and storesDF(“column2”) === employeesDF (“column2”)

E. usingColumns = Seq(“column1”, “column2”)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 112

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

The code block shown below contains an error. The code block is intended to return a new DataFrame that is the result of a position-wise union between DataFrame storesDF and DataFrame acquiredStoresDF.

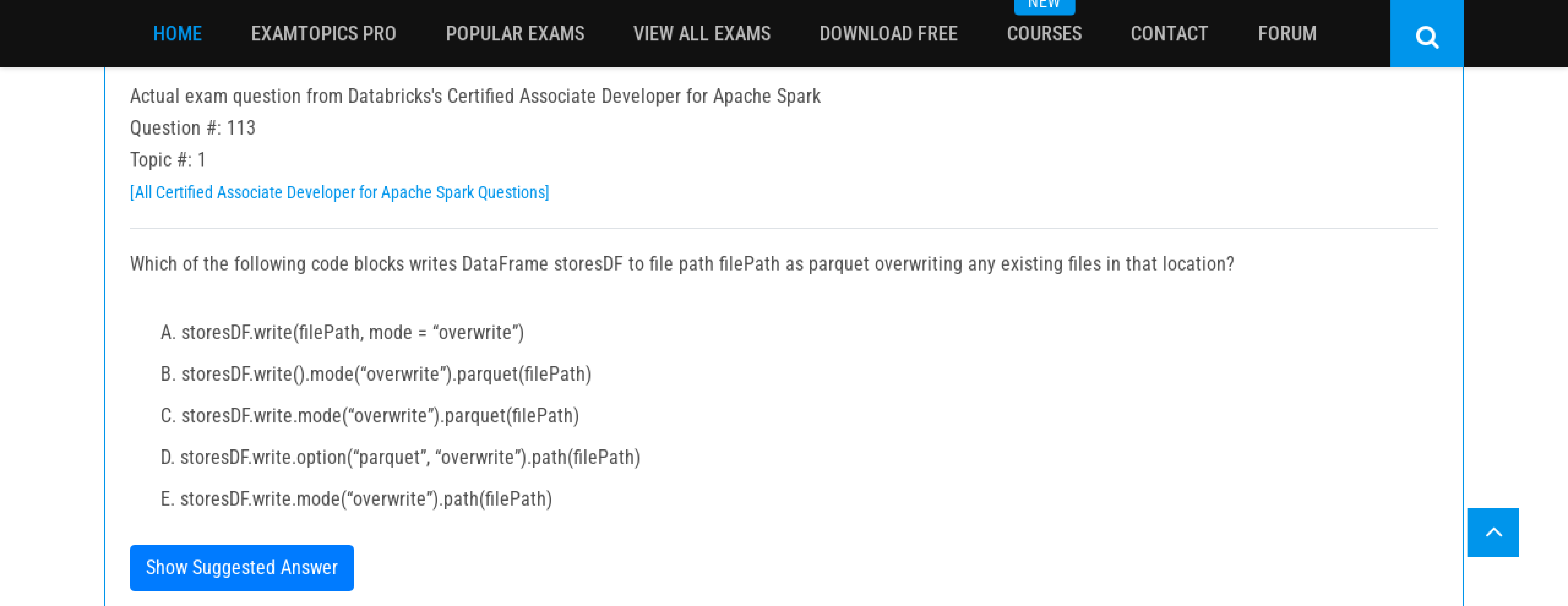
A. concat(storesDF, acquiredStoresDF)

B. storesDF.unionByName(acquiredStoresDF)

C. union(storesDF, acquiredStoresDF)

D. unionAll(storesDF, acquiredStoresDF)

E. storesDF.union(acquiredStoresDF)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 113

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks writes DataFrame storesDF to file path filePath as parquet overwriting any existing files in that location?

A. storesDF.write(filePath, mode = “overwrite”)

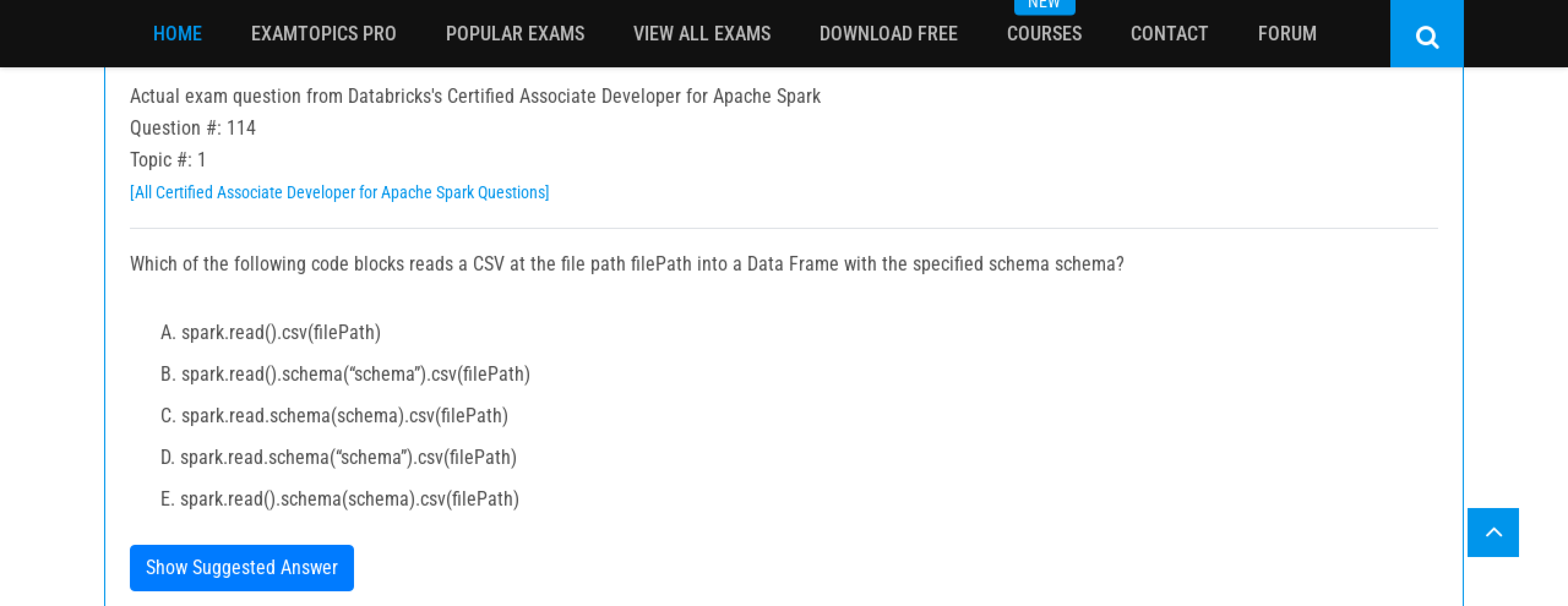
B. storesDF.write().mode(“overwrite”).parquet(filePath)

C. storesDF.write.mode(“overwrite”).parquet(filePath)

D. storesDF.write.option(“parquet”, “overwrite”).path(filePath)

E. storesDF.write.mode(“overwrite”).path(filePath)

Show Suggested Answer



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 114

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following code blocks reads a CSV at the file path filePath into a Data Frame with the specified schema schema?

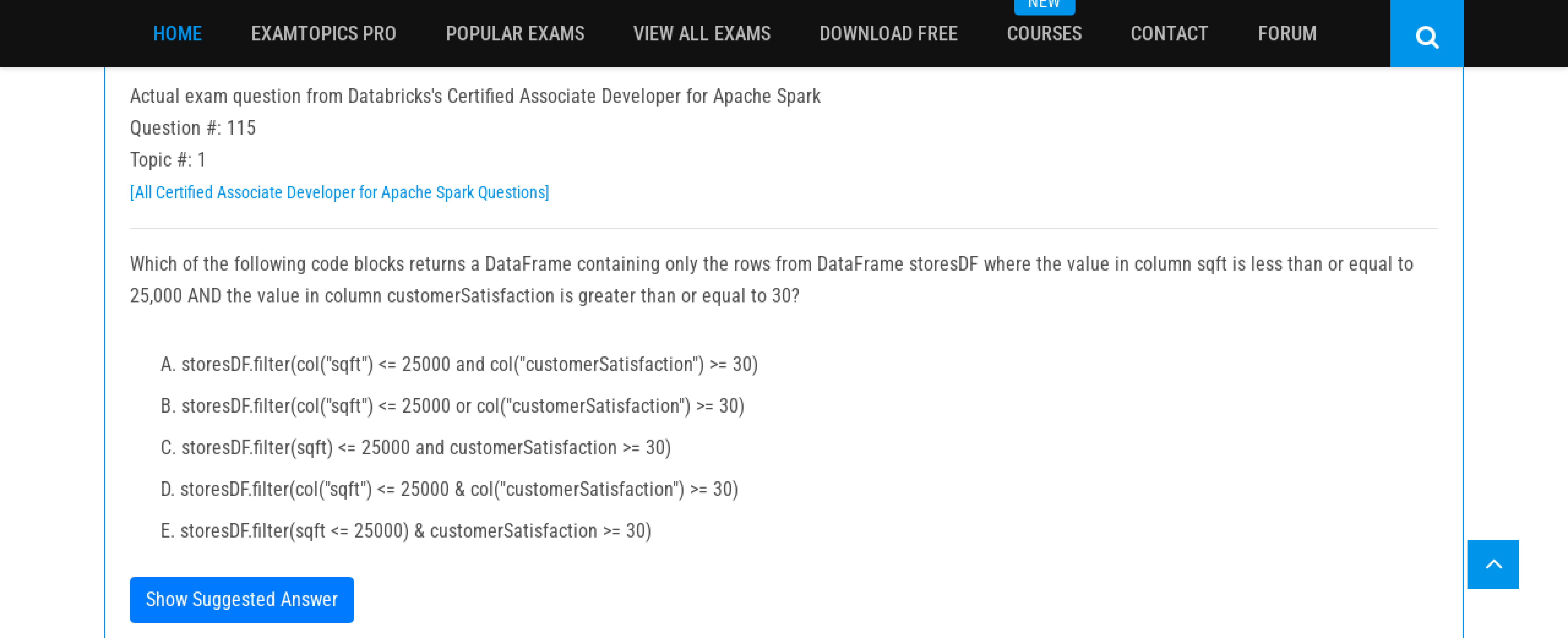
A. spark.read().csv(filePath)

B. spark.read().schema(“schema”).csv(filePath)

C. spark.read.schema(schema).csv(filePath)

D. spark.read.schema(“schema”).csv(filePath)

E. spark.read().schema(schema).csv(filePath)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 115

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

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 AND the value in column customerSatisfaction is greater than or equal to 30?

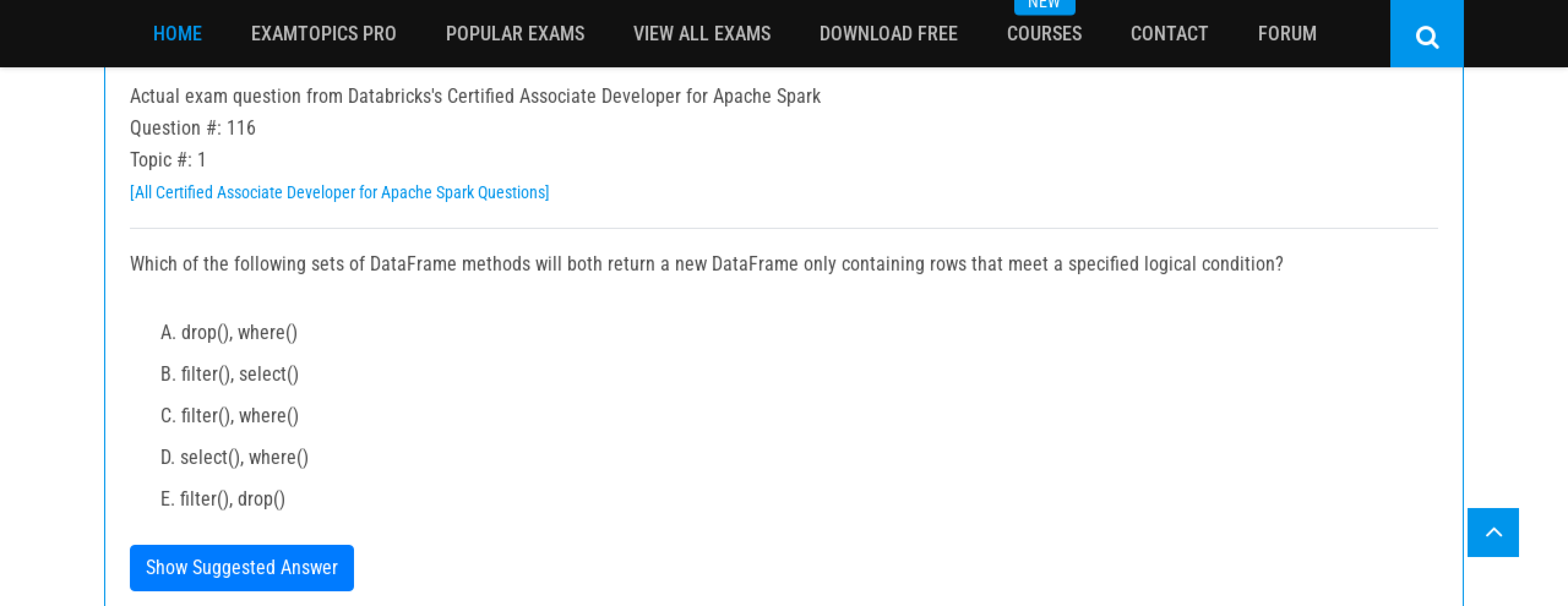
A. storesDF.filter(col("sqft") <= 25000 and col("customerSatisfaction") >= 30)

B. storesDF.filter(col("sqft") <= 25000 or col("customerSatisfaction") >= 30)

C. storesDF.filter(sqft) <= 25000 and customerSatisfaction >= 30)

D. storesDF.filter(col("sqft") <= 25000 & col("customerSatisfaction") >= 30)

E. storesDF.filter(sqft <= 25000) & customerSatisfaction >= 30)



Actual exam question from Databricks's Certified Associate Developer for Apache Spark

Question #: 116

Topic #: 1

[All Certified Associate Developer for Apache Spark Questions]

Which of the following sets of DataFrame methods will both return a new DataFrame only containing rows that meet a specified logical condition?

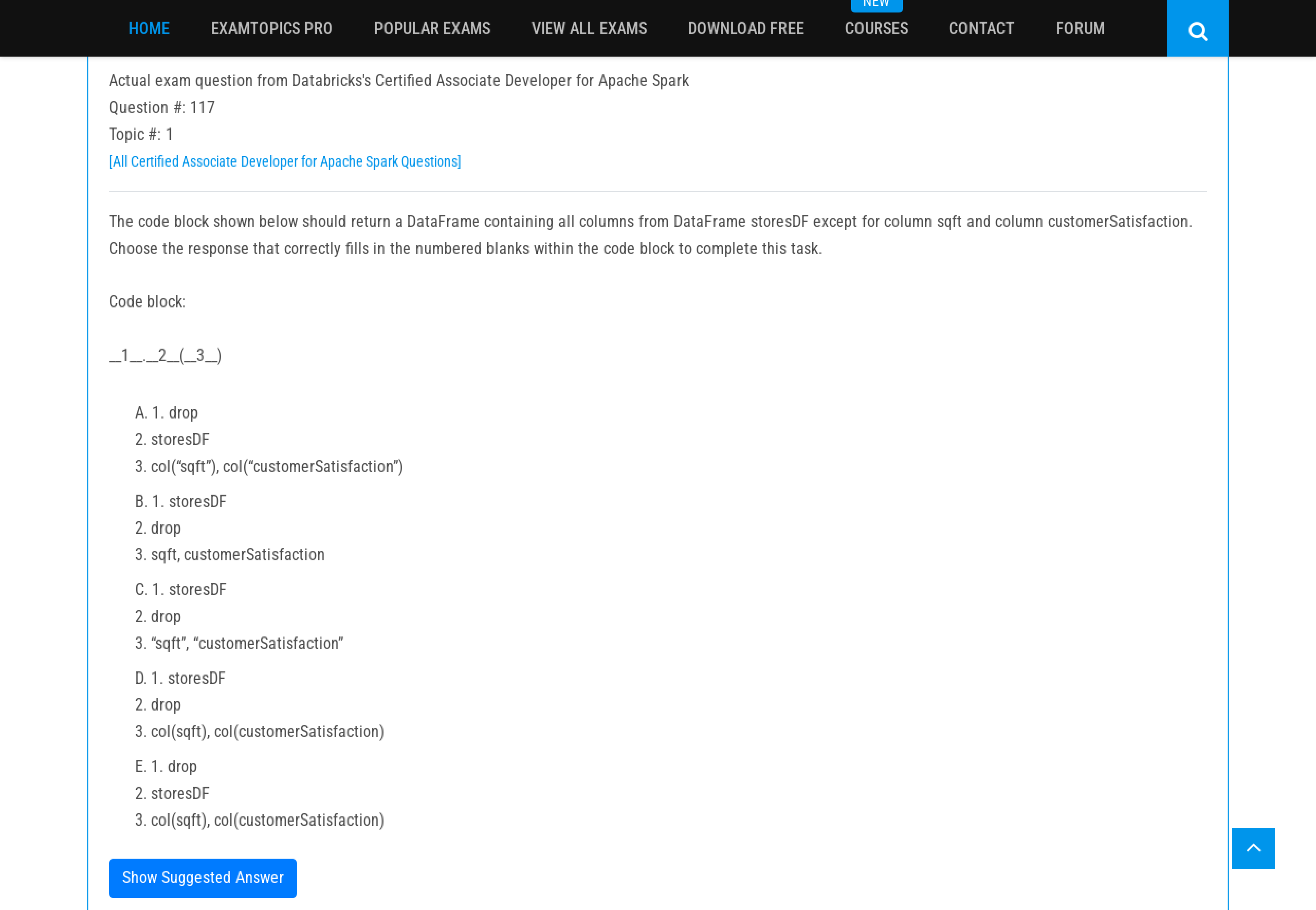
A. drop(), where()

B. filter(), select()

C. filter(), where()

D. select(), where()

E. filter(), drop()



Actual exam question from Databricks's Certified Associate Developer for Apache Spark
Question #: 117
Topic #: 1
[All Certified Associate Developer for Apache Spark Questions]

The code block shown below should return a DataFrame containing all columns from DataFrame storesDF except for column sqft and column customerSatisfaction.
Choose the response that correctly fills in the numbered blanks within the code block to complete this task.

Code block:

```
__1__.__2__(__3__)
```

A. 1. drop
2. storesDF
3. col(“sqft”), col(“customerSatisfaction”)

B. 1. storesDF
2. drop
3. sqft, customerSatisfaction

C. 1. storesDF
2. drop
3. “sqft”, “customerSatisfaction”

D. 1. storesDF
2. drop
3. col(sqft), col(customerSatisfaction)

E. 1. drop
2. storesDF
3. col(sqft), col(customerSatisfaction)