

PRACTICE QUESTIONS

Databricks Certified Associate Developer for Apache Spark

SHARED BY [u/noblekoala1984](#) on aiexamcerts

EDITION	Free Set 1
---------	------------

QUESTIONS	10
-----------	----

SHARED	October 15, 2026
--------	------------------

EXAM	Databricks Certified Associate Developer for Apache Spark
------	---

Shared by [u/noblekoala1984](#) on aiexamcerts

10 practice questions for the Databricks Certified Associate Developer for Apache Spark exam, each followed by its answer and a short explanation. More free sets and the full practice exam:

www.aiexamcerts.com/databricks/databricks-certified-associate-developer-for-apache-spark. Spotted a mistake? Email support@aiexamcerts.com with the exam name and the question number.

1. Which statement about transformations and actions in Spark is correct?

- A Actions such as count only extend the plan, while transformations such as select and filter trigger execution
- B Transformations such as filter and select are lazy and only extend the plan, while actions such as count and collect trigger execution
- C Both transformations and actions execute immediately when they are called
- D Only collect triggers execution, and every other call is lazy and never runs

Answer: B

Spark records transformations lazily in a logical plan and runs the job only when an action such as count, collect or write needs results. Reversing the roles is the tempting distractor, because a count looks like a computation, but it is the action that starts the job. Immediate execution for every call contradicts Spark's lazy model, and collect is only one of several actions.

2. The Spark entry point that creates DataFrames and runs SQL queries in PySpark is the _____.

Answer: SparkSession (also accepted: Spark session)

SparkSession is the unified entry point for DataFrame and SQL work, and it gives access to the underlying SparkContext. SparkContext is the tempting distractor, since it was the older entry point for RDD work, but it is not the primary DataFrame entry point.

3. A join must combine a large orders DataFrame with a 5 MB lookup DataFrame, and the lookup should be sent to every executor instead of shuffling the large side. Which hint does this?

- A repartition(lookup_df)
- B cache(lookup_df)
- C coalesce(lookup_df, 1)
- D broadcast(lookup_df)

Answer: D

Wrapping the small DataFrame in broadcast () asks Spark to copy it to every executor, so the large side is not shuffled for the join. repartition is the tempting distractor, since it reshapes data before a join, but it shuffles the data rather than avoiding a shuffle. cache keeps data in memory and coalesce reduces partition count, so neither is a join hint.

4. Parquet files store their schema in the file footer, so Spark can read the schema without scanning every row of the dataset.
- A True
 - B False

Answer: A (True)

Parquet keeps the schema in its footer metadata, so Spark reads it without a full scan. The statement is true. The tempting confusion is with CSV, where schema inference may need to read the data.

5. Which two statements about DataFrame joins are correct? (Choose two.)
- A An inner join keeps unmatched rows from both sides with nulls filled in
 - B A broadcast hint on the smaller side can avoid shuffling the larger side
 - C A cross join is the recommended default when join keys do not match
 - D Join keys must have the same name in both DataFrames
 - E A left semi join returns only left-side rows that have a match on the right

Answer: B, E

A left semi join keeps matching left rows with only left columns, and a broadcast hint on the smaller side avoids shuffling the large side. An inner join is the tempting distractor, since it looks like a general-purpose join, but keeping unmatched rows with nulls is what outer joins do. A cross join multiplies rows and is not a fallback for mismatched keys, and keys do not need the same name when an explicit condition is used.

6. A pipeline reads a very large CSV on every run and must avoid an extra pass over the data just to guess column types. Which approach fits best?
- A Use inferSchema=True on every run for safety
 - B Read the file as text and cast every column to string afterwards
 - C Define a StructType schema and pass it to the reader
 - D Load the file into pandas first and convert it to a Spark DataFrame

Answer: C

A predefined StructType avoids the extra pass that inference needs, and it keeps column types stable across runs. inferSchema is the tempting distractor, since it is convenient, but it scans the data to guess types and can guess differently when the data changes. Reading everything as text and loading through pandas add work and lose control of types.

7. Which write mode makes the job fail if the target path already contains data?

- A overwrite, which replaces the existing output at the path
- B append
- C ignore
- D errorifexists

Answer: D

errorifexists, which is also the default save mode, stops the write when data already exists at the target. ignore is the tempting distractor, since it also refuses to replace existing data, but it silently skips the write instead of failing. overwrite replaces the data, and append adds to it.

8. The DataFrame method that renames one column and returns a new DataFrame is _____.

Answer: withColumnRenamed

withColumnRenamed takes the existing name and the new name and returns a new DataFrame. withColumn is the tempting distractor, since it is the usual column method, but it adds or replaces a column using an expression.

9. Which aggregate counts the distinct customer_id values in a DataFrame aggregation?

- A F.countDistinct('customer_id')
- B F.count('customer_id', distinct=True)
- C F.distinct('customer_id').count
- D F.count_unique('customer_id')

Answer: A

F.countDistinct is the column aggregate that returns the number of distinct values. F.distinct('customer_id').count is the tempting distractor, since it looks like it would work, but distinct is a DataFrame method rather than a column function. The other two forms do not exist in PySpark.

10. Which aggregate collects the values of a group into an array and keeps duplicate values?

- A F.collect_set('item')
- B F.collect_list('item')
- C F.array('item')
- D F.concat_ws(',', 'item')

Answer: B

collect_list gathers every value of the group into an array and keeps duplicates. collect_set is the tempting distractor, since it also collects values, but it removes duplicates. array builds an array from columns within a row, and concat_ws joins values into one string.