

PRACTICE QUESTIONS

Databricks Certified Associate Developer for Apache Spark

SHARED BY [u/cleverkoala2001](#) on aiexamcerts

EDITION	Free Set 2
---------	------------

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

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

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

Shared by [u/cleverkoala2001](#) 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. A transformation must redistribute rows across the cluster so that all rows with the same key land together. Which term describes this data movement?
 - A A broadcast
 - B A checkpoint
 - C A shuffle
 - D A cache

Answer: C

A shuffle moves rows between executors so that matching keys end up in the same partition, which happens in wide transformations such as `groupBy` and sort-merge joins. A broadcast is the tempting distractor, since it also moves data across the cluster, but it copies a small dataset to every executor without redistributing the large side. Checkpoints write lineage to reliable storage, and caching keeps partitions in memory.

2. Setting `spark.sql.adaptive.enabled` to false is required before the planner can choose a broadcast join in Spark 3.x.
 - A True
 - B False

Answer: B (False)

Broadcast joins are chosen from size estimates and thresholds such as `autoBroadcastJoinThreshold`, and they do not require Adaptive Query Execution to be disabled. The statement is false. The tempting belief is that adaptive execution competes with broadcasting, when it can even convert joins to broadcasts at runtime.

3. A Python UDF is slowing a pipeline. Which change usually helps most?
 - A Replace the UDF with built-in functions from `pyspark.sql.functions` wherever they cover the logic
 - B Wrap the UDF in a second UDF so that it runs twice as fast
 - C Change the UDF's return type to `StringType` for every column
 - D Move the UDF call into a driver loop that calls `collect` for each row

Answer: A

Built-in functions run inside the JVM and can be optimized by Catalyst, while Python UDFs move rows between the JVM and Python worker processes. Wrapping a UDF in another UDF is the tempting distractor, since it looks like an extra optimization, but it adds the same serialization overhead again. Changing return types does not remove that overhead, and a driver loop with `collect` is slower still.

4. The Spark property that sets the maximum number of bytes packed into one input partition when files are read is `spark.sql.files._____`.

Answer: maxPartitionBytes

`maxPartitionBytes` controls how input files are packed into partitions during a read, which determines how many scan tasks are created. `shuffle.partitions` is the tempting distractor, since it also sets partition counts, but it applies after a shuffle rather than during a read.

5. Which two column expressions handle null values correctly? (Choose two.)

- A `F.coalesce(F.col('nick'), F.col('name'))` returns the first non-null value of the two columns
- B Comparing a column to `None` with `==` returns true for rows where the column is null
- C `df.fillna({'score': 0})` replaces null scores with zero
- D `F.coalesce` returns null whenever any of its arguments is null
- E `F.col('x') == F.lit(None)` is the standard way to test for null values

Answer: A, C

`coalesce` returns the first non-null argument, and `fillna` with a column-to-value mapping replaces nulls in the named column. The equality test is the tempting distractor, since it looks like a direct null check, but a comparison with null does not evaluate to true, so `isNull` is the correct test. `coalesce` returns the first non-null value rather than null, and comparing with a literal null is not the standard test either.

6. Output must be stored so that each country lands in its own folder under the output path. Which writer call does this?

- A `.write.partitionBy('country')`
- B `.write.bucketBy(10, 'country')`
- C `.write.sortBy('country')`
- D `.write.format('country')`

Answer: A

`partitionBy` creates a directory for each distinct value of the column under the output path. `bucketBy` is the tempting distractor, since it also organizes data by a column, but it hashes values into a fixed number of buckets and needs `saveAsTable`. `sortBy` orders rows within buckets, and `format` selects a file format rather than a column.

7. Which function parses a string column such as '2026-03-14' into a date column?

- A F.date_format
- B F.to_date
- C F.current_date
- D F.datediff

Answer: B

to_date parses a string into a DateType value, and it accepts a format pattern when the default does not match. date_format is the tempting distractor, since it also works with dates, but it converts a date or timestamp into a formatted string. current_date returns today's date, and datediff computes the gap between two dates.

8. A window function such as lag returns predictable results only when the window specification includes an ordering.

- A True
- B False

Answer: A (True)

Without an orderBy in the window, lag and lead have no defined row order, so their results can vary between runs. The statement is true. Partitioning alone is the tempting incomplete setup, since it groups rows without ordering them.

9. Which SQL statement returns total sales per month from a sales table with a DATE column named sale_date?

- A SELECT sale_date, SUM(amount) FROM sales
- B SELECT SUM(amount) FROM sales WHERE month = 'monthly'
- C SELECT month(sale_date), SUM(amount) FROM sales GROUP BY month(sale_date)
- D SELECT month(sale_date) FROM sales ORDER BY SUM(amount) WITHOUT GROUP BY

Answer: C

Grouping by the month expression produces one row per month with the sum of amounts. Selecting the raw date next to an aggregate without GROUP BY is the tempting distractor, since it looks close, but it mixes row-level and aggregated values in an invalid way. The other two statements use a month column that does not exist or aggregate without grouping.

10. A temporary view must be visible to every session within the same Spark application. Which kind of view provides this?
- A A temporary view, which is shared by all sessions in the application
 - B A permanent table stored in the metastore for each session
 - C A view created in the driver's local filesystem
 - D A global temporary view, which is stored in the global_temp database

Answer: D

A global temporary view lives in the global_temp database and is shared across sessions of the same application. A session temporary view is the tempting distractor, since it is also a temporary view, but it is visible only to the session that created it. A permanent table is stored beyond the application, and a driver filesystem view is not a Spark view.