Graphical user interface, text, application

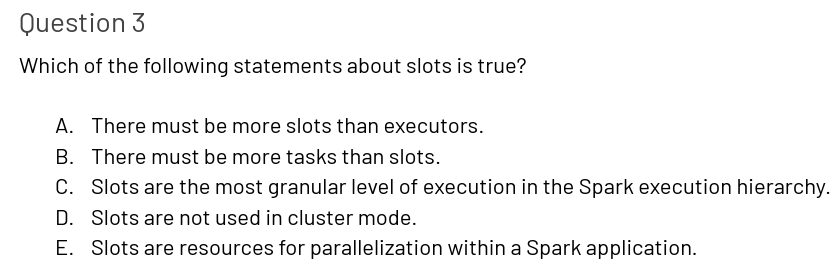
Description automatically generated

B. The spak driver is horizontally scaled to increase overall processing throughput. – Driver is just one node, vertically scaled.

Text

Description automatically generated

E. Worker nodes are machines that host the executors responsible for the execution of tasks.



E. Slots are resources for parallelization within a Spark application.

Text

Description automatically generated with medium confidence

D. Task.

Graphical user interface, text, application, email

Description automatically generated

E. Stage

Text

Description automatically generated

A. A shuffle is the process by which data is compared across partitions

Graphical user interface, text, application, email

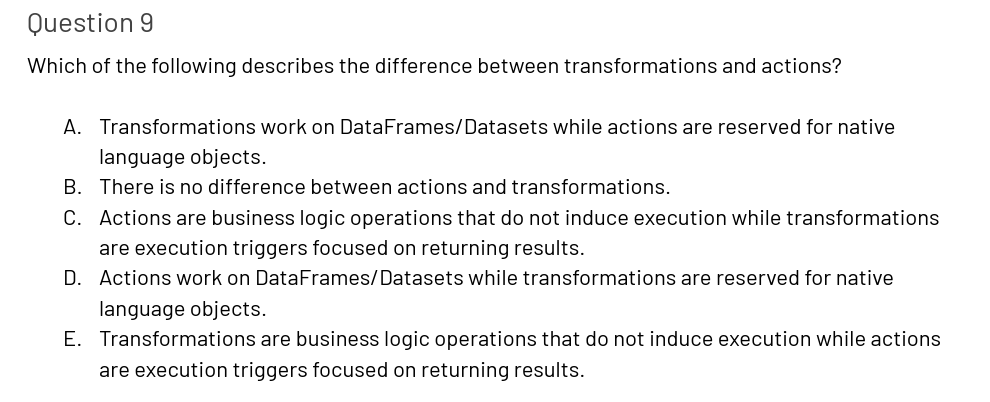
Description automatically generated

1. Performance will be suboptimal because not all executors will be utilized at the same time.

Text

Description automatically generated with low confidence

E. Dataframe.count() because is an action, not a transformer.



E. Transformations are business logic operations that do not induce execution while actions are execution triggers focused on returning results.

Graphical user interface, text, application, email

Description automatically generated

In Spark: a wide transformation is a transformation that requires data to be shuffled between executors, while a narrow transformation is a transformation that does not require data to be shuffled - [What is Wide and Narrow Transformation in Apache Spark - Nixon Data](https://nixondata.com/knowledge/apache-spark-fundamentals/apache-spark-wide-and-narrow-transformation/#:~:text=In%20Apache%20Spark%2C%20a%20wide%20transformation%20is%20a,that%20does%20not%20require%20data%20to%20be%20shuffled.).

D. Dataframe.select()

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

A. Spark’s execution/deployment mode determines where the driver and executors are physically located when a Spark application is run.

Graphical user interface, application, Word

Description automatically generated

B. They should all ensure completion because worker nodes are fault-tolerant.

Graphical user interface, text, application, email

Description automatically generated

Text, letter

Description automatically generated

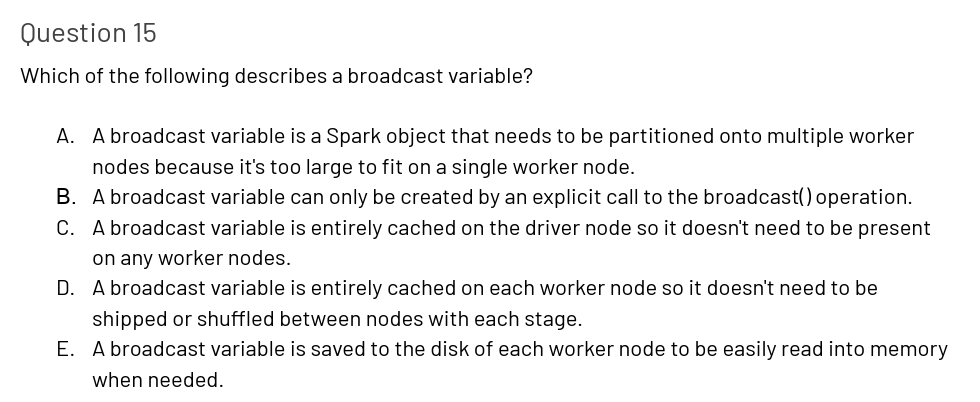
A. An out-of-memory error occurs when either the driver or an executor does not have enough memory to collect or process the data allocated to it.

Graphical user interface, text

Description automatically generated

A. MEMORY\_AND\_DISK / [pyspark.sql.DataFrame.persist — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.DataFrame.persist.html#pyspark.sql.DataFrame.persist)

MEMORY is for streaming. [Spark Streaming - Spark 3.3.1 Documentation (apache.org)](https://spark.apache.org/docs/latest/streaming-programming-guide.html#caching--persistence).



D. A broadcast variable is saved to the disk of each Worker node to be easily read into memory when needed.

[What are broadcast variables in Spark and when to use them? – Hadoop In Real World](https://www.hadoopinrealworld.com/what-are-broadcast-variables-in-spark-and-when-to-use-them/)

Graphical user interface, text, application

Description automatically generated

A picture containing text

Description automatically generated

D. Dataframe.coalesce(n)

Graphical user interface, text, application, email

Description automatically generated

C. RDDs.

Text

Description automatically generated

C. storesDF.select(“storeId”, “division”)

Table

Description automatically generated

Text

Description automatically generated

1. storesDF.drop(“sqft”, “customerSatisfaction”)

Text, letter

Description automatically generated

B. storesDF.filter( col(“sqft”) <= 25000

Graphical user interface, text

Description automatically generated with medium confidence

Text, letter

Description automatically generated

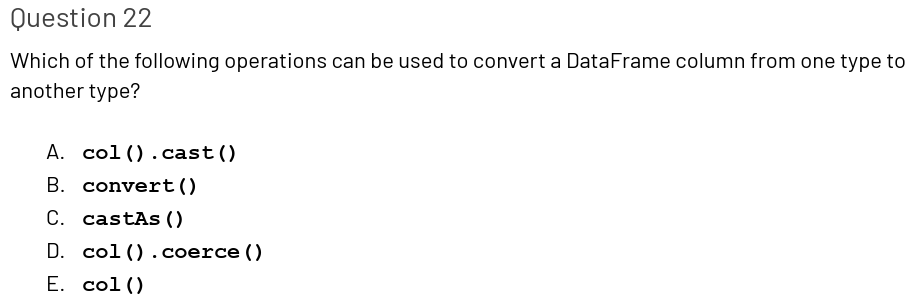
Text

Description automatically generated

Table

Description automatically generated with low confidence

A. storedDF.filter( (col(“sqft”) <= 25000 ) | ( col(“customerSatisfaction”) >= 30 ) )



A. col().cast()

Text, letter

Description automatically generated

D. storesDF.withColumn(“sqft100”, col(“sqft) / 100)

Text

Description automatically generated

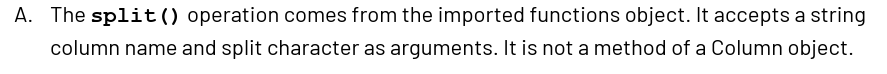
C. storesDF.withColumn( “numberOfManagers”, lit(1) )

Table

Description automatically generated

Text, letter

Description automatically generated



Text

Description automatically generated

B. 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.

Graphical user interface, text, application, email

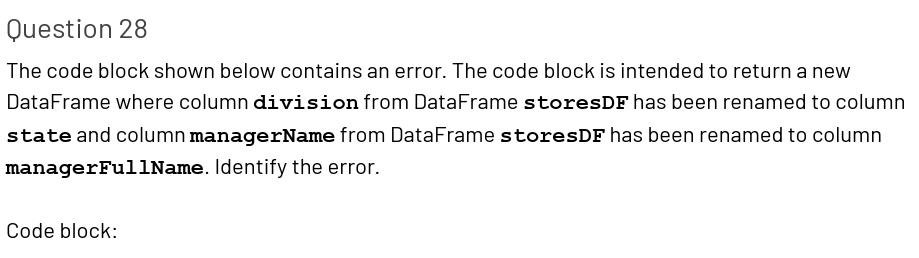
Description automatically generated

C. explode()

Text

Description automatically generated

A. storesDF.withColumn(“storeCategory”, lower(col(“storeCategory”)))



Text

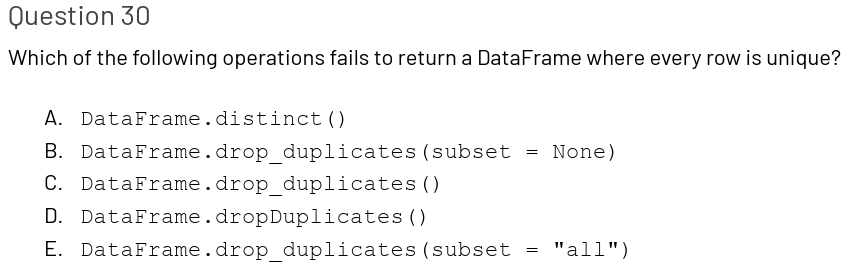
Description automatically generated

D. The first argument to operation withColumnRenamed() should be the old column name and the second argument should be the new column name

Graphical user interface, text, application

Description automatically generated

E. storesDF.na.drop(“all”) [pyspark.sql.DataFrame.dropna — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.DataFrame.dropna.html#pyspark.sql.DataFrame.dropna)



E. DataFrame.drop\_duplicates(subset = “all”)

A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

A. storesDF.agg(approx\_count\_distinct(col(“division”)).alias(“divisionDistinct”)) [pyspark.sql.functions.approx\_count\_distinct — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.functions.approx_count_distinct.html#pyspark.sql.functions.approx_count_distinct)

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

A. storesDF.agg(mean(col(“sqft”)).alias(“sqftMean”))

Graphical user interface, text, application

Description automatically generated

D. storesDF.count()

Text

Description automatically generated

E. storesDF.groupBy(“division”).agg(sum(col(“sqft”)))

Text

Description automatically generated

B. storesDF.describe(“sqft”)

[pyspark.sql.DataFrame.describe — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.DataFrame.describe.html#pyspark.sql.DataFrame.describe)

Text

Description automatically generated with low confidence

A. sort() and orderBy()

Graphical user interface, text, application

Description automatically generated



Graphical user interface, text

Description automatically generated

E. The first argument True sets the sampling to be with replacement

[pyspark.sql.DataFrame.sample — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.DataFrame.sample.html#pyspark.sql.DataFrame.sample) [pyspark.sql.DataFrame.sampleBy — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.DataFrame.sampleBy.html#pyspark.sql.DataFrame.sampleBy)

Graphical user interface, text, application, email

Description automatically generated

B. Dataframe.take(n) [pyspark.sql.DataFrame.take — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.DataFrame.take.html#pyspark.sql.DataFrame.take)

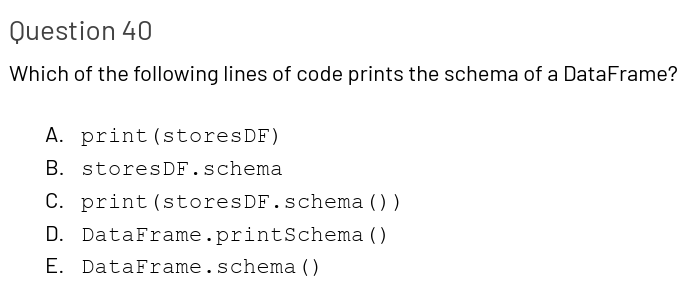
Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

D. storeDF.first().



D. DataFrame.printSchema()

Text, letter

Description automatically generated

B. 1, 4

[Spark SQL UDF (User Defined Functions) - Spark By {Examples} (sparkbyexamples.com)](https://sparkbyexamples.com/spark/spark-sql-udf/)

Question 42

Text

Description automatically generated

A picture containing table

Description automatically generated

1. 3, 4

Graphical user interface, text, application, email

Description automatically generated

C. spark.sql()

Graphical user interface, text, application

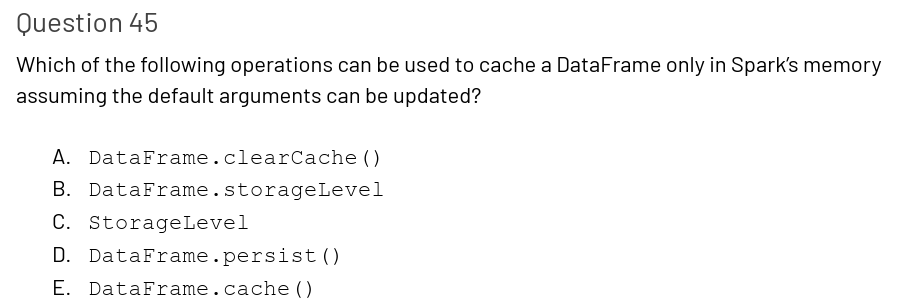
Description automatically generated

Text

Description automatically generated with medium confidence

B. spark.createDataFrame(years, IntegerType())





D. DataFrame.persist(…) [DataFrame — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/dataframe.html)

Graphical user interface, text, application, email

Description automatically generated

D. The repartition operation induced a full shuffle. The coalesce operation should be used instead.

Text

Description automatically generated with low confidence

C. storesDF.repartition(12) -> shuffle tasks to secure you always get N-partitions

Graphical user interface, text, application, email

Description automatically generated

A. spark.sql.shuffle.partitions -> [performance - What is the difference between spark.sql.shuffle.partitions and spark.default.parallelism? - Stack Overflow](https://stackoverflow.com/questions/45704156/what-is-the-difference-between-spark-sql-shuffle-partitions-and-spark-default-pa)

Text, letter

Description automatically generated

Text, letter

Description automatically generated

B. 2,1

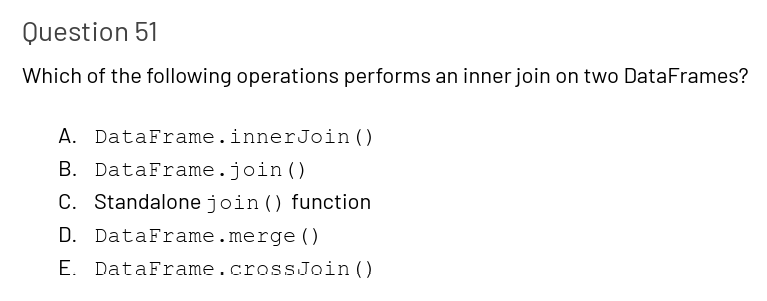
Graphical user interface, text, letter

Description automatically generated

Text

Description automatically generated

1. storesDF.withColumn(“openDateFormat”, col(“openDate”).cast(“Date”)).withColumn(“month”, month(col(“openDateFormat”)))



B. DataFrame.join()

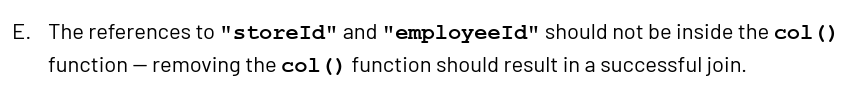
Text

Description automatically generated

A. storesDF.join(employeesDF, “storeId”,”outer”)

Graphical user interface, text, application, letter

Description automatically generated



E. The references to “storeId” and “employeeId” should not be inside the col() function – removing the col() function should result in a successful join.

- [pyspark.sql.DataFrame.join — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.DataFrame.join.html#pyspark.sql.DataFrame.join)

Graphical user interface, text, application

Description automatically generated

A. Spark.sql.autoBroadcastJoinThreshold

[Does spark.sql.autoBroadcastJoinThreshold work for joins using Dataset's join operator? - Stack Overflow](https://stackoverflow.com/questions/43984068/does-spark-sql-autobroadcastjointhreshold-work-for-joins-using-datasets-join-op)

Text, letter

Description automatically generated

Text

Description automatically generated

E. storesDF crossJoin employeesDF

Graphical user interface, text

Description automatically generated with medium confidence

E. DataFrame.union()- [pyspark.sql.DataFrame.unionByName — PySpark 3.3.1 documentation (apache.org)](https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.DataFrame.unionByName.html#pyspark.sql.DataFrame.unionByName)

Graphical user interface, text

Description automatically generated

E. storesDF.write.parquet(filePath)

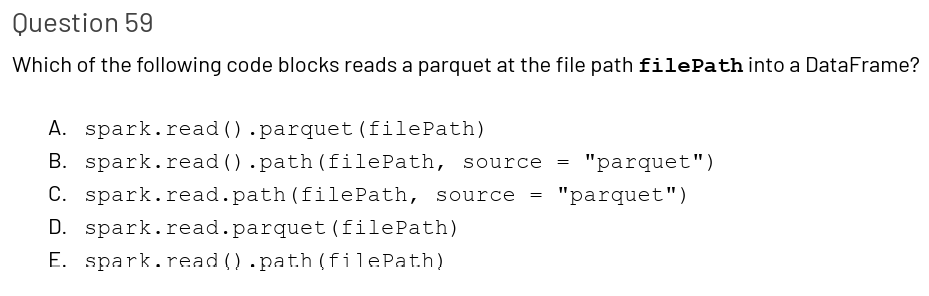
Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated

C. There is no repartition() operation for DataFrameWriter – the partitionBy() operation should be used instead.



D. spark.read.parquet(filePath)

Text

Description automatically generated

E. spark.read.schema(schema).format(“json”).load(filePath)