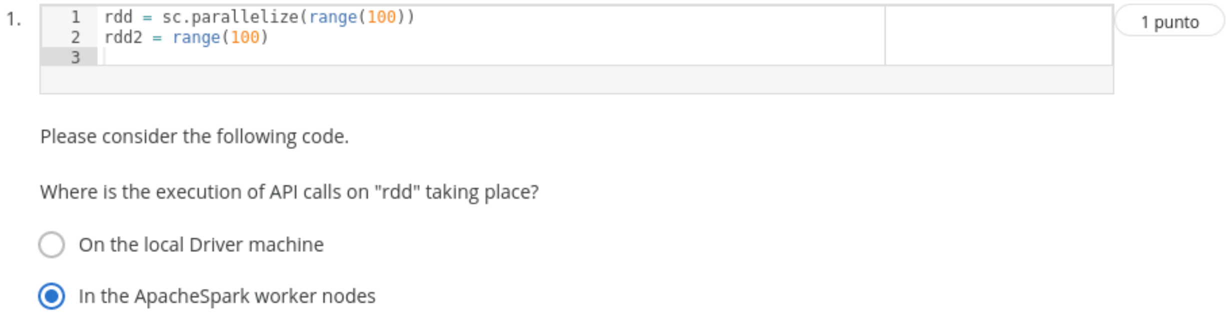
Apache Spark and parallel data processing

Pregunta 1:



1.

```
rdd = sc.parallelize(range(100))
rdd2 = range(100)

```

1 punto

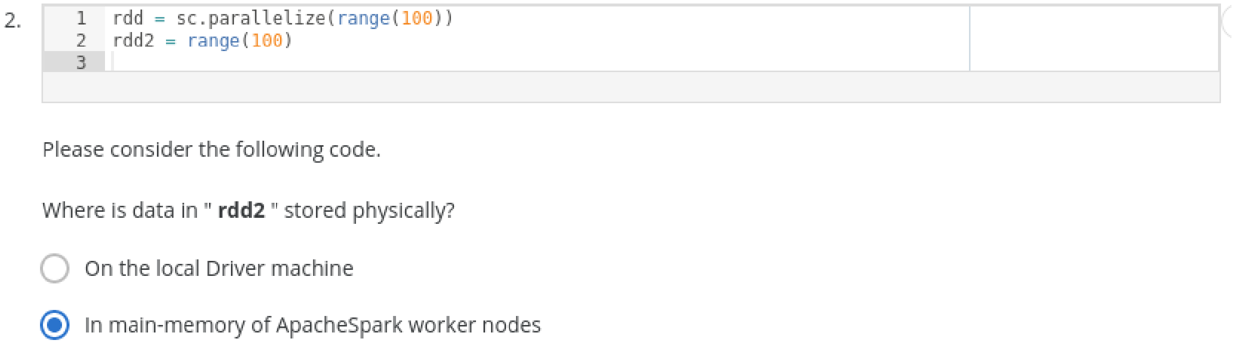
Please consider the following code.

Where is the execution of API calls on "rdd" taking place?

○ On the local Driver machine

● In the ApacheSpark worker nodes

Pregunta 2:



2.

```
rdd = sc.parallelize(range(100))
rdd2 = range(100)

```

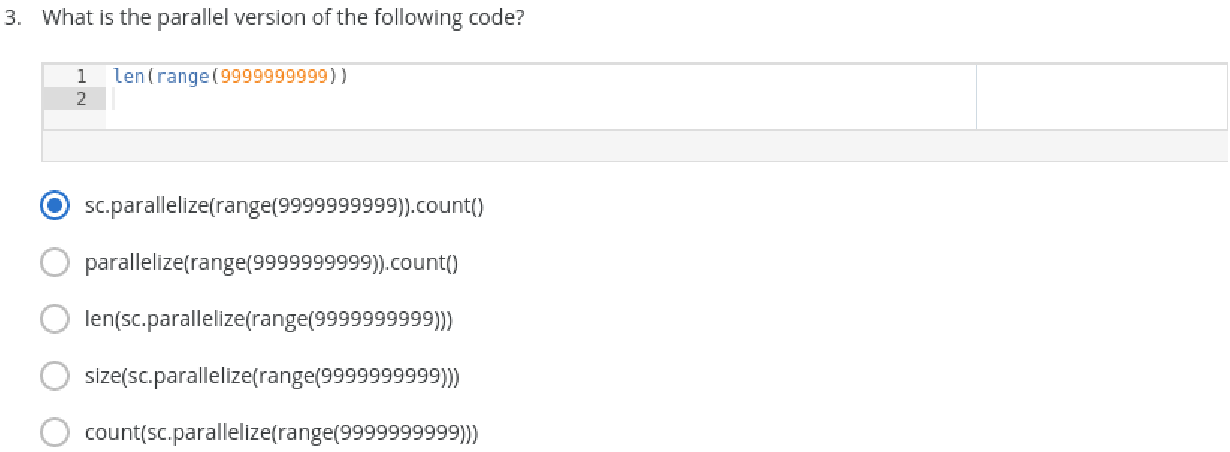
Please consider the following code.

Where is data in " **rdd2** " stored physically?

○ On the local Driver machine

● In main-memory of ApacheSpark worker nodes

Pregunta 3:



3. What is the parallel version of the following code?

```
len(range(9999999999))

```

● sc.parallelize(range(9999999999)).count()

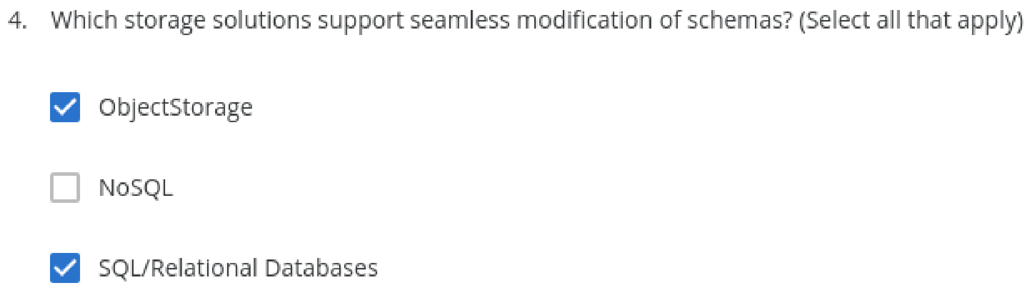
○ parallelize(range(9999999999)).count()

○ len(sc.parallelize(range(9999999999)))

○ size(sc.parallelize(range(9999999999)))

○ count(sc.parallelize(range(9999999999)))

Pregunta 4:



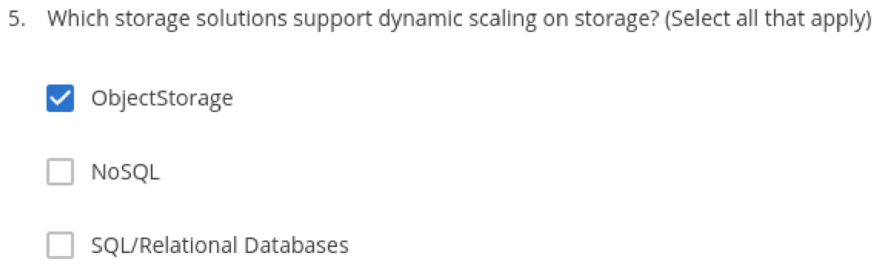
4. Which storage solutions support seamless modification of schemas? (Select all that apply)

☑ ObjectStorage

□ NoSQL

☑ SQL/Relational Databases

Pregunta 5:



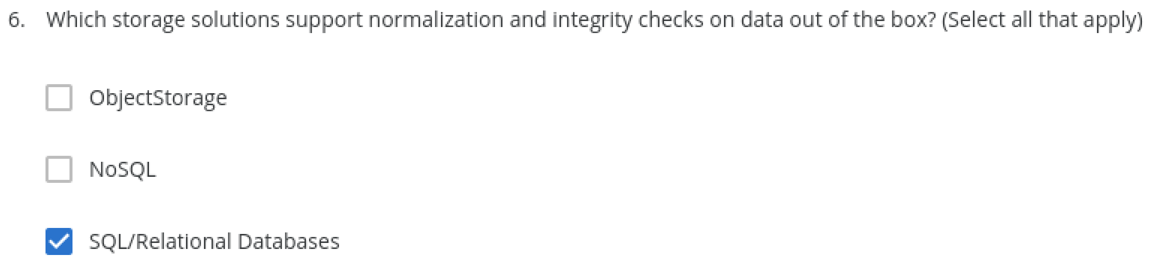
5. Which storage solutions support dynamic scaling on storage? (Select all that apply)

☑ ObjectStorage

□ NoSQL

□ SQL/Relational Databases

Pregunta 6:



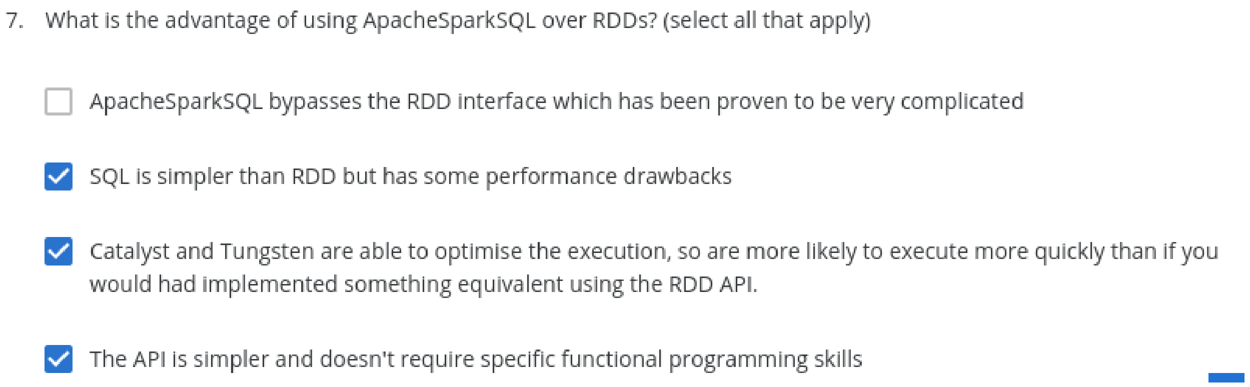
6. Which storage solutions support normalization and integrity checks on data out of the box? (Select all that apply)

□ ObjectStorage

□ NoSQL

☑ SQL/Relational Databases

Pregunta 7:



7. What is the advantage of using ApacheSparkSQL over RDDs? (select all that apply)

□ ApacheSparkSQL bypasses the RDD interface which has been proven to be very complicated

☑ SQL is simpler than RDD but has some performance drawbacks

☑ Catalyst and Tungsten are able to optimise the execution, so are more likely to execute more quickly than if you would had implemented something equivalent using the RDD API.

☑ The API is simpler and doesn't require specific functional programming skills