1. RDD is Resilient Distributed Dataset.
2. It’s the collection of objects like list in python. It is immutable and in-memory processing.
3. By using parallelize() function of SparkContext we can create an RDD.
4. Resilience: RDDs are fault-tolerant. If a partition of an RDD is lost, Spark can rebuild it using the transformations that created the RDD.
5. Distributed: RDDs are distributed across the nodes of a cluster, enabling parallel processing.
6. Immutability: RDDs are immutable, meaning their contents cannot be modified once created. You can only create new RDDs through transformations.
7. Laziness: Transformations on RDDs are evaluated lazily. This means that Spark doesn't compute the result of a transformation until an action is called, optimizing execution plans.

from pyspark.sql import \*  
from pyspark.sql.functions import \*  
from pyspark.sql.types import \*  
  
*# Create a Spark session*spark = SparkSession.builder.appName("rdd").getOrCreate()  
  
myData = [(1,'Harsha'),(2,'Mokshit')]  
  
rdd1 = spark.sparkContext.parallelize(myData)  
  
print(rdd1.collect())  
  
df = rdd1.toDF(schema=['id','name'])  
df.show()  
  
df1 = spark.createDataFrame(rdd1,schema=['id','name'])  
df1.show()

**Output:**

[(1, 'Harsha'), (2, 'Mokshit')]

+---+-------+

| id| name|

+---+-------+

| 1| Harsha|

| 2|Mokshit|

+---+-------+

+---+-------+

| id| name|

+---+-------+

| 1| Harsha|

| 2|Mokshit|

+---+-------+