Problem Statement

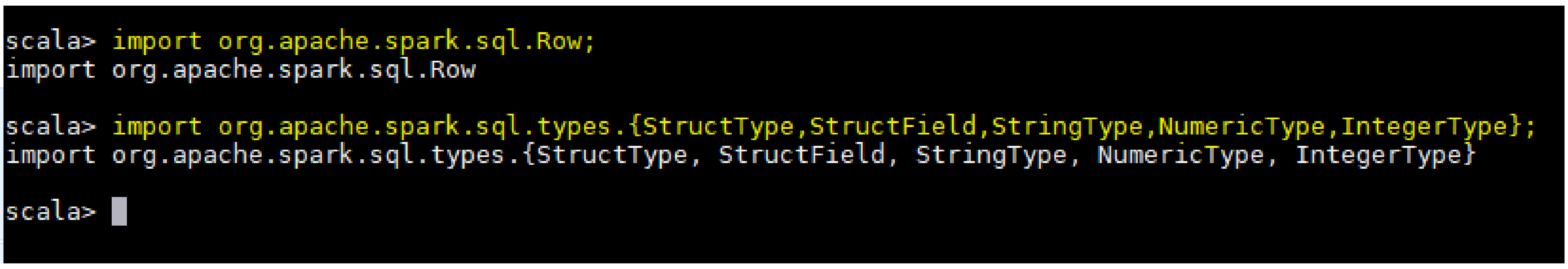
Using spark-sql, Find:

1. **What are the total number of gold medal winners every year**

Below is the command used to find the result-

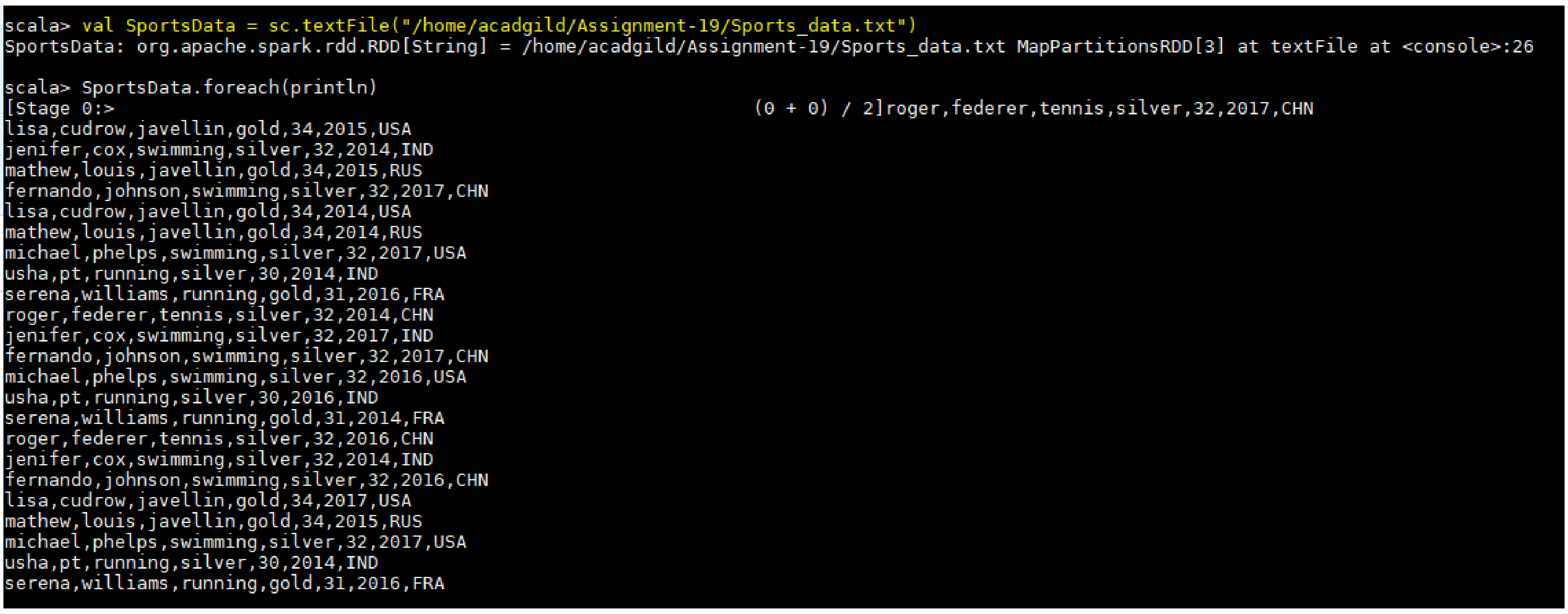
- val SportsData = sc.textFile("/home/acadgild/Assignment-19/Sports_data.txt")
- val schemaString = "firstname:string,lastname:string,sports:string,medal:string,age:integer,year:integer,country:string"
- val schema = StructType(schemaString.split(",").map(x => StructField(x.split(":")(0), if (x.split(":")(1).equals("string")) StringType else IntegerType, true)))
- val rowRDD = SportsData.map(_.split(",")).map(r => Row(r(0), r(1), r(2), r(3), r(4).toInt, r(5).toInt, r(6)))
- val SportsDataDF = spark.createDataFrame(rowRDD, schema)
- SportsDataDF.createOrReplaceTempView("Sports_Data")
- val result1DF = spark.sql("SELECT year,COUNT(*) FROM Sports_Data WHERE medal = 'gold' GROUP BY year")
- result1DF.show()

In order to proceed we need to import some dependencies as shown below-



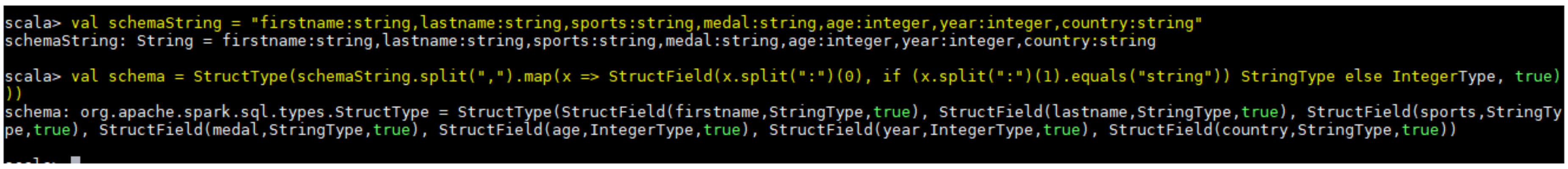
```
scala> import org.apache.spark.sql.Row;
import org.apache.spark.sql.Row

scala> import org.apache.spark.sql.types.{StructType,StructField,StringType,NumericType,IntegerType};
import org.apache.spark.sql.types.{StructType, StructField, StringType, NumericType, IntegerType}

scala>
```

Now we are creating a RDD which reads from the input file-



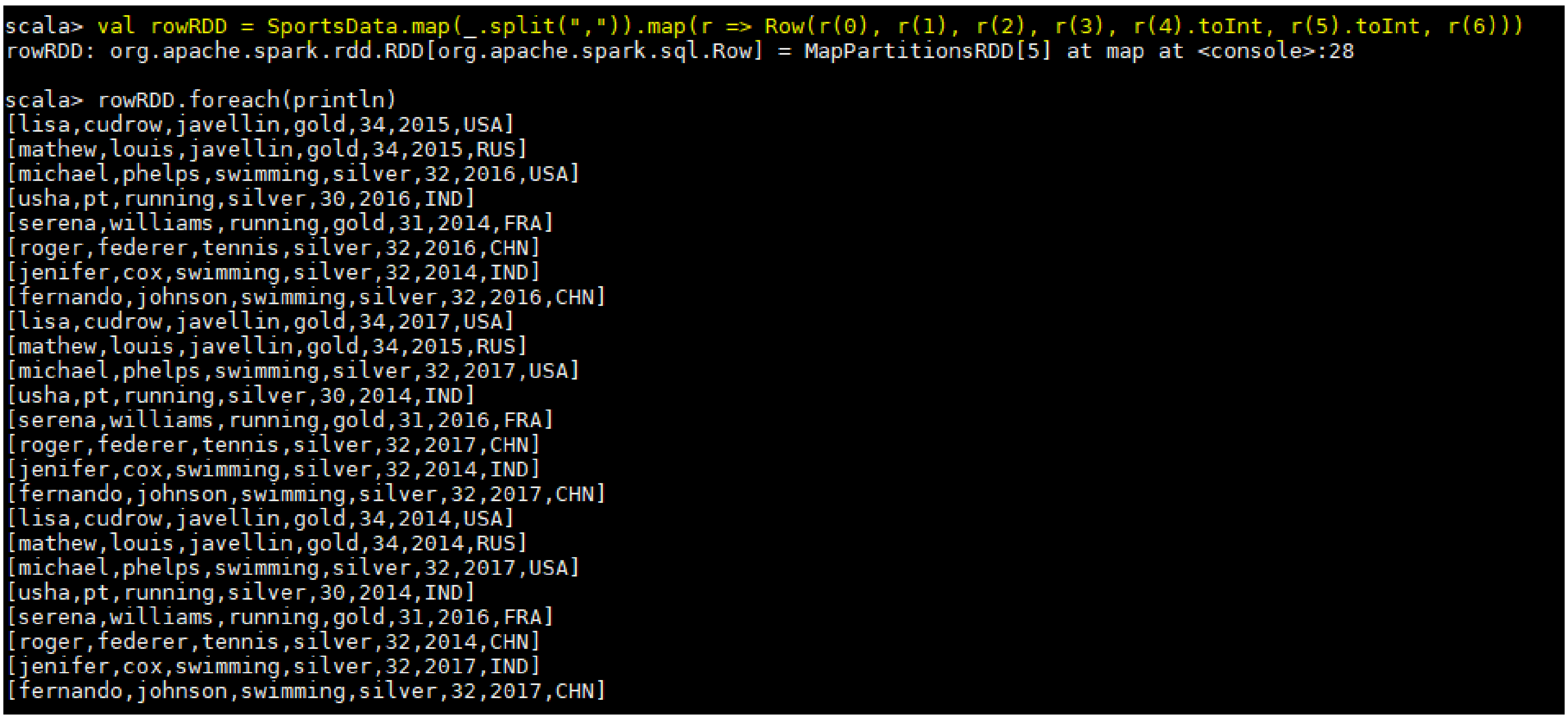
```
scala> val SportsData = sc.textFile("/home/acadgild/Assignment-19/Sports_data.txt")
SportsData: org.apache.spark.rdd.RDD[String] = /home/acadgild/Assignment-19/Sports_data.txt MapPartitionsRDD[3] at textFile at <console>:26

scala> SportsData.foreach(println)
[Stage 0:>                                                          (0 + 0) / 2]roger,federer,tennis,silver,32,2017,CHN
lisa,cudrow,javellin,gold,34,2015,USA
jenifer,cox,swimming,silver,32,2014,IND
mathew,louis,javellin,gold,34,2015,RUS
fernando,johnson,swimming,silver,32,2017,CHN
lisa,cudrow,javellin,gold,34,2014,USA
mathew,louis,javellin,gold,34,2014,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
roger,federer,tennis,silver,32,2014,CHN
jenifer,cox,swimming,silver,32,2017,IND
fernando,johnson,swimming,silver,32,2017,CHN
michael,phelps,swimming,silver,32,2016,USA
usha,pt,running,silver,30,2016,IND
serena,williams,running,gold,31,2014,FRA
roger,federer,tennis,silver,32,2016,CHN
jenifer,cox,swimming,silver,32,2014,IND
fernando,johnson,swimming,silver,32,2016,CHN
lisa,cudrow,javellin,gold,34,2017,USA
mathew,louis,javellin,gold,34,2015,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
```

Since it is a text file we need to define schema too. Below screenshot shows the same-



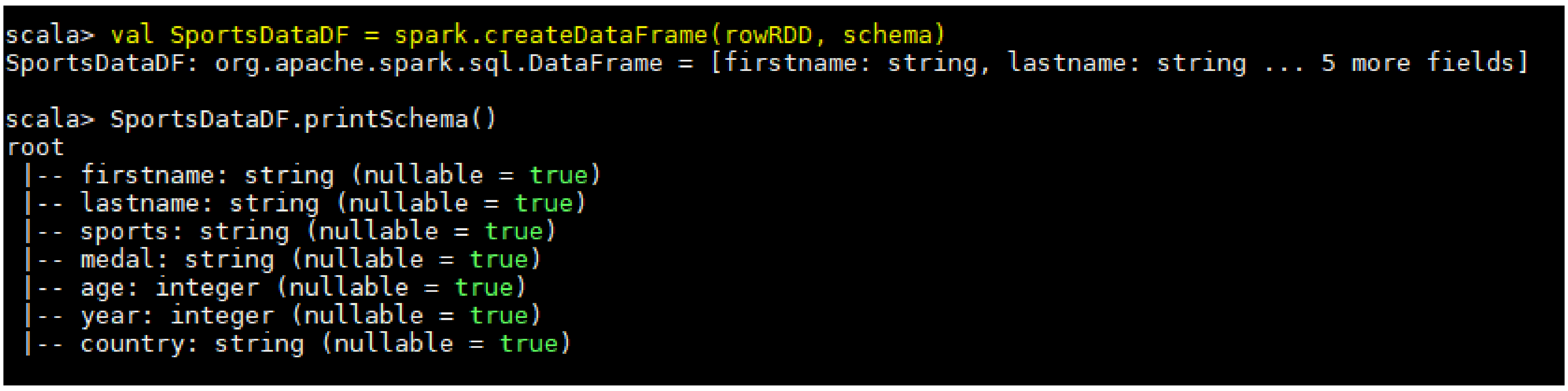
```
scala> val schemaString = "firstname:string,lastname:string,sports:string,medal:string,age:integer,year:integer,country:string"
schemaString: String = firstname:string,lastname:string,sports:string,medal:string,age:integer,year:integer,country:string

scala> val schema = StructType(schemaString.split(",").map(x => StructField(x.split(":")(0), if (x.split(":")(1).equals("string")) StringType else IntegerType, true)
))
schema: org.apache.spark.sql.types.StructType = StructType(StructField(firstname,StringType,true), StructField(lastname,StringType,true), StructField(sports,StringTy
pe,true), StructField(medal,StringType,true), StructField(age,IntegerType,true), StructField(year,IntegerType,true), StructField(country,StringType,true))

scala>
```

Now we are splitting the input file and extracting the rows from it-



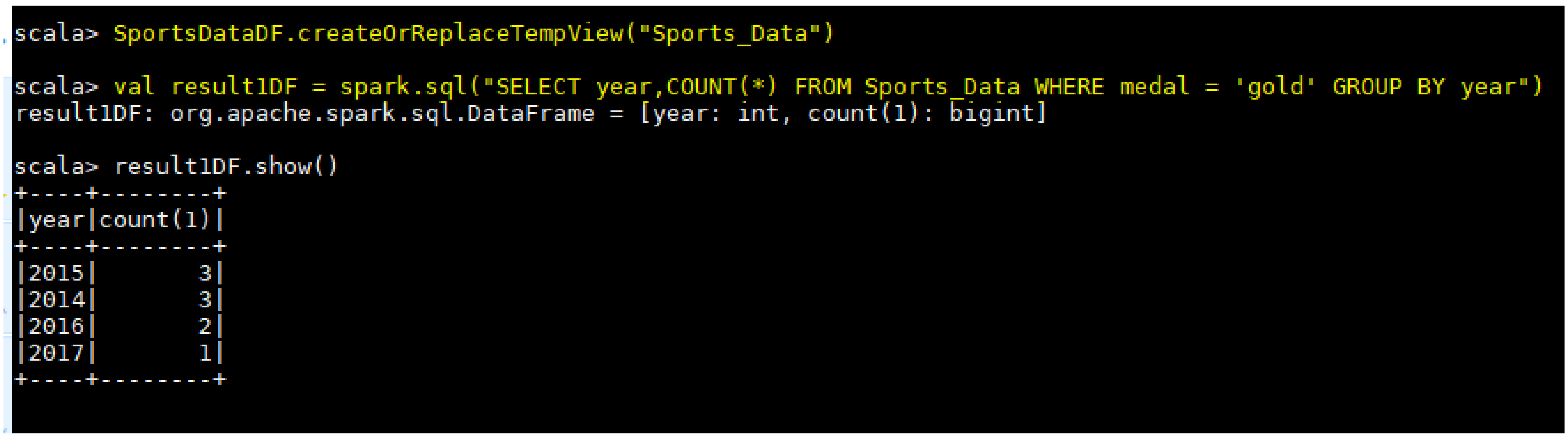
```
scala> val rowRDD = SportsData.map(_.split(",")).map(r => Row(r(0), r(1), r(2), r(3), r(4).toInt, r(5).toInt, r(6)))
rowRDD: org.apache.spark.rdd.RDD[org.apache.spark.sql.Row] = MapPartitionsRDD[5] at map at <console>:28

scala> rowRDD.foreach(println)
[lisa,cudrow,javellin,gold,34,2015,USA]
[mathew,louis,javellin,gold,34,2015,RUS]
[michael,phelps,swimming,silver,32,2016,USA]
[usha,pt,running,silver,30,2016,IND]
[serena,williams,running,gold,31,2014,FRA]
[roger,federer,tennis,silver,32,2016,CHN]
[jenifer,cox,swimming,silver,32,2014,IND]
[fernando,johnson,swimming,silver,32,2016,CHN]
[lisa,cudrow,javellin,gold,34,2017,USA]
[mathew,louis,javellin,gold,34,2015,RUS]
[michael,phelps,swimming,silver,32,2017,USA]
[usha,pt,running,silver,30,2014,IND]
[serena,williams,running,gold,31,2016,FRA]
[roger,federer,tennis,silver,32,2017,CHN]
[jenifer,cox,swimming,silver,32,2014,IND]
[fernando,johnson,swimming,silver,32,2017,CHN]
[lisa,cudrow,javellin,gold,34,2014,USA]
[mathew,louis,javellin,gold,34,2014,RUS]
[michael,phelps,swimming,silver,32,2017,USA]
[usha,pt,running,silver,30,2014,IND]
[serena,williams,running,gold,31,2016,FRA]
[roger,federer,tennis,silver,32,2014,CHN]
[jenifer,cox,swimming,silver,32,2017,IND]
[fernando,johnson,swimming,silver,32,2017,CHN]
```

Now we are creating the dataframe by passing the RDD which reads the file and schema to spark session object-



```
scala> val SportsDataDF = spark.createDataFrame(rowRDD, schema)
SportsDataDF: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 5 more fields]

scala> SportsDataDF.printSchema()
root
 |-- firstname: string (nullable = true)
 |-- lastname: string (nullable = true)
 |-- sports: string (nullable = true)
 |-- medal: string (nullable = true)
 |-- age: integer (nullable = true)
 |-- year: integer (nullable = true)
 |-- country: string (nullable = true)
```

Here we are creating a temporary table first from the dataframe. Finally we can execute our SQL query on the temporary table to find the result-



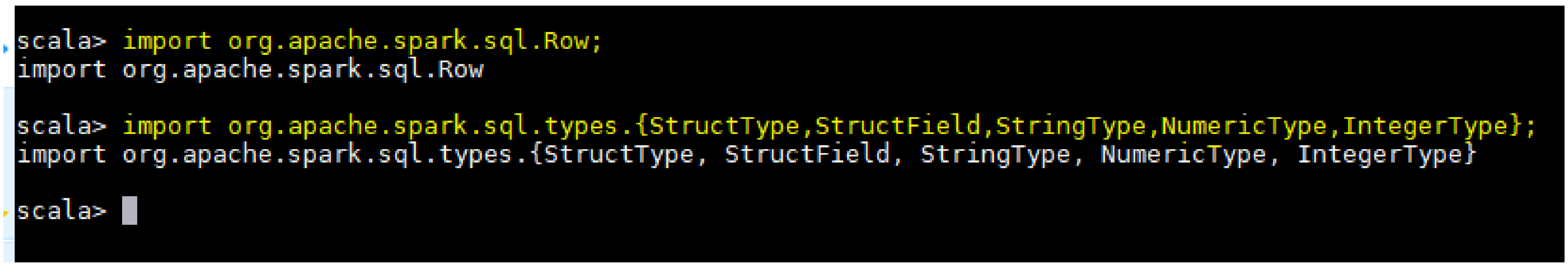
```
scala> SportsDataDF.createOrReplaceTempView("Sports_Data")

scala> val result1DF = spark.sql("SELECT year,COUNT(*) FROM Sports_Data WHERE medal = 'gold' GROUP BY year")
result1DF: org.apache.spark.sql.DataFrame = [year: int, count(1): bigint]

scala> result1DF.show()
+----+--------+
|year|count(1)|
+----+--------+
|2015|       3|
|2014|       3|
|2016|       2|
|2017|       1|
+----+--------+
```

2. How many silver medals have been won by USA in each sport?

Below is the code used to find the result-

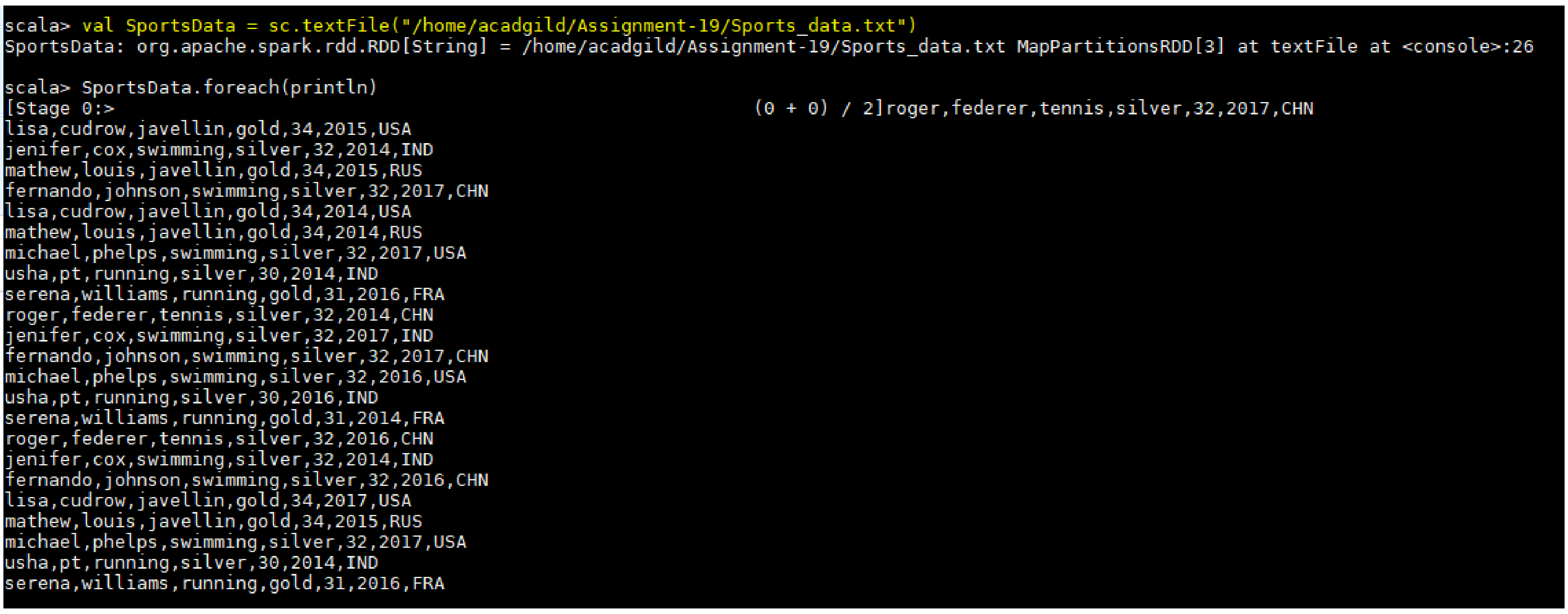
- val SportsData = sc.textFile("/home/acadgild/Assignment-19/Sports_data.txt")
- val schemaString = "firstname:string,lastname:string,sports:string,medal:string,age:integer,year:integer,country:string"
- val schema = StructType(schemaString.split(",").map(x => StructField(x.split(":")(0), if (x.split(":")(1).equals("string")) StringType else IntegerType, true)))
- val rowRDD = SportsData.map(_.split(",")).map(r => Row(r(0), r(1), r(2), r(3), r(4).toInt, r(5).toInt, r(6)))
- val SportsDataDF = spark.createDataFrame(rowRDD, schema)
- SportsDataDF.createOrReplaceTempView("Sports_Data")
- val result2DF = spark.sql("SELECT sports,COUNT(*) FROM Sports_Data WHERE medal = 'silver' and country ='USA' GROUP BY sports")
- result2DF.show()

In order to proceed we need to import some dependencies as shown below-



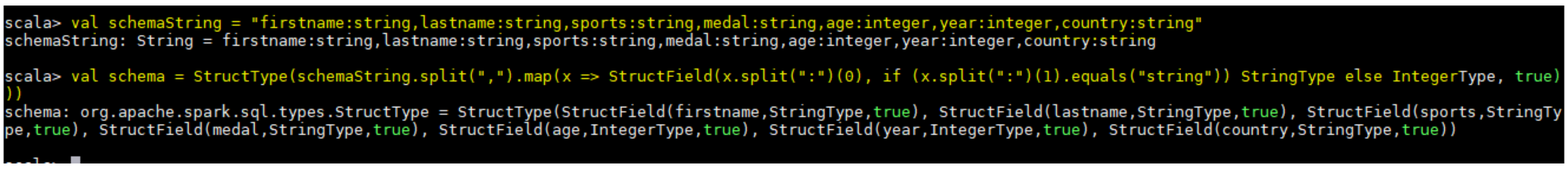
```
scala> import org.apache.spark.sql.Row;
import org.apache.spark.sql.Row

scala> import org.apache.spark.sql.types.{StructType,StructField,StringType,NumericType,IntegerType};
import org.apache.spark.sql.types.{StructType, StructField, StringType, NumericType, IntegerType}

scala> 
```

Now we are creating a RDD which reads from the input file-



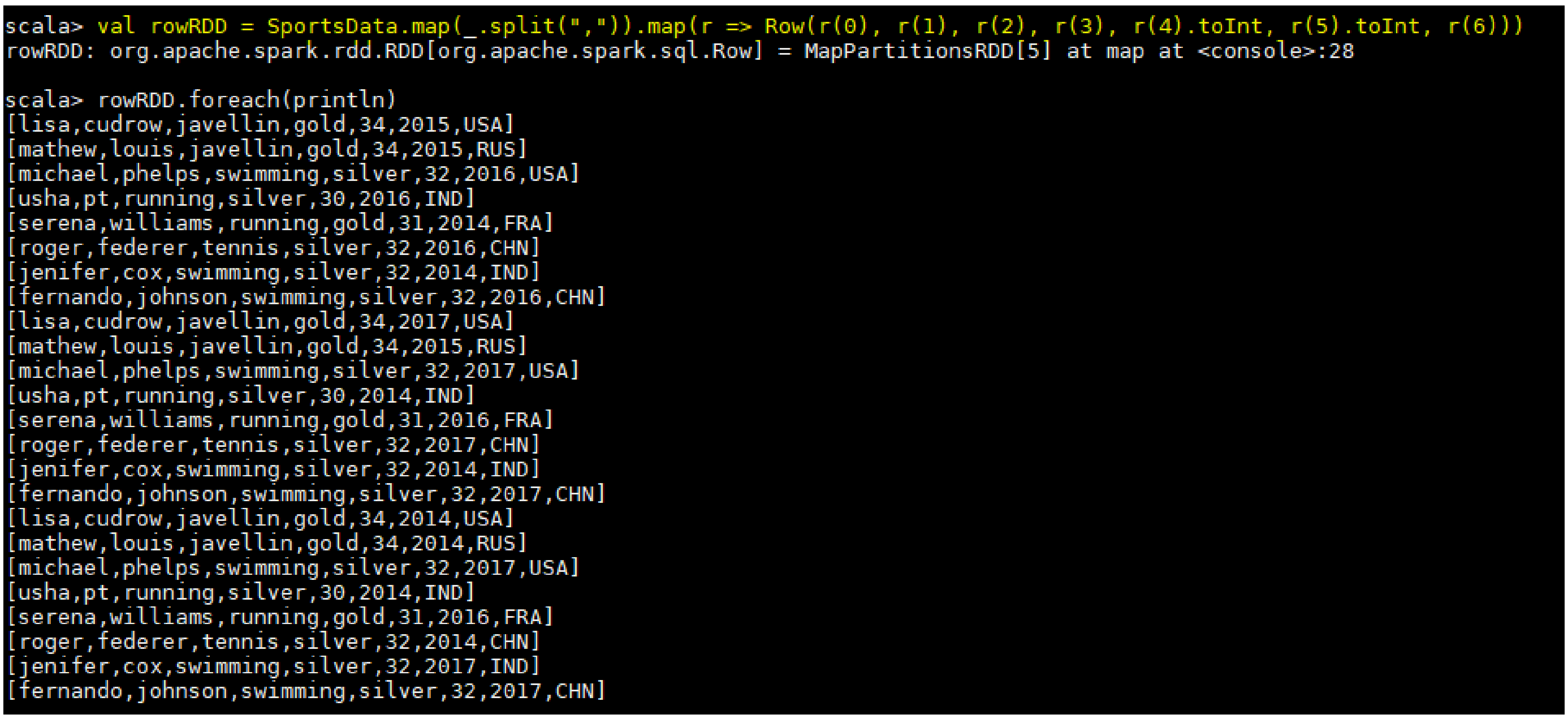
```
scala> val SportsData = sc.textFile("/home/acadgild/Assignment-19/Sports_data.txt")
SportsData: org.apache.spark.rdd.RDD[String] = /home/acadgild/Assignment-19/Sports_data.txt MapPartitionsRDD[3] at textFile at <console>:26

scala> SportsData.foreach(println)
[Stage 0:>                                                  (0 + 0) / 2]roger,federer,tennis,silver,32,2017,CHN
lisa,cudrow,javellin,gold,34,2015,USA
jenifer,cox,swimming,silver,32,2014,IND
mathew,louis,javellin,gold,34,2015,RUS
fernando,johnson,swimming,silver,32,2017,CHN
lisa,cudrow,javellin,gold,34,2014,USA
mathew,louis,javellin,gold,34,2014,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
roger,federer,tennis,silver,32,2014,CHN
jenifer,cox,swimming,silver,32,2017,IND
fernando,johnson,swimming,silver,32,2017,CHN
michael,phelps,swimming,silver,32,2016,USA
usha,pt,running,silver,30,2016,IND
serena,williams,running,gold,31,2014,FRA
roger,federer,tennis,silver,32,2016,CHN
jenifer,cox,swimming,silver,32,2014,IND
fernando,johnson,swimming,silver,32,2016,CHN
lisa,cudrow,javellin,gold,34,2017,USA
mathew,louis,javellin,gold,34,2015,RUS
michael,phelps,swimming,silver,32,2017,USA
usha,pt,running,silver,30,2014,IND
serena,williams,running,gold,31,2016,FRA
```

Since it is a text file we need to define schema too. Below screenshot shows the same-



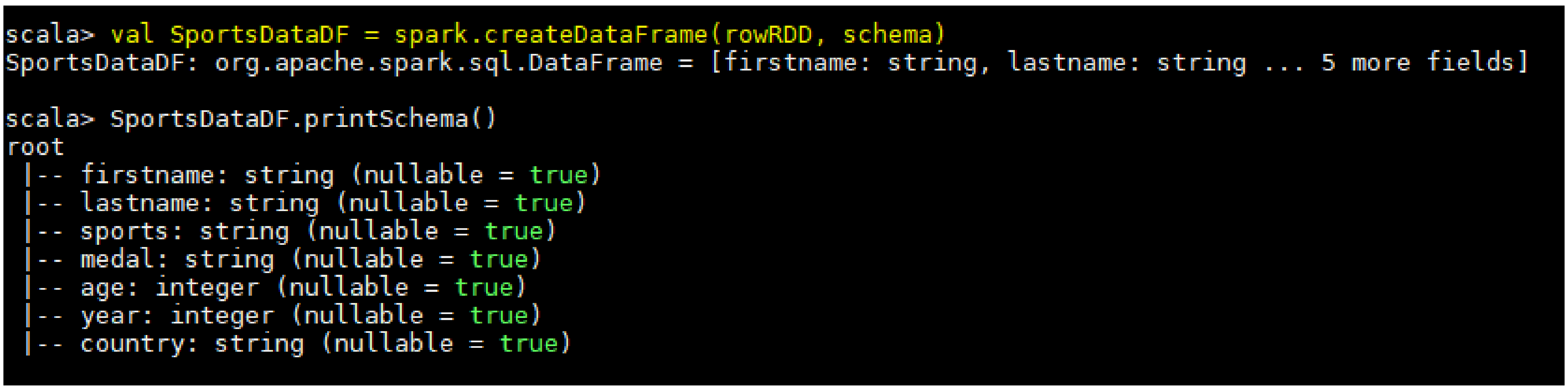
```
scala> val schemaString = "firstname:string,lastname:string,sports:string,medal:string,age:integer,year:integer,country:string"
schemaString: String = firstname:string,lastname:string,sports:string,medal:string,age:integer,year:integer,country:string

scala> val schema = StructType(schemaString.split(",").map(x => StructField(x.split(":")(0), if (x.split(":")(1).equals("string")) StringType else IntegerType, true)
))
schema: org.apache.spark.sql.types.StructType = StructType(StructField(firstname,StringType,true), StructField(lastname,StringType,true), StructField(sports,StringTy
pe,true), StructField(medal,StringType,true), StructField(age,IntegerType,true), StructField(year,IntegerType,true), StructField(country,StringType,true))
```

Now we are splitting the input file and extracting the rows from it-



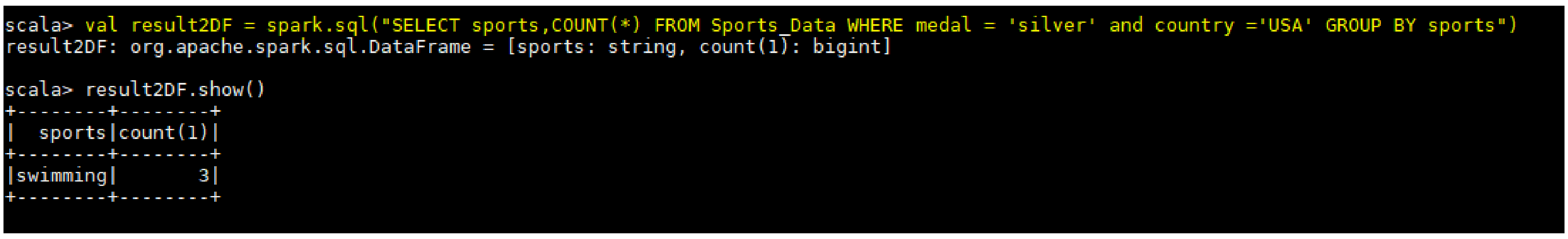
```
scala> val rowRDD = SportsData.map(_.split(",")).map(r => Row(r(0), r(1), r(2), r(3), r(4).toInt, r(5).toInt, r(6)))
rowRDD: org.apache.spark.rdd.RDD[org.apache.spark.sql.Row] = MapPartitionsRDD[5] at map at <console>:28

scala> rowRDD.foreach(println)
[lisa,cudrow,javellin,gold,34,2015,USA]
[mathew,louis,javellin,gold,34,2015,RUS]
[michael,phelps,swimming,silver,32,2016,USA]
[usha,pt,running,silver,30,2016,IND]
[serena,williams,running,gold,31,2014,FRA]
[roger,federer,tennis,silver,32,2016,CHN]
[jenifer,cox,swimming,silver,32,2014,IND]
[fernando,johnson,swimming,silver,32,2016,CHN]
[lisa,cudrow,javellin,gold,34,2017,USA]
[mathew,louis,javellin,gold,34,2015,RUS]
[michael,phelps,swimming,silver,32,2017,USA]
[usha,pt,running,silver,30,2014,IND]
[serena,williams,running,gold,31,2016,FRA]
[roger,federer,tennis,silver,32,2017,CHN]
[jenifer,cox,swimming,silver,32,2014,IND]
[fernando,johnson,swimming,silver,32,2017,CHN]
[lisa,cudrow,javellin,gold,34,2014,USA]
[mathew,louis,javellin,gold,34,2014,RUS]
[michael,phelps,swimming,silver,32,2017,USA]
[usha,pt,running,silver,30,2014,IND]
[serena,williams,running,gold,31,2016,FRA]
[roger,federer,tennis,silver,32,2014,CHN]
[jenifer,cox,swimming,silver,32,2017,IND]
[fernando,johnson,swimming,silver,32,2017,CHN]
```

Now we are creating the dataframe by passing the RDD which reads the file and schema to spark session object-



```
scala> val SportsDataDF = spark.createDataFrame(rowRDD, schema)
SportsDataDF: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 5 more fields]

scala> SportsDataDF.printSchema()
root
 |-- firstname: string (nullable = true)
 |-- lastname: string (nullable = true)
 |-- sports: string (nullable = true)
 |-- medal: string (nullable = true)
 |-- age: integer (nullable = true)
 |-- year: integer (nullable = true)
 |-- country: string (nullable = true)
```

Finally we can execute our query by applying it on the temporary table created-



```
scala> val result2DF = spark.sql("SELECT sports,COUNT(*) FROM Sports_Data WHERE medal = 'silver' and country ='USA' GROUP BY sports")
result2DF: org.apache.spark.sql.DataFrame = [sports: string, count(1): bigint]

scala> result2DF.show()
+--------+--------+
|  sports|count(1)|
+--------+--------+
|swimming|       3|
+--------+--------+
```