**Using udfs on dataframe**

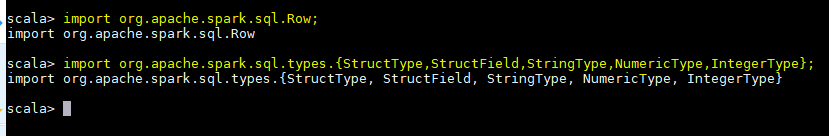
1. **Change firstname, lastname columns into Mr.first\_two\_letters\_of\_firstname<space>lastname for example - michael, phelps becomes Mr.mi phelps**

In order to proceed we need to define the dataframe first for the text input file we have- Below is code which is used to find the result-

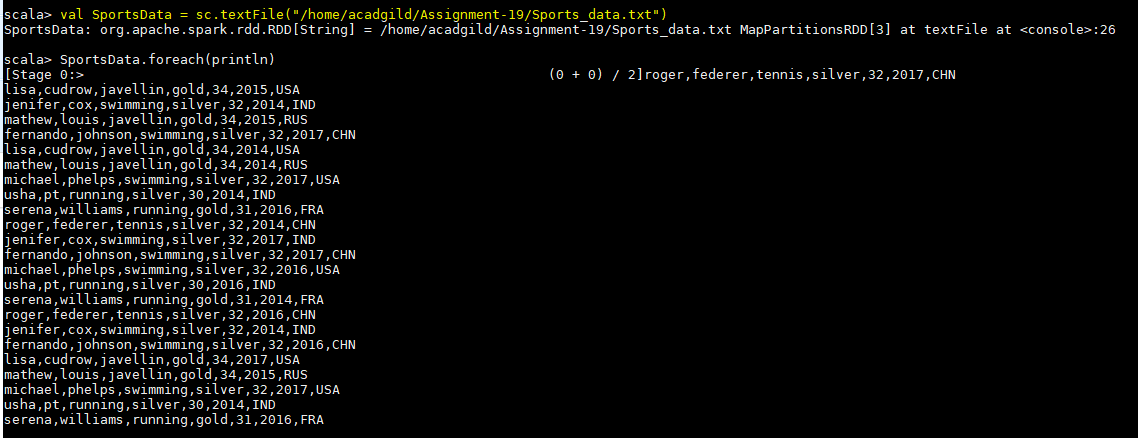
* import org.apache.spark.sql.functions.udf
* val SportsData = sc.textFile("/home/acadgild/Assignment-19/Sports\_data.txt")
* val schemaString = "firstname:string,lastname:string,sports:string,medal:string,age:integer,year:integer,country:s tring"
* val schema = StructType(schemaString.split(",").map(x => 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 Name = udf((firstname: String, lastname: String) => "Mr. ".concat(firstname.substring(0,2)).concat(" ")concat(lastname))
* spark.udf.register("Full\_Name", Name)
* val fname = spark.sql("SELECT Full\_Name(firstname, lastname) FROM Sports\_Data")
* fname.show()

Now lets see each and every line one by one.

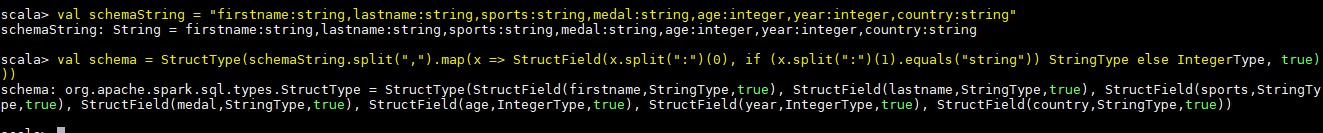
In order to proceed we need to import some dependencies as shown below-



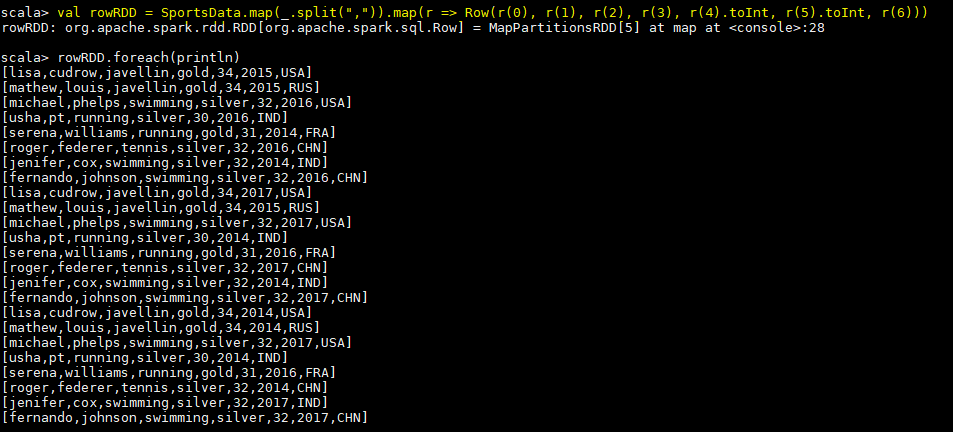
Now we are creating a RDD which reads from the input file-



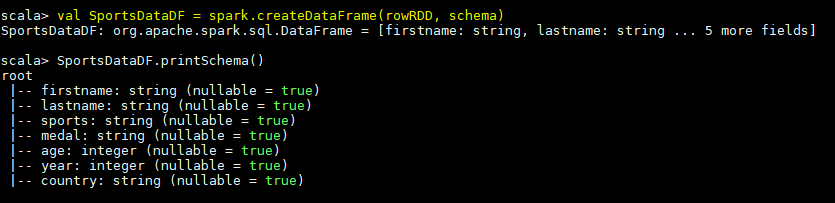
Since it is a text file we need to define schema too. Below screenshot shows the same-



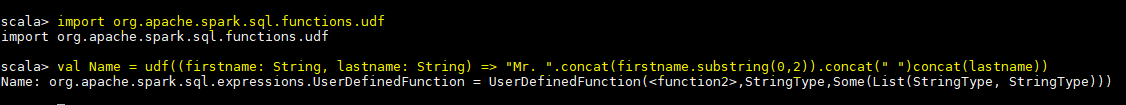
Now we are splitting the input file and extracting the rows from it-

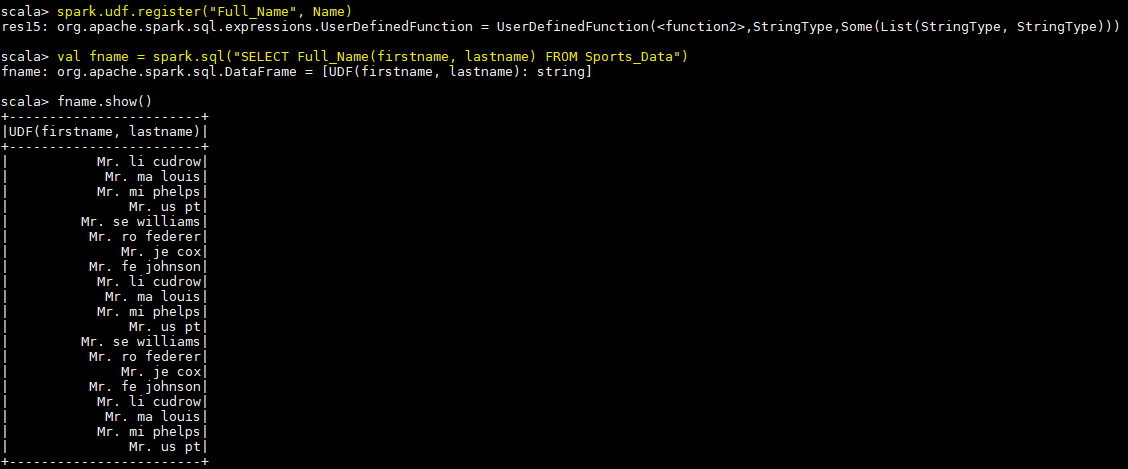


Now we are creating the dataframe by passing the RDD which reads the file and schema to spark session object-



Here we are defining the UDF which will take 2 strings (columns) as input and will concatenate them with Mr. appended in it-



Now we need to register the UDF. Here we doing the same and giving it an alias as Full\_Name. Finally we can apply this UDF on the columns to give the required result-

# Add a new column called ranking using udfs on dataframe, where : gold medalist, with age >= 32 are ranked as pro

**gold medalists, with age <= 31 are ranked amateur silver medalist, with age >= 32 are ranked as expert silver medalists, with age <= 31 are ranked rookie**

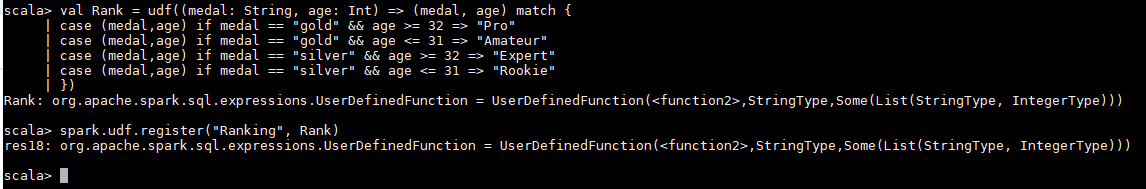
Here we will work with the dataframe created in above problem. We just need to write an UDF. Below is the UDF that we have used to define the new column-

* val Rank = udf((medal: String, age: Int) => (medal, age) match { case (medal,age) if medal == "gold" && age >= 32 => "Pro"

case (medal,age) if medal == "gold" && age <= 31 => "Amateur" case (medal,age) if medal == "silver" && age >= 32 => "Expert" case (medal,age) if medal == "silver" && age <= 31 => "Rookie"

* })

Here we are classifying each player based on age and the medal he has got-



Below code shows the registering of UDF and command to add a new column-

* spark.udf.register("Ranking", Rank)
* val RankRDD = SportsDataDF.withColumn("Ranking",Rank(SportsDataDF.col("medal"),SportsDataDF.col("age ")))

Below shows the final result for same-

