Accadgild_Session_19_Assignment_19.2_Solutions

Problem Statement

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

2. 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

Code For Question 1 And 2

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

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

val Sports_data = sc.textFile("/home/acadgild/Assignment-20/Sports_data.txt")

val schemaString =
"firstname:string,lastname:string,sports:string,medal_type:string,age:integer,year:integer,coun
try:string"

val schema = StructType(schemaString.split(",").map(fieldInfo =>
StructField(fieldInfo.split(":")(0), if (fieldInfo.split(":")(1).equals("string")) StringType else
IntegerType, true)))

val rowRDD = Sports_data.map(_.split(",")).map(r => Row(r(0), r(1), r(2), r(3), r(4).toInt,
r(5).toInt, r(6)))

val SportsDF = spark.createDataFrame(rowRDD, schema)
```

Accadgild_Session_19_Assignment_19.2_Solutions

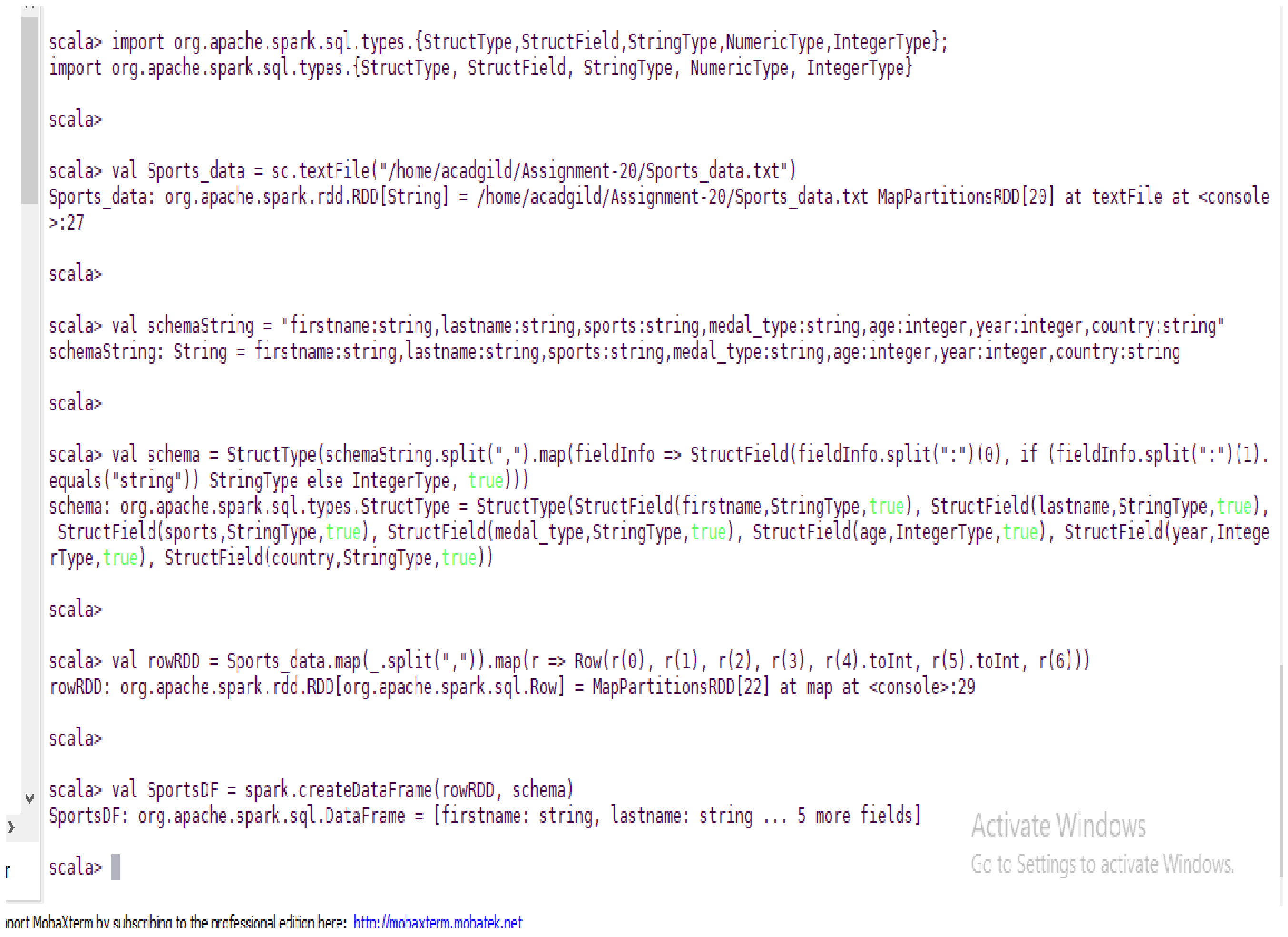
Question 1 Code:

```
SportsDF.createOrReplaceTempView("SportsData")

import org.apache.spark.sql.functions.udf

val Name = udf((firstname: String, lastname: String) => "Mr.
".concat(firstname.substring(0,2)).concat(" ")concat(lastname))

spark.udf.register("Full_Name", Name)

// Register the UDF with our SQLContext

val fname = spark.sql("SELECT Full_Name(firstname, lastname) FROM SportsData")

fname.show()
```

Question 2 Code:

```
val Rank = udf((medal_type: String, age: Int) => (medal_type, age) match {

case (medal_type,age) if medal_type == "gold" && age >= 32 => "Pro"

case (medal_type,age) if medal_type == "gold" && age <= 31 => "Amateur"

case (medal_type,age) if medal_type == "silver" && age >= 32 => "Expert"

case (medal_type,age) if medal_type == "silver" && age <= 31 => "Rookie"

})


spark.udf.register("Ranking", Rank)

// Register the UDF with our SQLContext


val RankRDD =
SportsDF.withColumn("Ranking",Rank(SportsDF.col("medal_type"),SportsDF.col("age")))

RankRDD.show()
```

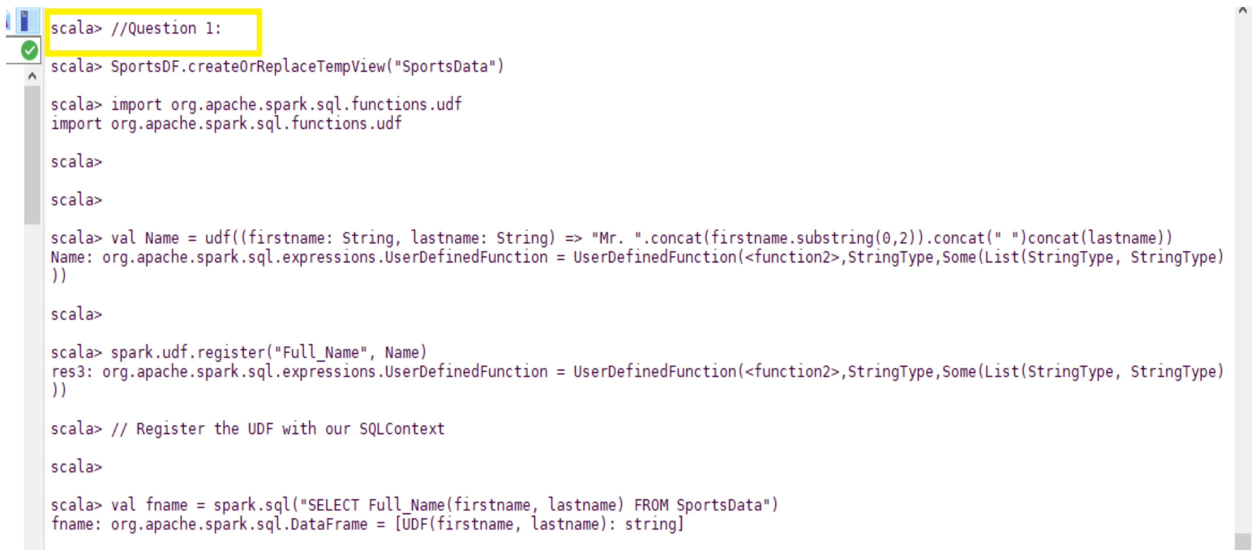
Accadgild_Session_19_Assignment_19.2_Solutions



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

scala>

scala> val Sports_data = sc.textFile("/home/acadgild/Assignment-20/Sports_data.txt")
Sports_data: org.apache.spark.rdd.RDD[String] = /home/acadgild/Assignment-20/Sports_data.txt MapPartitionsRDD[20] at textFile at <console
>:27

scala>

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

scala>

scala> val schema = StructType(schemaString.split(",").map(fieldInfo => StructField(fieldInfo.split(":")(0), if (fieldInfo.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,StringType,true), StructField(medal_type,StringType,true), StructField(age,IntegerType,true), StructField(year,Intege
rType,true), StructField(country,StringType,true))

scala>

scala> val rowRDD = Sports_data.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[22] at map at <console>:29

scala>

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

scala>
```

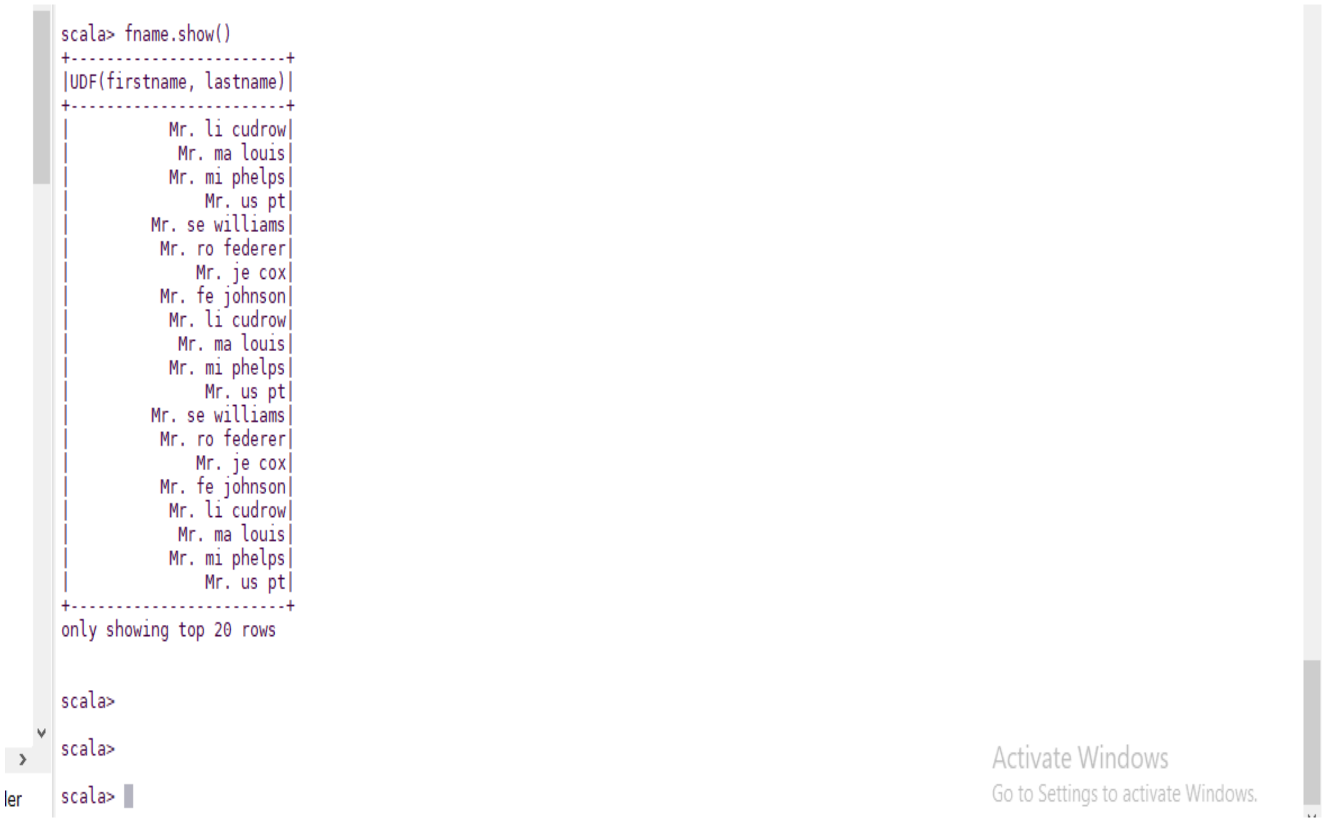
Activate Windows
Go to Settings to activate Windows.

ıport MobaXterm by subscribing to the professional edition here: http://mobaxterm.mobatek.net

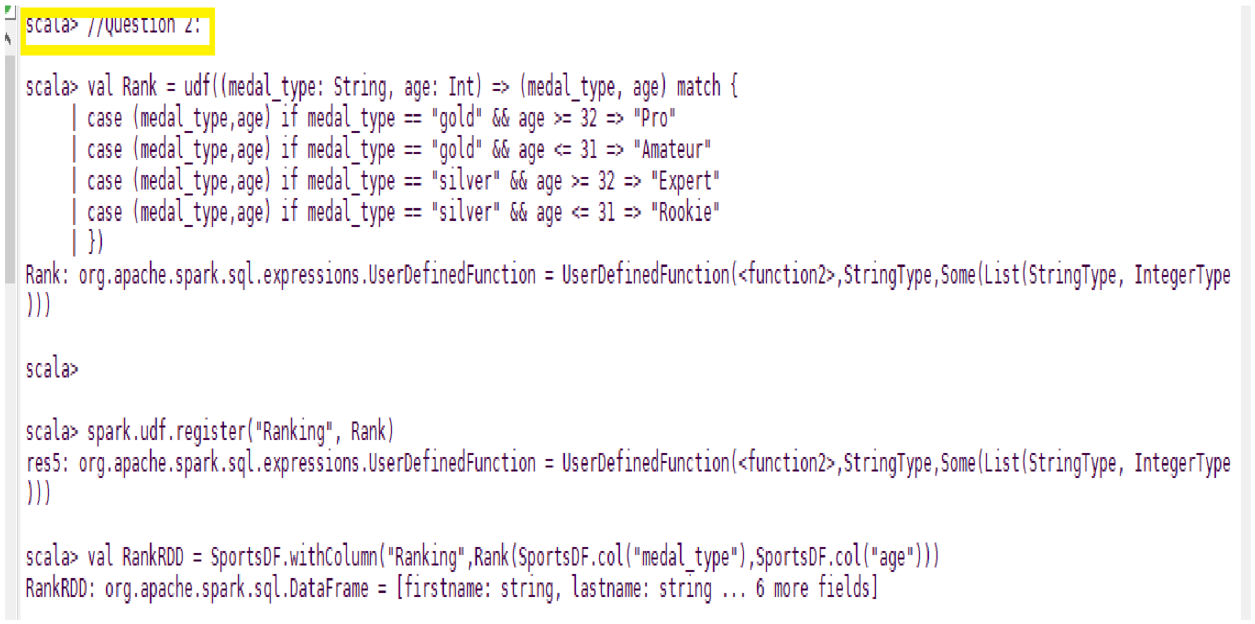


```
scala> //Question 1:

scala> SportsDF.createOrReplaceTempView("SportsData")

scala> import org.apache.spark.sql.functions.udf
import org.apache.spark.sql.functions.udf

scala>

scala>

scala> val Name = udf((firstname: String, lastname: String) => "Mr. ".concat(firstname.substring(0,2)).concat(" ")concat(lastname))
Name: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function2>,StringType,Some(List(StringType, StringType)
))

scala>

scala> spark.udf.register("Full_Name", Name)
res3: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function2>,StringType,Some(List(StringType, StringType)
))

scala> // Register the UDF with our SQLContext

scala>

scala> val fname = spark.sql("SELECT Full_Name(firstname, lastname) FROM SportsData")
fname: org.apache.spark.sql.DataFrame = [UDF(firstname, lastname): string]
```

Accadgild_Session_19_Assignment_19.2_Solutions

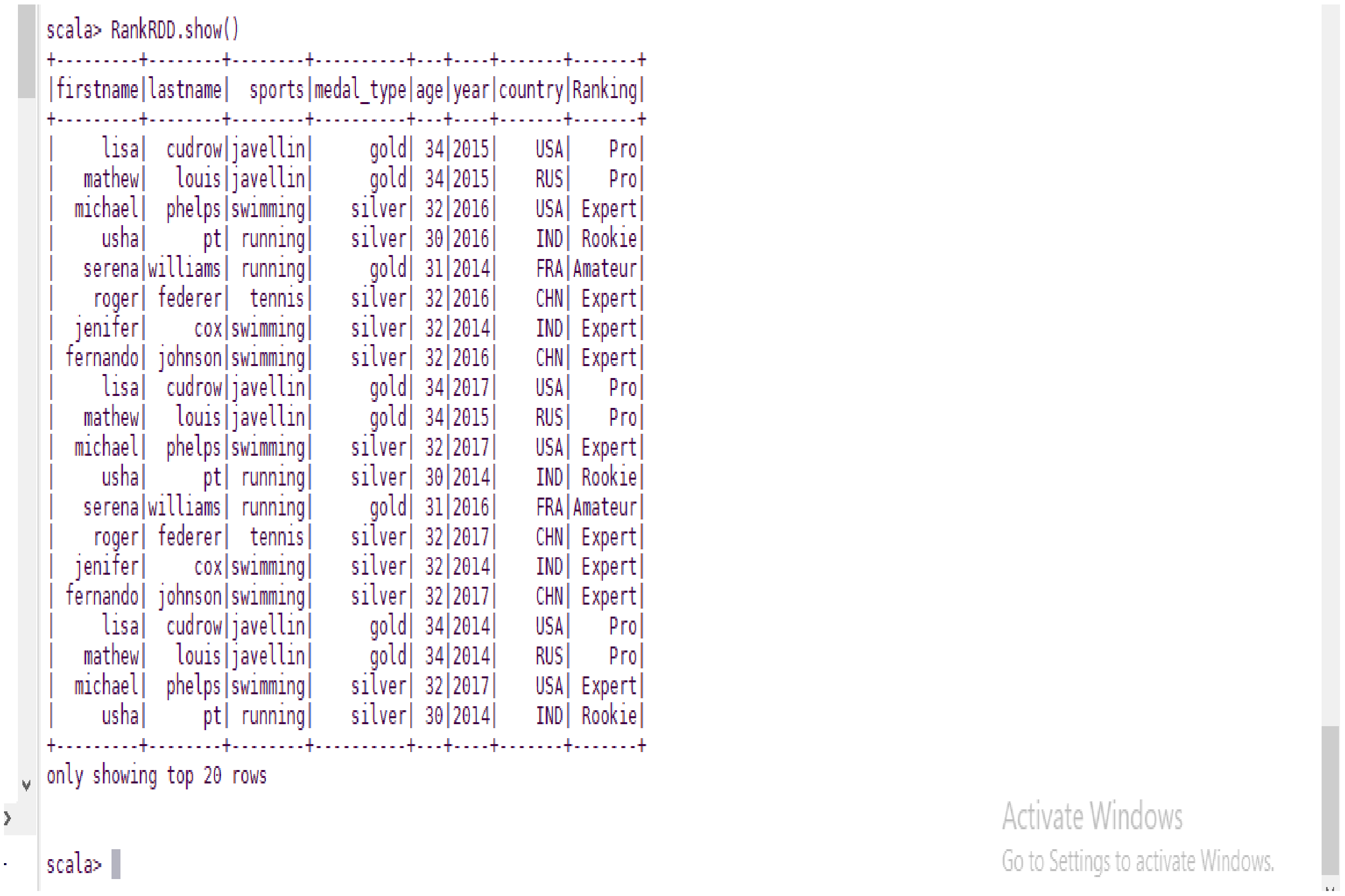


```
scala> fname.show()
+-----------------------+
|UDF(firstname, lastname)|
+-----------------------+
|          Mr. li cudrow|
|           Mr. ma louis|
|          Mr. mi phelps|
|              Mr. us pt|
|        Mr. se williams|
|         Mr. ro federer|
|             Mr. je cox|
|         Mr. fe johnson|
|          Mr. li cudrow|
|           Mr. ma louis|
|          Mr. mi phelps|
|              Mr. us pt|
|        Mr. se williams|
|         Mr. ro federer|
|             Mr. je cox|
|         Mr. fe johnson|
|          Mr. li cudrow|
|           Mr. ma louis|
|          Mr. mi phelps|
|              Mr. us pt|
+-----------------------+
only showing top 20 rows


scala>

scala>

scala>
```



```
scala> //Question 2:

scala> val Rank = udf((medal_type: String, age: Int) => (medal_type, age) match {
     | case (medal_type,age) if medal_type == "gold" && age >= 32 => "Pro"
     | case (medal_type,age) if medal_type == "gold" && age <= 31 => "Amateur"
     | case (medal_type,age) if medal_type == "silver" && age >= 32 => "Expert"
     | case (medal_type,age) if medal_type == "silver" && age <= 31 => "Rookie"
     | })
Rank: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function2>,StringType,Some(List(StringType, IntegerType
)))

scala>

scala> spark.udf.register("Ranking", Rank)
res5: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function2>,StringType,Some(List(StringType, IntegerType
)))

scala> val RankRDD = SportsDF.withColumn("Ranking",Rank(SportsDF.col("medal_type"),SportsDF.col("age")))
RankRDD: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 6 more fields]
```

Accadgild_Session_19_Assignment_19.2_Solutions



```
scala> RankRDD.show()
+---------+--------+--------+----------+---+----+-------+-------+
|firstname|lastname|  sports|medal_type|age|year|country|Ranking|
+---------+--------+--------+----------+---+----+-------+-------+
|     lisa|  cudrow|javellin|      gold| 34|2015|    USA|    Pro|
|   mathew|   louis|javellin|      gold| 34|2015|    RUS|    Pro|
|  michael|  phelps|swimming|    silver| 32|2016|    USA| Expert|
|     usha|      pt| running|    silver| 30|2016|    IND| Rookie|
|   serena|williams| running|      gold| 31|2014|    FRA|Amateur|
|    roger| federer|  tennis|    silver| 32|2016|    CHN| Expert|
|   jenifer|     cox|swimming|    silver| 32|2014|    IND| Expert|
| fernando| johnson|swimming|    silver| 32|2016|    CHN| Expert|
|     lisa|  cudrow|javellin|      gold| 34|2017|    USA|    Pro|
|   mathew|   louis|javellin|      gold| 34|2015|    RUS|    Pro|
|  michael|  phelps|swimming|    silver| 32|2017|    USA| Expert|
|     usha|      pt| running|    silver| 30|2014|    IND| Rookie|
|   serena|williams| running|      gold| 31|2016|    FRA|Amateur|
|    roger| federer|  tennis|    silver| 32|2017|    CHN| Expert|
|  jenifer|     cox|swimming|    silver| 32|2014|    IND| Expert|
| fernando| johnson|swimming|    silver| 32|2017|    CHN| Expert|
|     lisa|  cudrow|javellin|      gold| 34|2014|    USA|    Pro|
|   mathew|   louis|javellin|      gold| 34|2014|    RUS|    Pro|
|  michael|  phelps|swimming|    silver| 32|2017|    USA| Expert|
|     usha|      pt| running|    silver| 30|2014|    IND| Rookie|
+---------+--------+--------+----------+---+----+-------+-------+
only showing top 20 rows


scala>
```

Submitted By Shishir