Session 21: SPARK SQL II

Task – 1.1 - What are the total number of gold medal winners every year

Step -1 – we are creating a RDD from Input DataSet

```
scala> val inputData = sc.textFile("/user/acadgild/hadoop/Sports_data.txt")
inputData: org.apache.spark.rdd.RDD[String] = /user/acadgild/hadoop/Sports_data.txt MapPartitionsRDD[112] at textFile at <console>:26

scala> inputData.foreach(println')
<console>:1: error: unclosed character literal
inputData.foreach(println')
                          ^

scala> inputData.foreach(println)
firstname,lastname,sports,medal_type,age,year,country
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
```

Step -2 – we are defining a schema since it is a text file and splitting the input file using the delimiters and extracting the rows from it.

```
scala> val schemaString = "firstname:string,lastname:string,sports:string,medal_type:string,age:string,year:string,country:string"
schemaString: String = firstname:string,lastname:string,sports:string,medal_type:string,age:string,year:string,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,StringType,true), StructField(medal_type,StringType,true), StructField(age,StringType,true), StructField(year,StringT
ype,true), StructField(country,StringType,true))

scala> val rowRDD = inputData.map(_.split(",")).map(r=>Row(r(0), r(1), r(2), r(3), r(4), r(5), r(6)))
rowRDD: org.apache.spark.rdd.RDD[org.apache.spark.sql.Row] = MapPartitionsRDD[114] at map at <console>:30

scala>
```

We have created the **dataframe** by passing the RDD which reads the file and schema to spark session object-

The schema of the created **Dataframe** can be seen below.

```
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_type: string (nullable = true)
 |-- age: string (nullable = true)
 |-- year: string (nullable = true)
 |-- country: string (nullable = true)
```

Expected Result

Now, we are using the simple SQL query so that we can execute our query by applying it on the temporary table created.

```
scala> val total_gold_per_year = spark.sql("select year,count(*) from SportsData where medal_type='gold' group by year")
total_gold_per_year: org.apache.spark.sql.DataFrame = [year: string, count(1): bigint]

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

Task – 1.2 - How many silver medals have been won by USA in each sport

```
scala> val silver_usa_per_sport = spark.sql("select sports,count(*) from SportsData where medal_type='silver' and country = 'USA' group b
y sports")
silver_usa_per_sport: org.apache.spark.sql.DataFrame = [sports: string, count(1): bigint]

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

Task – 2.1 - Change firstname, lastname columns into Mr.first_two_letters_of_firstname<space>lastname

for example - michael, phelps becomes Mr.mi phelps t

```
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> spark.udf.register("FUll_Name",Name)
res32: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function2>,StringType,Some(List(StringType, StringType
)))

scala> val fname = spark.sql("select Full_Name(firstname,lastname) from SportsData").show()
+----------------------+
|UDF(firstname, lastname)|
+----------------------+
|         Mr. fi 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|
+----------------------+
only showing top 20 rows

fname: Unit = ()
```

Task – 2.2 - 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

Here we are classifying each player based on age and medals:

```
scala> val ranking = udf((medal:String,age:Int)=>(medal,age)match{case(medal,age)if medal=="gold" && age>=32=>"Pro" case(medal,age)if med
al=="gold" && age<=32=>"amateur" case(medal,age)if medal=="silver" && age>=32=>"expert" case(medal,age)if medal=="silver" && age<=32=>"ro
okie" })
ranking: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function2>,StringType,Some(List(StringType, IntegerT
ype)))
```

Below code shows registering of UDF and command to add a new column and the expected output:

```
scala> spark.udf.register("Ranks",ranking)
res35: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function2>,StringType,Some(List(StringType, IntegerTyp
e)))

scala> val RankingRDD = SportsDataDF.withColumn("Ranks", ranking(SportsDataDF.col("medal_type"),SportsDataDF.col("age")))
RankingRDD: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 6 more fields]

scala> RankingRDD.show()
+---------+--------+--------+----------+---+----+-------+-------+
|firstname|lastname|  sports|medal_type|age|year|country|  Ranks|
+---------+--------+--------+----------+---+----+-------+-------+
|firstname|lastname|  sports|medal_type|age|year|country|   null|
|     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|
+---------+--------+--------+----------+---+----+-------+-------+
```