BIG DATA HADOOP AND SPARK DEVLOPMENT

ASSIGNMENT 21

Table of Contents:

BIG DATA HADOOPAND SPARK DEVELOPMENT

1. Introduction

In this assignment, the given tasks are performed and Output of the tasks are recorded in the form of Screenshots.

2. Objective

This Assignment consolidates the deeper understanding of the Session – 21
SPARK SQL 2

3. Problem Statement

- Task 1

  Using spark-sql, Find:
  1. What are the total number of gold medal winners every year
  2. How many silver medals have been won by USA in each sport

- Task 2

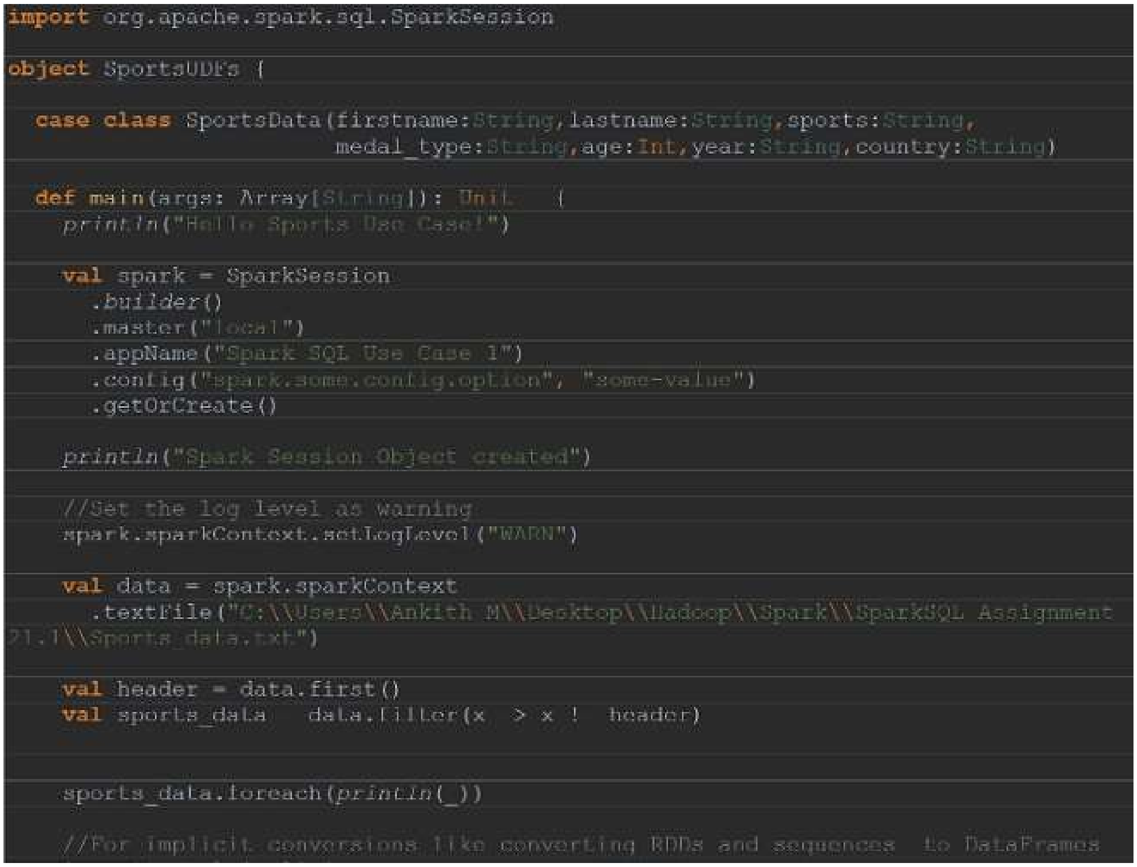
  Using udfs on dataframe

  1. Change firstname, lastname columns into Mr.first_two_letters_of_firstnamelastname for example - michael, phelps becomes Mr.mi phelps

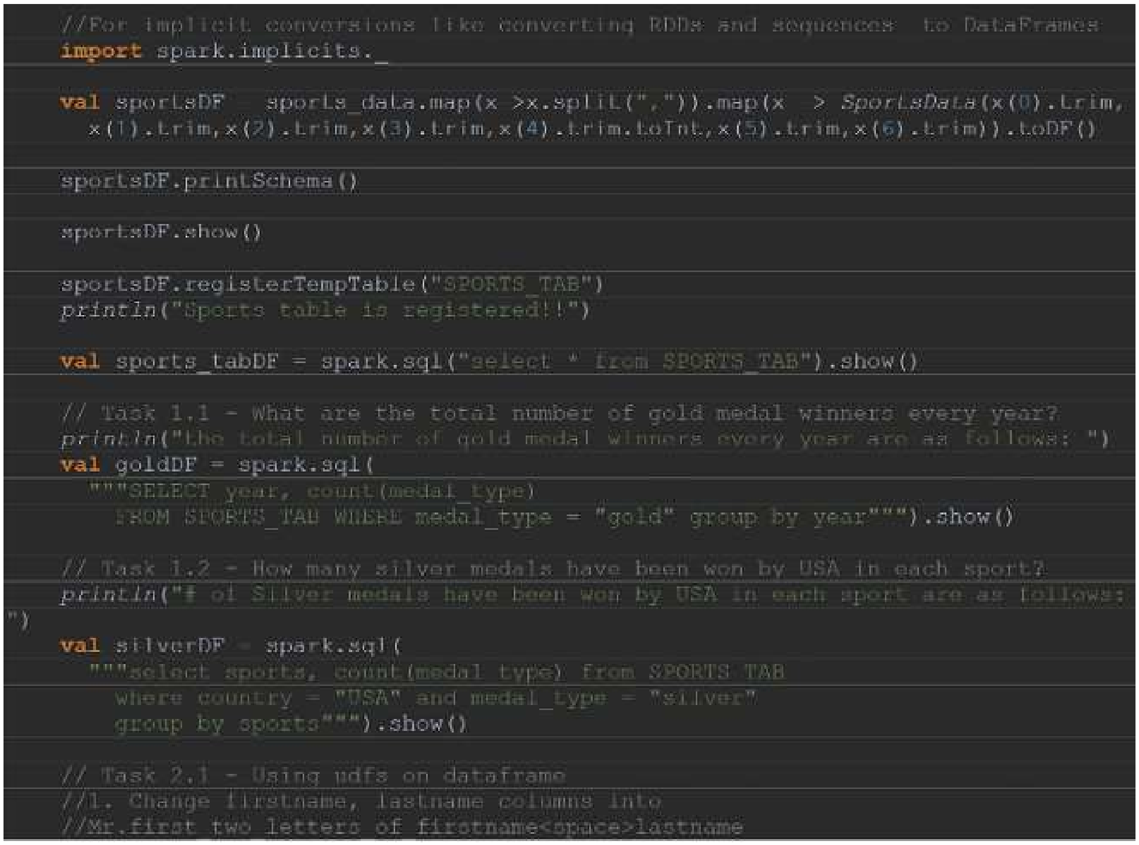
  2. Add a new column called ranking using udfs on dataframe, where : gold medalist, with age >= 32 are ranked as pro gold medalists, with age <= 31 are ranked amateur silver medalist, with age >= 32 are ranked as expert silver medalists, with age <= 31 are ranked rookie

Expected Output

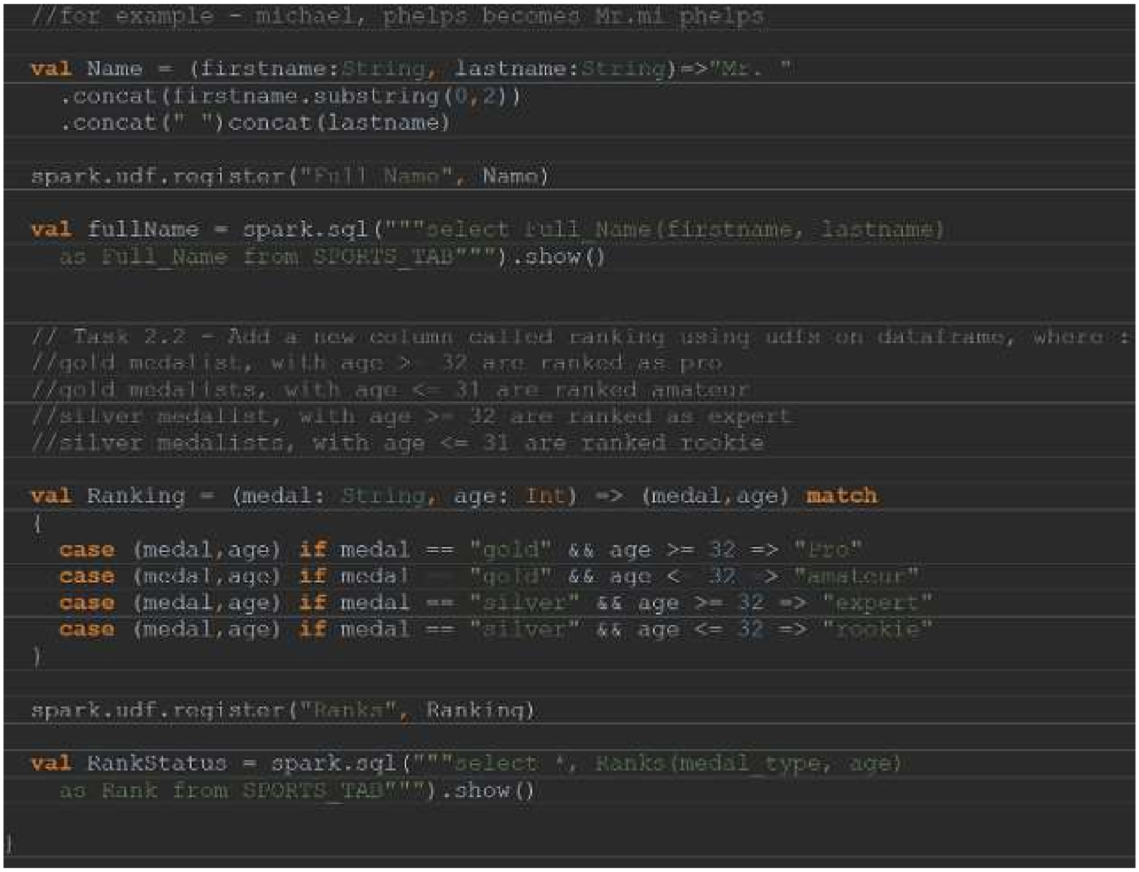
The whole code



```
import org.apache.spark.sql.SparkSession

object SportsUDFs {

  case class SportsData(firstname:String,lastname:String,sports:String,
                        medal_type:String,age:Int,year:String,country:String)

  def main(args: Array[String]): Unit = {
    println("Hello Sports Use Case!")

    val spark = SparkSession
      .builder()
      .master("local")
      .appName("Spark SQL Use Case 1")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")

    //Set the log level as warning
    spark.sparkContext.setLogLevel("WARN")

    val data = spark.sparkContext
      .textFile("C:\\Users\\Ankith M\\Desktop\\Hadoop\\Spark\\SparkSQL Assignment
21.1\\Sports_data.txt")

    val header = data.first()
    val sports_data = data.filter(x => x != header)


    sports_data.foreach(println(_))

    //For implicit conversions like converting RDDs and sequences  to DataFrames
```



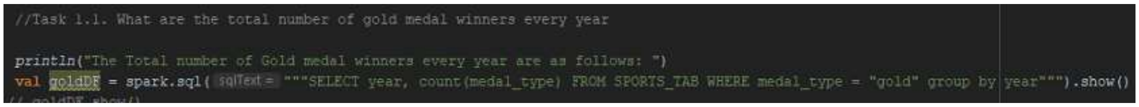
```
    //For implicit conversions like converting RDDs and sequences  to DataFrames
    import spark.implicits._

    val sportsDF   sports_data.map(x >x.split(",")).map(x  > SportsData(x(0).trim,
      x(1).trim,x(2).trim,x(3).trim,x(4).trim.toInt,x(5).trim,x(6).trim)).toDF()

    sportsDF.printSchema()

    sportsDF.show()

    sportsDF.registerTempTable("SPORTS_TAB")
    println("Sports table is registered!!")

    val sports_tabDF = spark.sql("select * from SPORTS_TAB").show()

    // Task 1.1 - What are the total number of gold medal winners every year?
    println("the total number of gold medal winners every year are as follows: ")
    val goldDF = spark.sql(
      """SELECT year, count(medal_type)
        FROM SPORTS_TAB WHERE medal_type = "gold" group by year""").show()

    // Task 1.2 - How many silver medals have been won by USA in each sport?
    println("# of Silver medals have been won by USA in each sport are as follows:
")
    val silverDF = spark.sql(
      """select sports, count(medal_type) from SPORTS_TAB
        where country = "USA" and medal_type = "silver"
        group by sports""").show()

    // Task 2.1 - Using udfs on dataframe
    //1. Change firstname, lastname columns into
    //Mr.first two letters of firstname<space>lastname
```



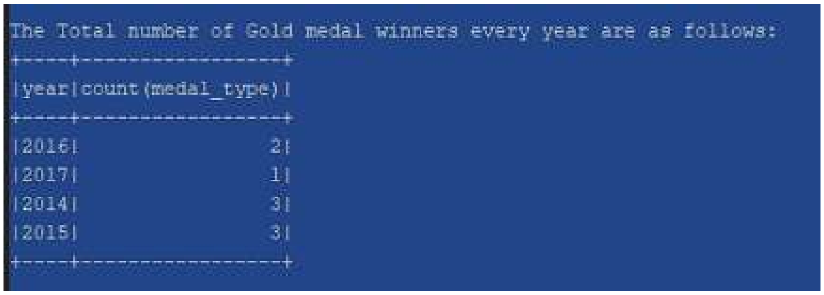
```
  //for example - michael, phelps becomes Mr.mi phelps

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

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

  val fullName = spark.sql("""select Full_Name(firstname, lastname)
    as Full_Name from SPORTS_TAB""").show()


  // 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

  val Ranking = (medal: String, age: Int) => (medal,age) match
  {
    case (medal,age) if medal == "gold" && age >= 32 => "Pro"
    case (medal,age) if medal == "gold" && age <  32 => "amateur"
    case (medal,age) if medal == "silver" && age >= 32 => "expert"
    case (medal,age) if medal == "silver" && age <= 32 => "rookie"
  }

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

  val RankStatus = spark.sql("""select *, Ranks(medal_type, age)
    as Rank from SPORTS_TAB""").show()

}
```

- Task 1

  Using spark-sql, Find:
- What are the total number of gold medal winners every year

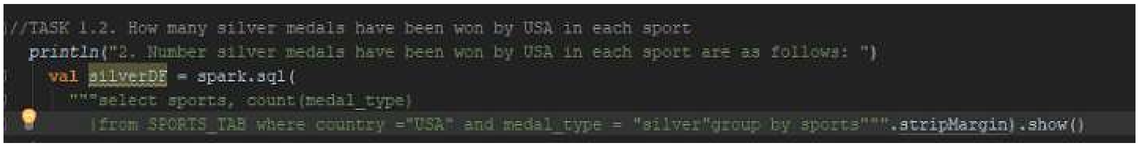


```
//Task 1.1. What are the total number of gold medal winners every year

println("The Total number of Gold medal winners every year are as follows: ")
val goldDF = spark.sql( sqlText = """SELECT year, count(medal_type) FROM SPORTS_TAB WHERE medal_type = "gold" group by year""").show()
```

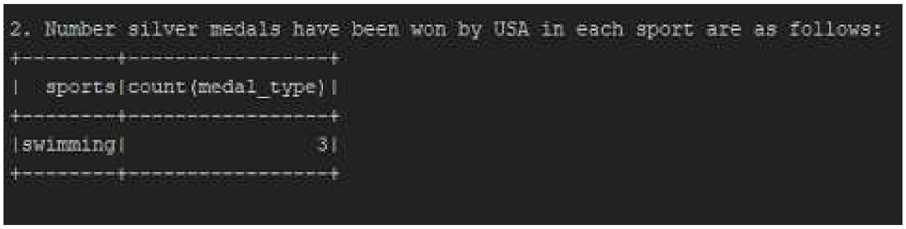


```
The Total number of Gold medal winners every year are as follows:
+----+-----------------+
|year|count(medal_type)|
+----+-----------------+
|2016|                2|
|2017|                1|
|2014|                3|
|2015|                3|
+----+-----------------+
```

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



```
//TASK 1.2. How many silver medals have been won by USA in each sport
  println("2. Number silver medals have been won by USA in each sport are as follows: ")
    val silverDF = spark.sql(
      """select sports, count(medal_type)
        |from SPORTS_TAB where country ="USA" and medal_type = "silver"group by sports""".stripMargin).show()
```

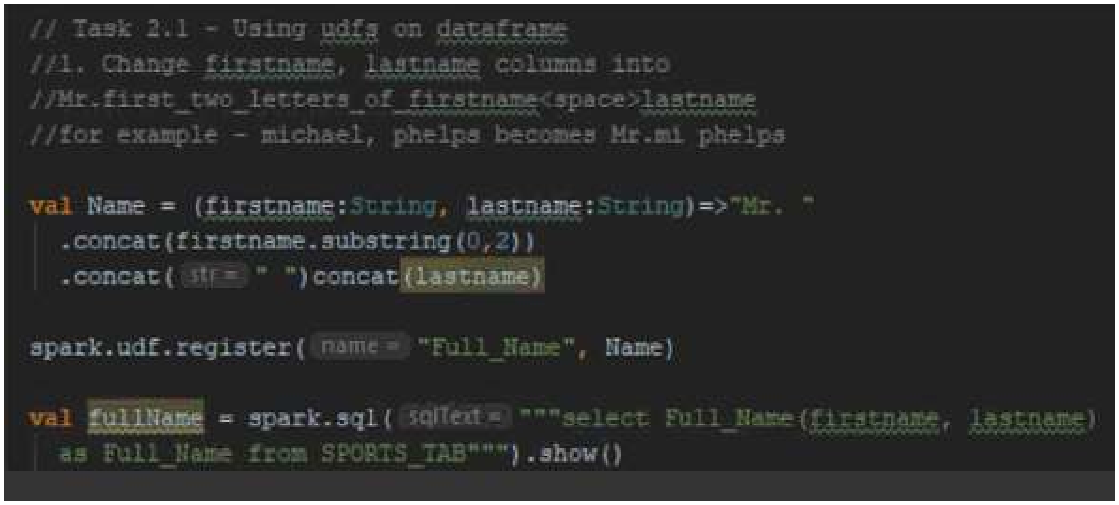


```
2. Number silver medals have been won by USA in each sport are as follows:
+--------+-----------------+
|  sports|count(medal_type)|
+--------+-----------------+
|swimming|                3|
+--------+-----------------+
```

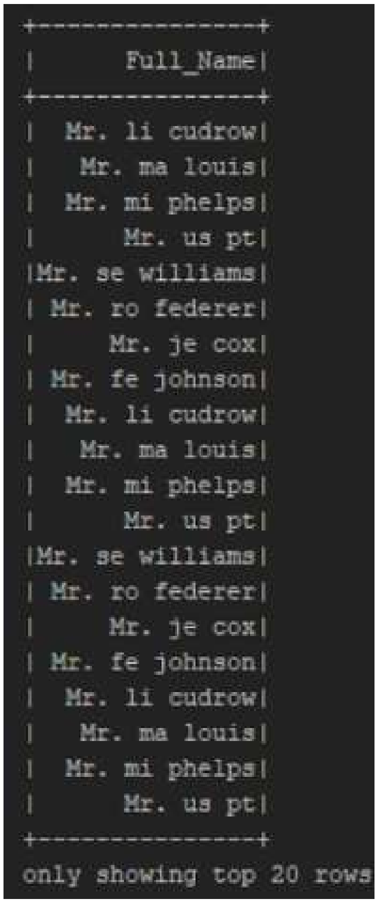
- Task 2

Using udfs on dataframe

Change firstname, lastname columns into Mr.first_two_letters_of_firstnamelastname for example - michael, phelps becomes Mr.mi phelps

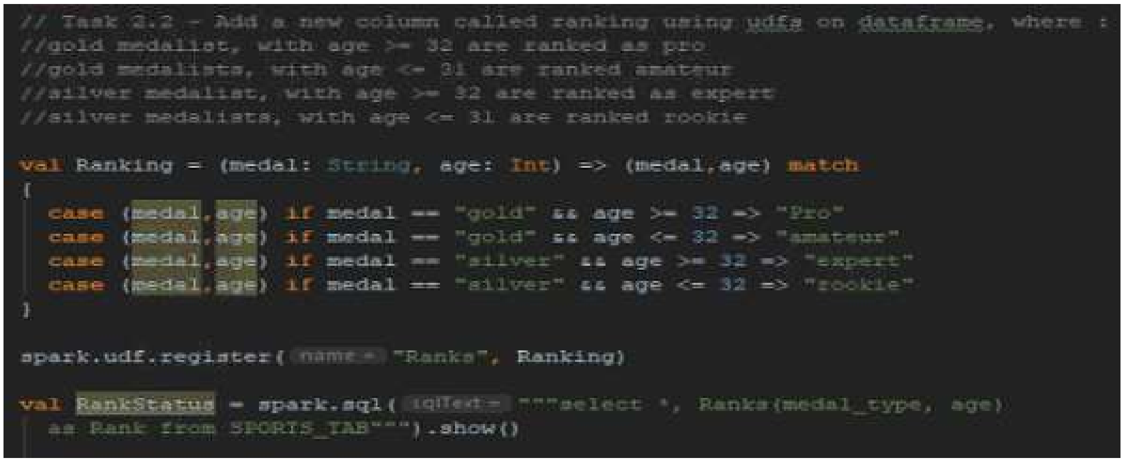


```
// Task 2.1 - Using udfs on dataframe
//1. Change firstname, lastname columns into
//Mr.first_two_letters_of_firstname<space>lastname
//for example - michael, phelps becomes Mr.mi phelps

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

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

val fullName = spark.sql( sqlText = """select Full_Name(firstname, lastname)
  as Full_Name from SPORTS_TAB""").show()
```

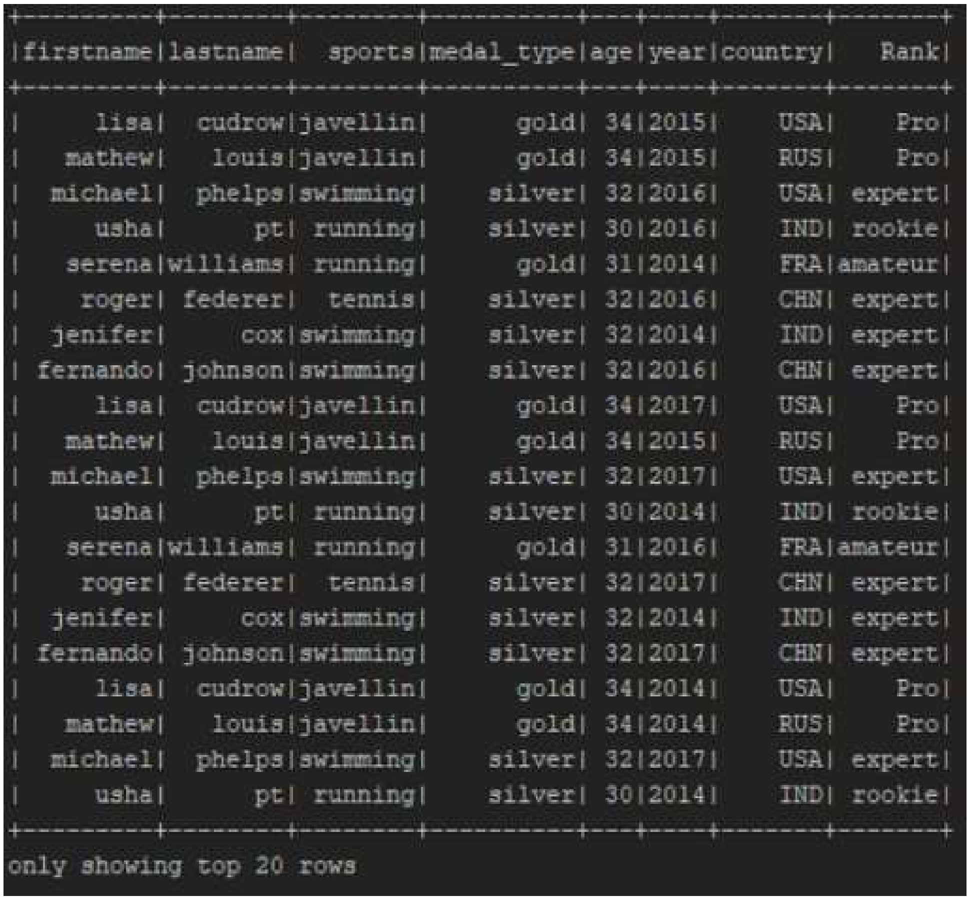


```
+---------------+
|      Full_Name|
+---------------+
|  Mr. li cudrow|
|   Mr. ma louis|
|  Mr. mi phelps|
|      Mr. us pt|
|Mr. se williams|
| Mr. ro federer|
|     Mr. je cox|
| Mr. fe johnson|
|  Mr. li cudrow|
|   Mr. ma louis|
|  Mr. mi phelps|
|      Mr. us pt|
|Mr. se williams|
| Mr. ro federer|
|     Mr. je cox|
| Mr. fe johnson|
|  Mr. li cudrow|
|   Mr. ma louis|
|  Mr. mi phelps|
|      Mr. us pt|
+---------------+
only showing top 20 rows
```

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



```
// 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

val Ranking = (medal: String, age: Int) => (medal,age) match
{
  case (medal,age) if medal == "gold" && age >= 32 => "Pro"
  case (medal,age) if medal == "gold" && age <= 32 => "amateur"
  case (medal,age) if medal == "silver" && age >= 32 => "expert"
  case (medal,age) if medal == "silver" && age <= 32 => "rookie"
}

spark.udf.register( name = "Ranks", Ranking)

val RankStatus = spark.sql( sqlText = """select *, Ranks(medal_type, age)
  as Rank from SPORTS_TAB""").show()
```



```
+---------+--------+--------+----------+---+----+-------+-------+
|firstname|lastname|  sports|medal_type|age|year|country|   Rank|
+---------+--------+--------+----------+---+----+-------+-------+
|     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
```