Assignment 21 SPARK SQL 2

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_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

SOURCE CODE :

```
package SQL

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

object Assignment21 {

  case class
Sports_data(firstname:String,lastname:String,sports:String,medal_type:String,age:Int,year:Lo
ng,country:String)

  def main(args: Array[String]): Unit = {

    println("hey scala")
```

```
//Let us create a spark session object

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("/Users/Vidya Sagar/Sports_data.txt")
    println("User details Data->>" + data.count())
    val header = data.first()

    val data1 = data.filter(row => row != header)

    println("Header removed from the data !")

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

    val usersdata = data1.map(x => x.split(",")).map(x => Sports_data(x(0), x(1), x(2), x(3),
x(4).toInt, x(5).toLong, x(6))).toDF

    usersdata.show()

    println("user Data Dataframe created !")

    usersdata.registerTempTable("USERS_DATA")

    val obj1 = spark.sql("select count(medal_type),year from USERS_DATA where
medal_type='gold' group by (year)")
    obj1.show()

    val obj2 = spark.sql("select count(medal_type),sports from USERS_DATA where
medal_type='silver' and country = 'USA' group by (sports) ")
    obj2.show()

    //val obj3 = spark.sql("select concat('Mr.', firstname,' ',lastname) AS
Full_Name,sports,medal_type,age,year,country from USERS_DATA")
    val obj3 = spark.sql("select concat('Mr.', substring(firstname,0,2),' ',lastname) AS
Full_Name,sports,medal_type,age,year,country from USERS_DATA")
    obj3.show()
```

```
  val first_and_last_name_concat = udf((first_name: String, last_name: String) =>
"Mr.".concat(first_name.substring(0, 2)).concat(" ").concat(last_name))

  val new_sports_data_sports_df = usersdata.withColumn("fullName",
    first_and_last_name_concat(usersdata("firstname"), usersdata("lastname")))

new_sports_data_sports_df.select("fullName","sports","medal_type","age","year","country")
    .show()

  val Rank = udf((medal_type: String, age: Int) => {
    if (medal_type == "gold" && age >= 32) "pro"
    else if (medal_type == "gold" && age <= 31) "amateur"
    else if (medal_type == "silver" && age >= 32) "expert"
    else if (medal_type == "silver" && age <= 32) "rookie" else " no-level" })
   usersdata.withColumn("ranking", Rank(usersdata("medal_type"),
    usersdata("age"))).show()
  }
}
```

OUT PUT SCREEN SHOTS : Below is the Data file

```
Spark Session Object created
User details Data->>25
Header removed from the data !
+---------+--------+--------+----------+---+----+-------+
|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|
+---------+--------+--------+----------+---+----+-------+
only showing top 20 rows
```

1. Total number of gold medal winners every year

```
select count(medal_type),year from USERS_DATA where medal_type='gold' group by (year)
```

```
user Data Dataframe created !
+-----------------+----+
|count(medal_type)|year|
+-----------------+----+
|                3|2014|
|                2|2016|
|                1|2017|
|                3|2015|
+-----------------+----+
```

2. Silver medals have been won by USA in each sport

```
select count(medal_type),sports from USERS_DATA where medal_type='silver' and country = 'USA' group by (sports)
```

```
+-----------------+--------+
|count(medal_type)|  sports|
+-----------------+--------+
|                3|swimming|
+-----------------+--------+
```

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 first_and_last_name_concat = udf((first_name: String, last_name: String) =>
"Mr.".concat(first_name.substring(0, 2)).concat(" ").concat(last_name))

val new_sports_data_sports_df = usersdata.withColumn("fullName",
  first_and_last_name_concat(usersdata("firstname"), usersdata("lastname")))
new_sports_data_sports_df.select("fullName","sports","medal_type","age","year","country")
  .show()
```

```
+--------------+--------+----------+---+----+-------+
|      fullName|  sports|medal_type|age|year|country|
+--------------+--------+----------+---+----+-------+
|  Mr.li cudrow|javellin|      gold| 34|2015|    USA|
|   Mr.ma louis|javellin|      gold| 34|2015|    RUS|
|  Mr.mi phelps|swimming|    silver| 32|2016|    USA|
|      Mr.us pt| running|    silver| 30|2016|    IND|
|Mr.se williams| running|      gold| 31|2014|    FRA|
| Mr.ro federer|  tennis|    silver| 32|2016|    CHN|
|     Mr.je cox|swimming|    silver| 32|2014|    IND|
| Mr.fe johnson|swimming|    silver| 32|2016|    CHN|
|  Mr.li cudrow|javellin|      gold| 34|2017|    USA|
|   Mr.ma louis|javellin|      gold| 34|2015|    RUS|
|  Mr.mi phelps|swimming|    silver| 32|2017|    USA|
|      Mr.us pt| running|    silver| 30|2014|    IND|
|Mr.se williams| running|      gold| 31|2016|    FRA|
| Mr.ro federer|  tennis|    silver| 32|2017|    CHN|
|     Mr.je cox|swimming|    silver| 32|2014|    IND|
| Mr.fe johnson|swimming|    silver| 32|2017|    CHN|
|  Mr.li cudrow|javellin|      gold| 34|2014|    USA|
|   Mr.ma louis|javellin|      gold| 34|2014|    RUS|
|  Mr.mi phelps|swimming|    silver| 32|2017|    USA|
|      Mr.us pt| running|    silver| 30|2014|    IND|
+--------------+--------+----------+---+----+-------+
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

```
val Rank = udf((medal_type: String, age: Int) => {
  if (medal_type == "gold" && age >= 32) "pro"
  else if (medal_type == "gold" && age <= 31) "amateur"
  else if (medal_type == "silver" && age >= 32) "expert"
  else if (medal_type == "silver" && age <= 32) "rookie" else " no-level" })
usersdata.withColumn("ranking", Rank(usersdata("medal_type"),
  usersdata("age"))).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


Process finished with exit code 0
```