Assignment-21

Task 1

Using spark-sql, Find:

1. **What are the total number of gold medal winners every year**

The scala code to the above question is shown in the below screenshot.

```
package Spark_SQL_2_Assign

import org.apache.spark.sql.SparkSession

object Question_1 {

  case class holidays(first_name:String,lastname:String,sports:String,medal_type:String,age:Int,year:Int, country: String)

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

    val spark = SparkSession
      .builder()
      .master( master = "local")
      .appName( name = "Spark Sql Assign 2")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")
    spark.sparkContext.setLogLevel("WARN")

    val Sports_data = spark.sparkContext.textFile( path = "D:\\barath\\Sports_data.txt");
    val header = Sports_data.first()
    val data_1 = Sports_data.filter(row => row != header)

    println("Header removed from the data !")

    import spark.implicits._

    val sports_data = data_1.map(x=>x.split( regex = ",")).map(x => holidays(x(0),x(1),x(2),x(3),x(4).toInt,x(5).toInt,x(6))).toDF()
    sports_data.registerTempTable( tableName = "sport_data")
    val data1 = spark.sql( sqlText = "select s.year, count(*) from sport_data s where s.medal_type= 'gold' group by s.year")
    data1.show()
  }
}
```

The output of total number of gold medal winners every year is shown in the below screenshot.

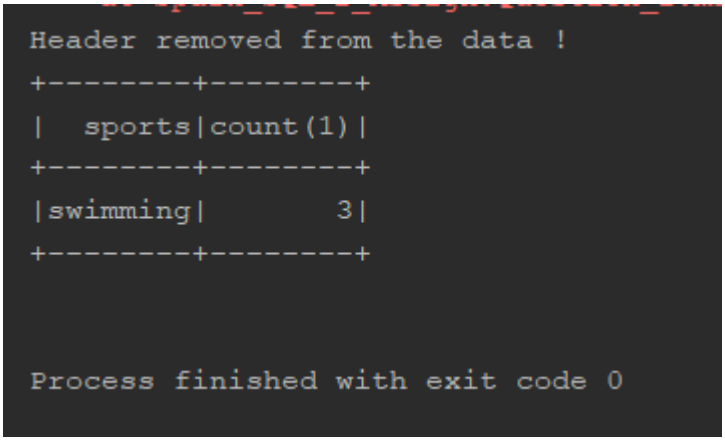
```
Header removed from the data !
+----+--------+
|year|count(1)|
+----+--------+
|2015|       3|
|2014|       3|
|2016|       2|
|2017|       1|
+----+--------+


Process finished with exit code 0
```

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

The scala code to the above question is shown in the below screenshot.

```
package Spark_SQL_2_Assign

import org.apache.spark.sql.SparkSession

object Question_2 {

  case class sportsdata(first_name:String,lastname:String,sports:String,medal_type:String,age:Int,year:Int, country: String)

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

    val spark = SparkSession
      .builder()
      .master( master = "local")
      .appName( name = "Spark Sql Assign 2")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")

    spark.sparkContext.setLogLevel("WARN")

    val Sports_data = spark.sparkContext.textFile( path = "D:\\barath\\Sports_data.txt");
    val header = Sports_data.first()
    val data_1 = Sports_data.filter(row => row != header)

    println("Header removed from the data !")

    import spark.implicits._

    val sports_data = data_1.map(x => x.split( regex = ",")).map(x => sportsdata(x(0), x(1), x(2), x(3), x(4).toInt, x(5).toInt, x(6))).toDF()

    sports_data.registerTempTable( tableName = "sport_data")

    val data1 = spark.sql( sqlText = "select s.sports, count(*) from sport_data s where s.medal_type= 'silver' and s.country = 'USA' group by s.sports")

    data1.show()
  }
}
```

The output of total number of silver medals won by USA in each sport is shown in the below screenshot.



```
Header removed from the data !
+--------+--------+
|  sports|count(1)|
+--------+--------+
|swimming|       3|
+--------+--------+


Process finished with exit code 0
```

Task 2

Using udfs on dataframe

1. Change firstname, lastname columns into

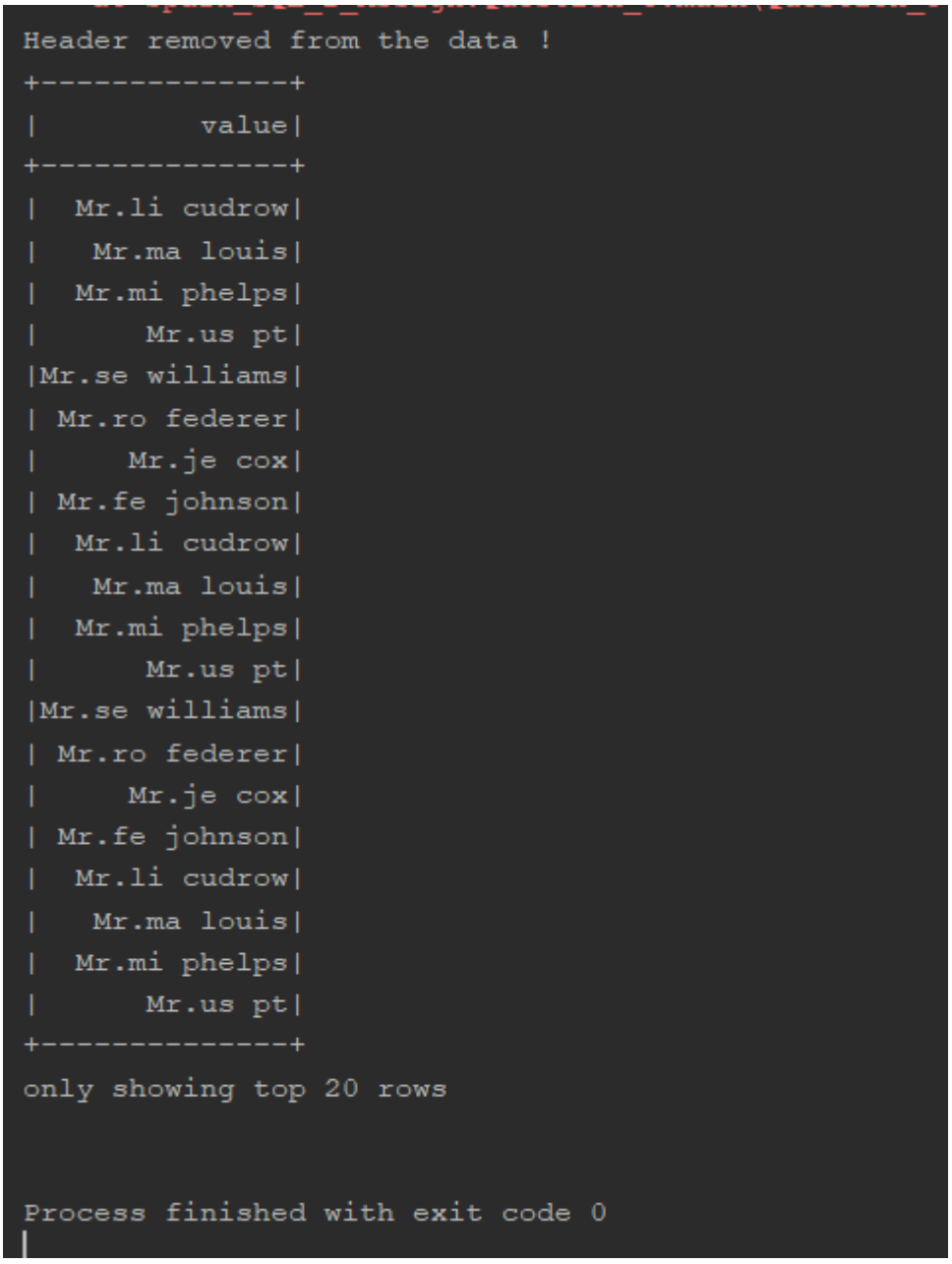
Mr.first_two_letters_of_firstname<space>lastname

for example - michael, phelps becomes Mr.mi phelps

The scala code to the above question using UDF is shown in the below screenshot.

```
package Spark_SQL_2_Assign

import ...

object Question_3 {

  case class sportsdata(first_name:String,lastname:String,sports:String,medal_type:String,age:Int,year:Int, country: String)

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

    val spark = SparkSession
      .builder()
      .master( master = "local")
      .appName( name = "Spark Sql assign 2 ")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")
    spark.sparkContext.setLogLevel("WARN")

    val Sports_data = spark.sparkContext.textFile( path = "D:\\barath\\Sports_data.txt");
    val header = Sports_data.first()
    val data_1 = Sports_data.filter(row => row != header)

    println("Header removed from the data !")

    import spark.implicits._

    val build = data_1.map(x=> x.split( regex = ",")).map(x => sportsdata(x(0),x(1),x(2),x(3),x(4).toInt,x(5).toInt,x(6))).toDF
    build.createOrReplaceTempView( viewName = "sportsTable")

    val build3 = spark.sql( sqlText = "select first_name,lastname from sportsTable")

    val udf = build3.map(x => ("Mr.".concat(x(0).toString.substring(0,2)).concat( str = " ").concat(x(1).toString)))

    udf.show()
  }
  }
```

The output to the above code is shown in the below screenshot.



```
Header removed from the data !
+--------------+
|         value|
+--------------+
|  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


Process finished with exit code 0
```

2. Add a new column called ranking using udfs on dataframe, where:

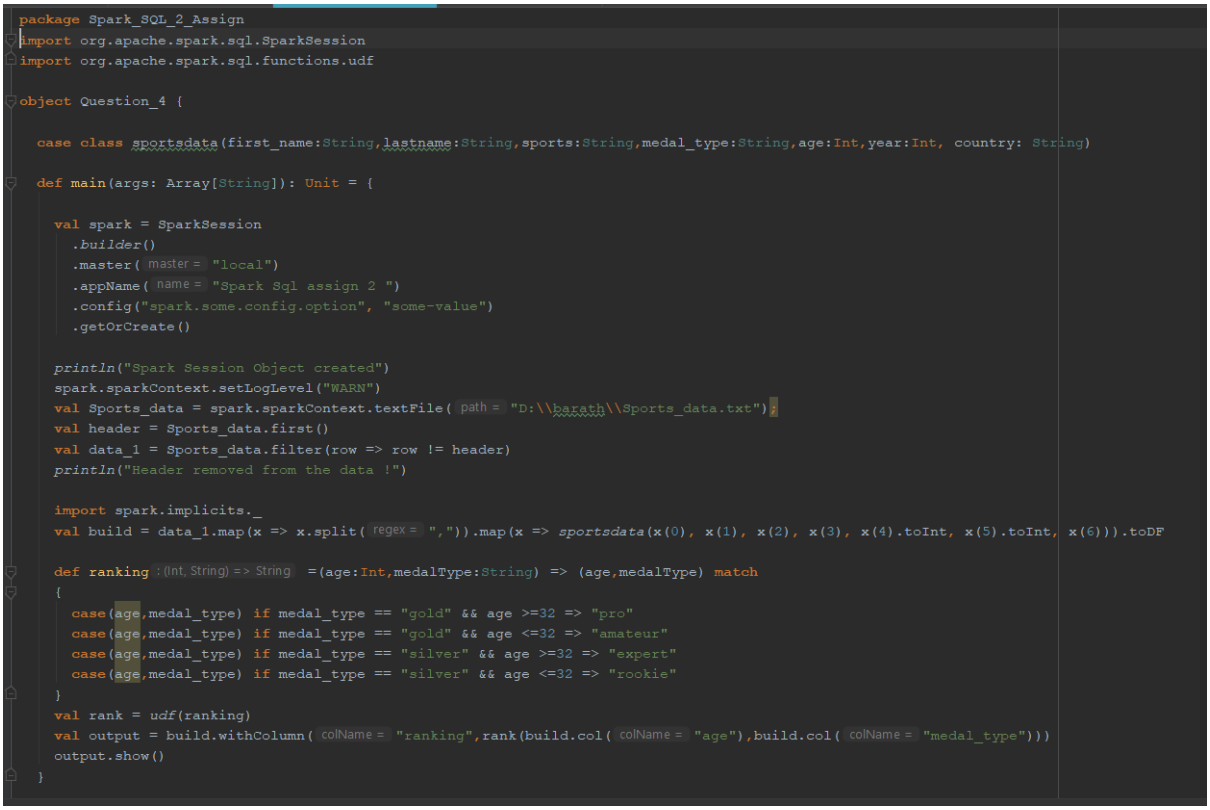
gold medalist, with age >= 32 are ranked as pro

gold medalists, with age <= 31 are ranked amateur

silver medalist, with age >= 32 are ranked as expert

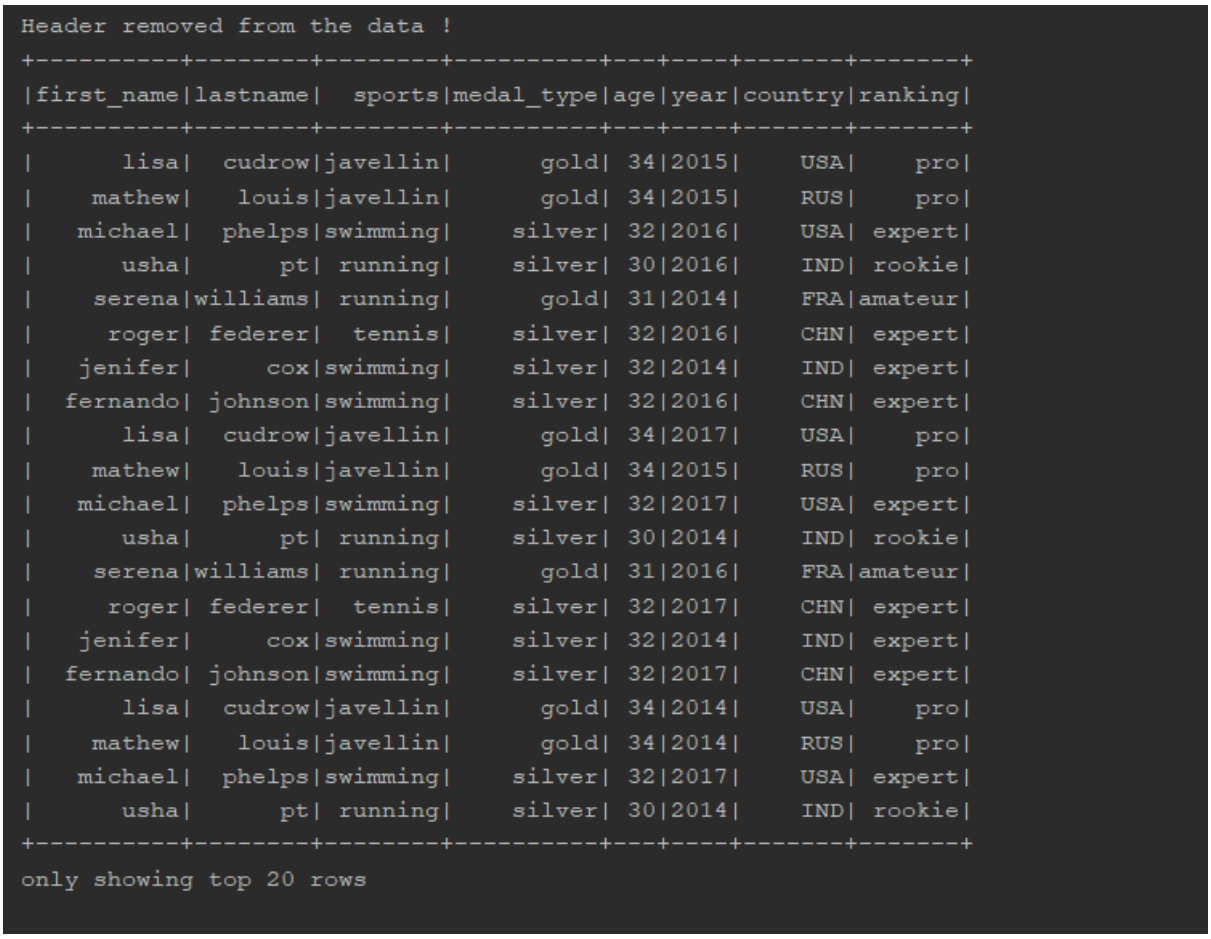
silver medalists, with age <= 31 are ranked rookie

The scala code to the above question is shown in the below screenshot.



```
package Spark_SQL_2_Assign
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions.udf

object Question_4 {

  case class sportsdata(first_name:String,lastname:String,sports:String,medal_type:String,age:Int,year:Int, country: String)

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

    val spark = SparkSession
      .builder()
      .master( master = "local")
      .appName( name = "Spark Sql assign 2 ")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")
    spark.sparkContext.setLogLevel("WARN")
    val Sports_data = spark.sparkContext.textFile( path = "D:\\barath\\Sports_data.txt");
    val header = Sports_data.first()
    val data_1 = Sports_data.filter(row => row != header)
    println("Header removed from the data !")

    import spark.implicits._
    val build = data_1.map(x => x.split( regex = ",")).map(x => sportsdata(x(0), x(1), x(2), x(3), x(4).toInt, x(5).toInt, x(6))).toDF

    def ranking : (Int, String) => String =(age:Int,medalType:String) => (age,medalType) match
    {
      case(age,medal_type) if medal_type == "gold" && age >=32 => "pro"
      case(age,medal_type) if medal_type == "gold" && age <=32 => "amateur"
      case(age,medal_type) if medal_type == "silver" && age >=32 => "expert"
      case(age,medal_type) if medal_type == "silver" && age <=32 => "rookie"
    }
    val rank = udf(ranking)
    val output = build.withColumn( colName = "ranking",rank(build.col( colName = "age"),build.col( colName = "medal_type")))
    output.show()
  }
```

The output to the above code is shown in the below screenshot.



```
Header removed from the data !
+----------+--------+--------+----------+---+----+-------+-------+
|first_name|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
```