Assignment 21.1
Spark SQL 2

Task 1

Using spark-sql, Find:

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

In below program, we have created **sports** case class for **sports data** file and then created Spark object.

Scala code :

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


object Assignment_21_Spark_SQL_2 {

  case class sports(firstname: 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("local")
      .appName("Spark SQL 2 Assignment")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")
```

Output :

Spark Session Object created

Then we have loaded data from **Sports data** csv file and then we have taken first (header) row from this file and filtered out header row from it.

Scala code :

```
    import spark.implicits._

    // Below statement will suppress all warnings
    spark.sparkContext.setLogLevel("WARN")

    val sportsFile = spark.sparkContext.textFile("C:\\AcadGild
Hadoop\\Assignments\\Sports_data.csv")

    println("Below we are removing header columns from the file")

    val header = sportsFile.first()

    val sportsFile2 = sportsFile.filter(x => x != header)
```

Assignment 21.1
Spark SQL 2

Output :

Below we are removing header columns from the file

```
------------------------------------------------------------------------------------------------------------------
```

Then we have converted **sportsFile2** RDD to DataFrame and then we have registered it as **sports** table.

Then we have used **sql** transformation to create sql query by using group by clause for year column from **sports** table and printed the result from this query.

Scala code :

```
    val sportsmap = sportsFile2.map(x => x.split(",")).map(x =>
sports(x(0),x(1),x(2).toString,x(3).toString,x(4).toInt,x(5).toInt,x(6).toString)).
toDF()

    sportsmap.createOrReplaceTempView("sports")

    println("sports table has been created from Sports Data file")

    println("Below is the total number of gold medal winners every year")

    val sportsSql = spark.sql("select year,count(*) no_of_gold_medal_winners from
sports where medal_type='gold' group by year order by year")

    sportsSql.show()
```

Output :

```
sports table has been created from Sports Data file
Below is the total number of gold medal winners every year
+----+------------------------+
|year|no_of_gold_medal_winners|
+----+------------------------+
|2014|                       3|
|2015|                       3|
|2016|                       2|
|2017|                       1|
+----+------------------------+
```

Assignment 21.1
Spark SQL 2

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

Then we have used **sql** transformation to create sql query by using group by clause for sports column from **sports** table and printed the result from this query.

Scala code :

```
println("Below is the number of silver medals have been won by USA in each sport")

val sportsSql2 = spark.sql("select sports,count(*)
no_of_silver_medal_winners_in_USA from sports where medal_type='silver' and country
= 'USA' group by sports")

sportsSql2.show()
```

Output :

```
Below is the number of silver medals have been won by USA in each sport
+--------+---------------------------------+
|  sports|no_of_silver_medal_winners_in_USA|
+--------+---------------------------------+
|swimming|                                3|
+--------+---------------------------------+
```

Assignment 21.1
Spark SQL 2

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

Below we have created **Name** function and registered it as **Full_Name** UDF function. Then we have used this UDF in sql query to produce the result and we have displayed the result.

Scala code :

```
println("Using udfs on dataframe change firstname, lastname columns into
Mr.first_two_letters_of_firstname<space>lastname")

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

val Full_Name = udf(Name(_:String,_:String):String)

spark.udf.register("Full_Name", Name(_:String,_:String):String)

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

fname.show()
```

Assignment 21.1
Spark SQL 2

<u>Output :</u>

Using udfs on dataframe change firstname, lastname columns into Mr.first_two_letters_of_firstname<space>lastname

```
+--------------+
|     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
```

Assignment 21.1
Spark SQL 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

Below we have created **Rank** function and registered it as **ranking** UDF function. Then we have used this UDF to add this as extra column **ranking** at the end and we have displayed the result.

Scala code :

```
println("Below we have added a new column called ranking using udfs on dataframe")

def Rank(medal : String, age : Int ): String = (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"
}

val ranking = udf(Rank(_:String,_:Int):String)

spark.udf.register("ranking",Rank(_:String,_:Int):String)

val RankingRDD = sportsmap.withColumn("ranking",
  ranking(sportsmap.col("medal_type"),sportsmap.col("age")))

RankingRDD.show()
```

Assignment 21.1
Spark SQL 2

Output :

Below we have added a new column called ranking using udfs on dataframe

```
+---------+--------+--------+----------+---+----+-------+-------+
|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
```

Assignment 21.1
Spark SQL 2

Complete Scala code :

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


object Assignment_21_Spark_SQL_2 {

  case class sports(firstname: 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("local")
      .appName("Spark SQL 2 Assignment")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    println("Spark Session Object created")

    import spark.implicits._

    // Below statement will suppress all warnings
    spark.sparkContext.setLogLevel("WARN")

    val sportsFile = spark.sparkContext.textFile("C:\\AcadGild
Hadoop\\Assignments\\Sports_data.csv")

    println("Below we are removing header columns from the file")

    val header = sportsFile.first()

    val sportsFile2 = sportsFile.filter(x => x != header)

    val sportsmap = sportsFile2.map(x => x.split(",")).map(x =>
sports(x(0),x(1),x(2).toString,x(3).toString,x(4).toInt,x(5).toInt,x(6).toString)).
toDF()

    sportsmap.createOrReplaceTempView("sports")

    println("sports table has been created from Sports Data file")

    println("Below is the total number of gold medal winners every year")

    val sportsSql = spark.sql("select year,count(*) no_of_gold_medal_winners from
sports where medal_type='gold' group by year order by year")

    sportsSql.show()



    println("Below is the number of silver medals have been won by USA in each
sport")

    val sportsSql2 = spark.sql("select sports,count(*)
no_of_silver_medal_winners_in_USA from sports where medal_type='silver' and country
= 'USA' group by sports")

    sportsSql2.show()


    println("Using udfs on dataframe change firstname, lastname columns into
Mr.first_two_letters_of_firstname<space>lastname")
```

Assignment 21.1
Spark SQL 2

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

    val Full_Name = udf(Name(_:String,_:String):String)

    spark.udf.register("Full_Name", Name(_:String,_:String):String)

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

    fname.show()

    //////////////////////////

    println("Below we have added a new column called ranking using udfs on
dataframe")

    def Rank(medal : String, age : Int ): String = (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"
    }

    val ranking = udf(Rank(_:String,_:Int):String)

    spark.udf.register("ranking",Rank(_:String,_:Int):String)

    val RankingRDD = sportsmap.withColumn("ranking",
      ranking(sportsmap.col("medal_type"),sportsmap.col("age")))

    RankingRDD.show()

  }
}
```

Assignment 21.1
Spark SQL 2

Complete Output :

```
Spark Session Object created
Below we are removing header columns from the file
sports table has been created from Sports Data file
Below is the total number of gold medal winners every year
+----+------------------------+
|year|no_of_gold_medal_winners|
+----+------------------------+
|2014|                       3|
|2015|                       3|
|2016|                       2|
|2017|                       1|
+----+------------------------+

Below is the number of silver medals have been won by USA in each sport
+--------+---------------------------------+
|  sports|no_of_silver_medal_winners_in_USA|
+--------+---------------------------------+
|swimming|                                3|
+--------+---------------------------------+

Using udfs on dataframe change firstname, lastname columns into
Mr.first_two_letters_of_firstname<space>lastname
+--------------+
|     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
```

Assignment 21.1
Spark SQL 2

Below we have added a new column called ranking using udfs on dataframe

```
+---------+--------+--------+----------+---+----+-------+-------+
|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
```