

Session 21:

SPARK SQL 2


Assignment 1

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

Dataset :

 Sports_data - Notepad

File Edit Format View Help

```
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
```

Code Used:

```
package Asg_21
import org.apache.spark.sql.{Row, SparkSession}
import org.apache.spark.sql.types.{IntegerType, StringType, StructField,
StructType}
//import org.apache.spark.sql.types._
import org.apache.spark.sql.SparkSession
```

```

object Task1 {

  def main(args: Array[String]): Unit = {
    println("Assignment Number 21 !!!")

    // Use new SparkSession interface in Spark
    val spark = SparkSession
      .builder()
      .master("local[*]")
      .appName("Assignment 20 task no 7 ")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()
    println("Spark Session Object created")
    // import spark.implicits._

    val SportsData =
      spark.sparkContext.textFile("/C:/Users/admin/Desktop/Assignment_to_be
submitted/asg_21/Sports_data.txt")
    val schemaString =

"firstname:string,lastname:string,sports:string,medal_type:string,age:string,year:s
tring,country:string"
    val schema = StructType(schemaString.split(",").map(x =>
      StructField(x.split(":")(0),if(x.split(":")(1).equals("string"))StringType
else IntegerType, true)))
    val rowRDD = SportsData.map(_.split(",")).map(r=> Row(r(0), r(1), r(2),
r(3), r(4), r(5), r(6)))
    val SportsDataDF = spark.createDataFrame(rowRDD, schema)
    SportsDataDF.createOrReplaceTempView("SportsData")

    //1.What are the total number of gold medal winners every year
    val resultDF = spark.sql("SELECT year,COUNT (*) FROM SportsData WHERE
medal_type = 'gold' GROUP BY year")
    resultDF.show()
    println("Total Number Of Gold Medal Winners Every Year ")

    //2. How many silver medals have been won by USA in each sport
    val result2DF = spark.sql("SELECT sports, COUNT (*) FROM SportsData WHERE
medal_type = 'silver' and country = 'USA' GROUP BY sports")
    result2DF.show()
    println("Silver Medals Won by USA in Each Sport ")
  }
}

```

Output:

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

```

18/09/21 11:24:15 INFO CodeGenerator: Code generated in 16.58961 ms
+----+-----+
|year|count(1)|
+----+-----+
|2016|      2|
|2017|      1|
|2014|      3|
|2015|      3|
+----+-----+

Total Number Of Gold Medal Winners Every Year
18/09/21 11:24:15 INFO SparkContext: Invoking stop() from shutdown hook
18/09/21 11:24:15 INFO SparkUI: Stopped Spark web UI at http://192.168.100.4:4040
18/09/21 11:24:15 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
18/09/21 11:24:15 INFO MemoryStore: MemoryStore cleared

```

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

```
18/09/21 11:26:51 INFO TaskSetManager: Finished task 31.0 in stage 19.0 (TID 360) in 250 ms on localhost (executor driver) (73/75)
18/09/21 11:26:51 INFO TaskSetManager: Finished task 73.0 in stage 19.0 (TID 401) in 30 ms on localhost (executor driver) (74/75)
+-----+
| sports|count(1)|
+-----+
|swimming|      3|
+-----+

Silver Medals Won by USA in Each Sport
18/09/21 11:26:51 INFO Executor: Finished task 51.0 in stage 19.0 (TID 403). 2607 bytes result sent to driver
18/09/21 11:26:51 INFO TaskSetManager: Finished task 51.0 in stage 19.0 (TID 403) in 10 ms on localhost (executor driver) (75/75)
18/09/21 11:26:51 INFO TaskSchedulerImpl: Removed TaskSet 19.0, whose tasks have all completed, from pool
```

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

Code Used :

```
package Asg_21

import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.{Row, SparkSession}
import org.apache.spark.sql.types.{IntegerType, NumericType, StringType, StructField, StructType}
import org.apache.spark.sql.functions.udf
import org.apache.spark.sql.functions._
import org.apache.spark.sql.expressions.UserDefinedFunction
//import org.apache.spark.sql.DataFrame
object Task2 {
  def main(args: Array[String]): Unit = {
    println("Assignment Number 21 !!!")
    // Use new SparkSession interface in Spark
    val spark = SparkSession
      .builder()
      .master("local[*]")
      .appName("Assignment 20 task no 7 ")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()
    println("Spark Session Object created")
    // import spark.implicits._

    //Using udfs on dataframe

    val SportsData =
      spark.sparkContext.textFile("/C:/Users/admin/Desktop/Assignment_to_be
submitted/asg_21/Sports_data.txt")
    val schemaString =
      "firstname:string,lastname:string,sports:string,medal_type:string,age:string,year:s
```

```

tring, country: string"
    val schema = StructType(schemaString.split(",").map(x =>
    StructField(x.split(":")(0), if (x.split(":")(1).equals("string")) StringType else
    IntegerType, true)))
    val rowRDD = SportsData.map(_._split(",")).map(r => Row(r(0), r(1), r(2), r(3),
    r(4), r(5), r(6)))
    val SportsDataDF = spark.createDataFrame(rowRDD, schema)
    SportsDataDF.registerTempTable("Sports_Data")
    val fname = spark.sql("SELECT * FROM Sports_Data")
    fname.show()

//Using udfs on dataframe
//1. Change firstname, lastname columns into
//Mr.first_two_letters_of_firstname<space>lastname
    def ChangeName = udf((firstname: String, lastname: String) => {"Mr." + firstname + "
    " + lastname})

    val SportsChangeName =
    SportsDataDF.withColumn("name", ChangeName(SportsDataDF("firstname").substr(1, 2), SportsDataDF("lastname")))
    .drop(SportsDataDF("firstname")).drop(SportsDataDF("lastname"))
    SportsChangeName.show()

//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
    def AddRanking = udf((medal_type: String, age: Int) => (medal_type, age) match
    {
        case (medal_type, age) if medal_type == "gold" && age >= 32 => "Pro"
        case (medal_type, age) if medal_type == "gold" && age <= 31 => "amateur"
        case (medal_type, age) if medal_type == "silver" && age >= 32 => "expert"
        case (medal_type, age) if medal_type == "silver" && age <= 31 => "rookie"
    })
    val SportsDataFinal
    = SportsDataDF.withColumn("ranking", AddRanking(SportsDataDF("medal_type"), SportsDataDF("age")))
    SportsDataFinal.show()
}
}

```

Output :

1. Change firstname, lastname columns into
Mr.first_two_letters_of_firstname<space>lastname

```

+-----+-----+-----+-----+-----+-----+
| firstname|lastname| sports|medal_type|age|year|country|
+-----+-----+-----+-----+-----+-----+
|firstname|lastname| sports|medal_type|age|year|country|
| lisa| cudrow|javelin| gold| 34|2015| USA|
| mathew| louis|javelin| 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|javelin| gold| 34|2017| USA|
| mathew| louis|javelin| 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|javelin| gold| 34|2014| USA|
| mathew| louis|javelin| gold| 34|2014| RUS|
| michael| phelps|swimming| silver| 32|2017| USA|
+-----+-----+-----+-----+-----+-----+
only showing top 20 rows

```

```

18/09/22 00:31:31 INFO CodeGenerator: Code generated in 46.27325 ms
18/09/22 00:31:31 INFO SparkContext: Starting job: show at Task2.scala:40

```

```

18/09/22 00:31:31 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
18/09/22 00:31:31 INFO CodeGenerator: Code generated in 44.897122 ms
+-----+-----+-----+-----+-----+-----+
| sports|medal_type|age|year|country| name|
+-----+-----+-----+-----+-----+-----+
| sports|medal_type|age|year|country|Mr.f lastname|
|javelin| gold| 34|2015| USA| Mr.li cudrow|
|javelin| gold| 34|2015| RUS| Mr.ma louis|
|swimming| silver| 32|2016| USA| Mr.mi phelps|
| running| silver| 30|2016| IND| Mr.us pt|
| running| gold| 31|2014| FRA|Mr.se williams|
| tennis| silver| 32|2016| CHN| Mr.ro federer|
|swimming| silver| 32|2014| IND| Mr.je cox|
|swimming| silver| 32|2016| CHN| Mr.fe johnson|
|javelin| gold| 34|2017| USA| Mr.li cudrow|
|javelin| gold| 34|2015| RUS| Mr.ma louis|
|swimming| silver| 32|2017| USA| Mr.mi phelps|
| running| silver| 30|2014| IND| Mr.us pt|
| running| gold| 31|2016| FRA|Mr.se williams|
| tennis| silver| 32|2017| CHN| Mr.ro federer|
|swimming| silver| 32|2014| IND| Mr.je cox|
|swimming| silver| 32|2017| CHN| Mr.fe johnson|
|javelin| gold| 34|2014| USA| Mr.li cudrow|
|javelin| gold| 34|2014| RUS| Mr.ma louis|
|swimming| silver| 32|2017| USA| Mr.mi phelps|
+-----+-----+-----+-----+-----+-----+
only showing top 20 rows
|
18/09/22 00:31:32 INFO CodeGenerator: Code generated in 34.815973 ms

```

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

18/09/22 00:31:32 INFO DAGScheduler: Job 3 finished. Show at 183x2.8618.00, took 0.000735 s
18/09/22 00:31:32 INFO CodeGenerator: Code generated in 27.488622 ms

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

only showing top 20 rows

18/09/22 00:31:32 INFO SparkContext: Invoking stop() from shutdown hook

18/09/22 00:31:32 INFO SparkUI: Stopped Spark web UI at <http://192.168.100.4:4040>