

Assignment Day 21

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

Ans:

Note: Program files are properly documented for a detailed description of each instruction used within the program.

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

ScreenShot:

The screenshot displays a Spark IDE interface. On the left, a project tree shows the file structure. The main editor area contains Scala code that reads a CSV file, converts it to a DataFrame, and creates an SQL view named 'sport'. The code then executes a query to find the total number of gold medal winners per year. The output of the query is shown in a console window at the bottom.

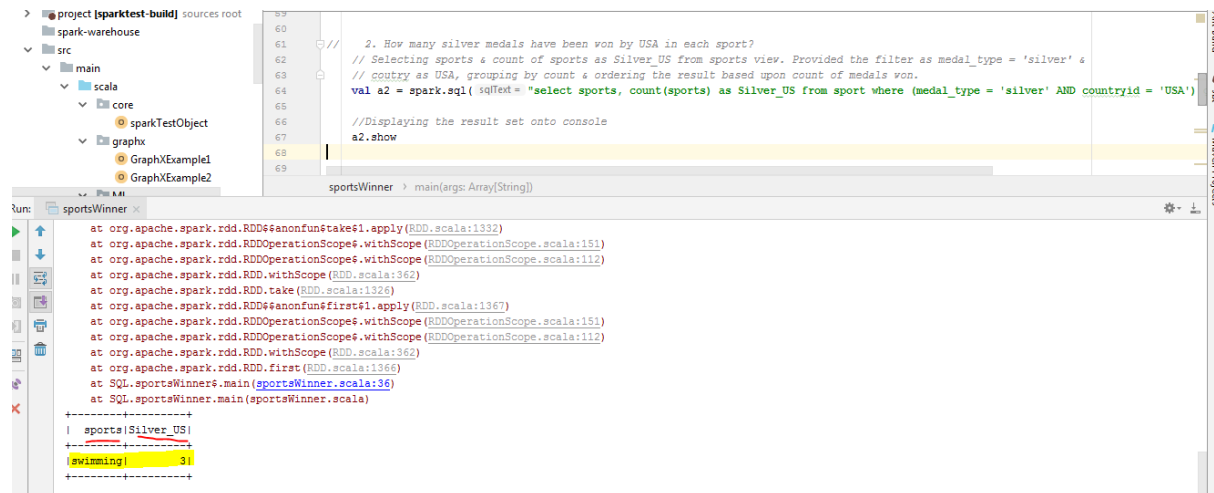
```
41 //Converting data into DataFrame splitting on comma character (,)
42 // trim is used to eliminate leading/trailing whitespace & converting values into respective data types
43 val sportsData = data.map(_.split(",")).map(x => Sports(x(0),x(1),x(2),x(3),x(4).trim.toInt,x(5).trim.toLong,x(6)))
44 .toDF()
45
46 //sportsData.show()
47
48 //Converting the above created schema into an SQL view named sport
49 sportsData.createOrReplaceTempView("sport")
50
51 // 1. What are the total number of gold medal winners every year?
52 // Selecting year & counting the occurrence of each year by filtering medal_type condition as gold.
53 // grouping by year & ordering the result based upon the year.
54 val a1 = spark.sql("Select year, count(year) as Gold_Medal_count from sport where medal_type = 'gold' group by year order by year")
55
56 //Displaying the result set onto console
57 a1.show
```

The console output shows the following table:

year	Gold_Medal_count
2014	3
2015	3
2016	2
2017	1

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

ScreenShot:



The screenshot shows an IDE with a project named 'sparktest-build' and a file named 'sportsWinner.scala'. The code defines a Spark SQL query to count silver medals won by the USA in each sport. The query is as follows:

```
2. How many silver medals have been won by USA in each sport?
// Selecting sports & count of sports as Silver_US from sports view. Provided the filter as medal_type = 'silver' &
// country as USA, grouping by count & ordering the result based upon count of medals won.
val a2 = spark.sql(s"select sports, count(sports) as Silver_US from sport where (medal_type = 'silver' AND countryid = 'USA')")
//Displaying the result set onto console
a2.show
```

The output of the query is displayed in the console, showing the following results:

```
at org.apache.spark.rdd.RDD$$anonfun$take$1.apply(RDD.scala:1332)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
at org.apache.spark.rdd.RDD.withScope(RDD.scala:362)
at org.apache.spark.rdd.RDD.take(RDD.scala:1326)
at org.apache.spark.rdd.RDD$$anonfun$first$1.apply(RDD.scala:1367)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
at org.apache.spark.rdd.RDD.withScope(RDD.scala:362)
at org.apache.spark.rdd.RDD.first(RDD.scala:1365)
at org.apache.spark.rdd.RDD.first(RDD.scala:1365)
at SQL.sportsWinner$.main(sportsWinner.scala:36)
at SQL.sportsWinner$.main(sportsWinner.scala:36)

+-----+
| sports|Silver_US|
+-----+
|swimming|31|
+-----+
```

Task 2:

Using UDFS on DataFrame

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

Ans:

Note: Program files are properly documented for a detailed description of each instruction used within the program.

ScreenShot:

```
100 //withColumn appends a new column in newly created dataframe i.e. df
101 here call to udf is made by passing two parameters i.e. firstname and lastname
102 and new name i.e. "name" is given to second parameter which is returned by udf
103 */
104 val df= sportsData.withColumn( colName= "name",udfChangeColumns($"firstname",$"lastname"))
105
106 //Two columns (firstname & lastname) of df are dropped and new df1 contains only undropped fields and corresponding data
107 val df1 = df.drop( colNames = "firstname","lastname")
108
109 //To interchange the position of columns, select api is used with df
110 val sol_df = df.select( col= "name", cols= "sports","medal_type","age","year","countryid")
111
112 //Displaying the result set onto console
113 sol_df.show
114
```

sportsWinner > main(args: Array[String])

name	sports	medal_type	age	year	countryid
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

2. Add a new column called ranking using UDFs on DataFrame, where :

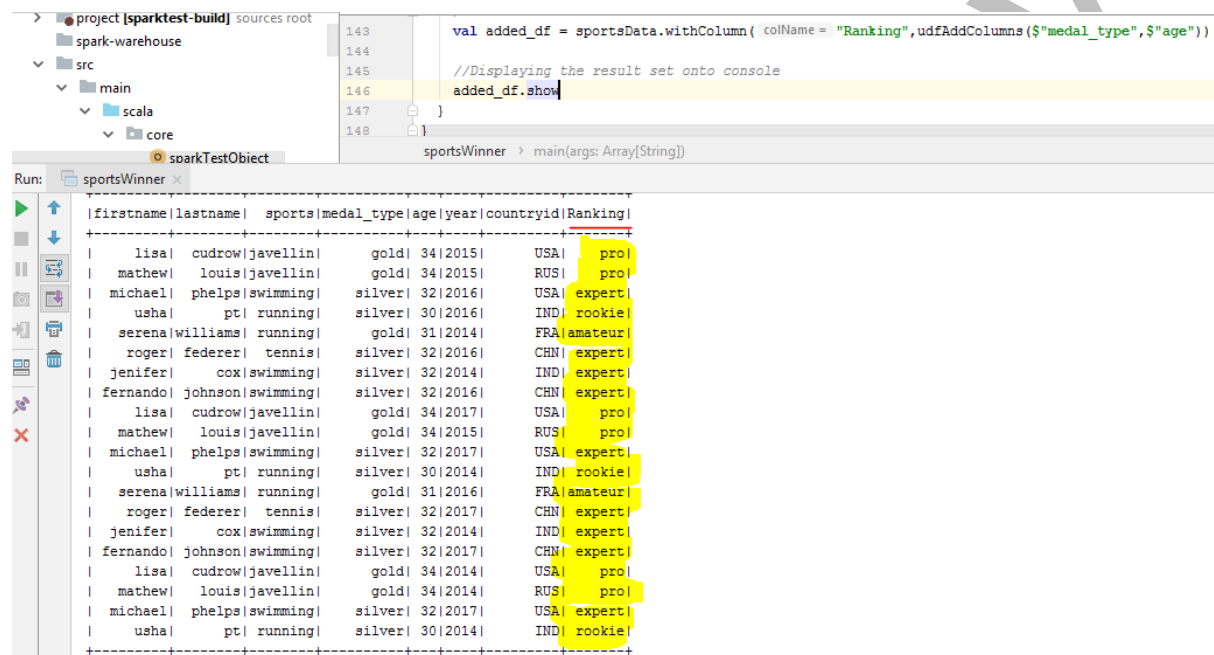
gold medalist, with age ≥ 32 are ranked as pro

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

ScreenShot:



The screenshot shows a Spark IDE with a Scala file named `sportsWinner.scala`. The code defines a UDF `udfAddColumns` and applies it to a DataFrame `sportsData` to create a new column `Ranking`. The output of `added_df.show()` is displayed in the console.

```
143 val added_df = sportsData.withColumn( colName = "Ranking", udfAddColumns($"medal_type", $"age"))
144
145 //Displaying the result set onto console
146 added_df.show()
147 }
148
```

The console output shows the following DataFrame:

firstname	lastname	sports	medal_type	age	year	countryid	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

End
