CS5590 BigData Programming - Lab Assignment 3

Team Id: 3

Member 1: Raju Nekadi
Class Id: 7

Member 2: Sushma Manne
Class Id: 3

Raju's GitHub Link:
https://github.com/rnekadi/CSEE5590_BIGDATA_PROGAMMING_Fall2018/tree/master/Lab3

Sushma's GitHub Link:
https://github.com/sushmamanne/CSEE5590_BIGDATA_PROGRAMMING_FALL2018

Video Link:
https://youtu.be/zLlH6hD7B7c

Introduction:

In Lab3, we will be using Apache Spark, Data frames and Spark SQL examples on various datasets.

Objective:

Facebook Common Friend Finding using Apache Spark.

Approaches:

Let us take basic friend list from Facebook given dataset and do our computation to find common friend.

0 1 3 4 5
1 0 2 4
2 1 3 4
3 0 2
4 0 1 2 5
5 0 4

Here 0 have friends 1 3 4 5 and 1 have friend 0 2 4 so the output of this program should be Common friend between 0, 1 is 4. Similarly, we will compute Common friend for other users given in example.

We will have map and reducer phase in our program to find the final Common friends.
Map function take input as 0 1 3 4 5 and generate pair of friend list by key and output along entire friend list.

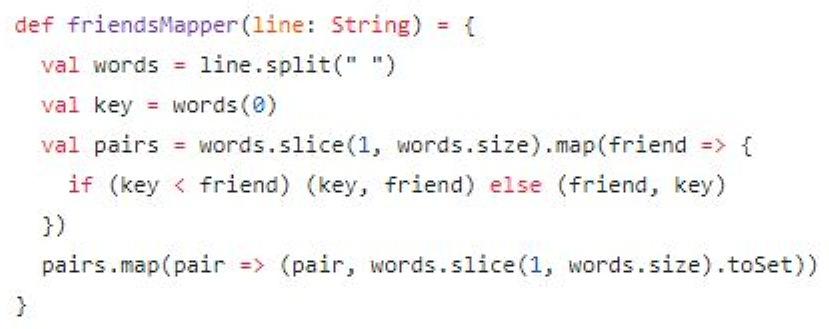
Reducer Function use group by key

Datasets:

https://github.com/rnekadi/CSEE5590_BIGDATA_PROGAMMING_Fall2018/blob/master/Lab 3/data/facebook_combined.txt

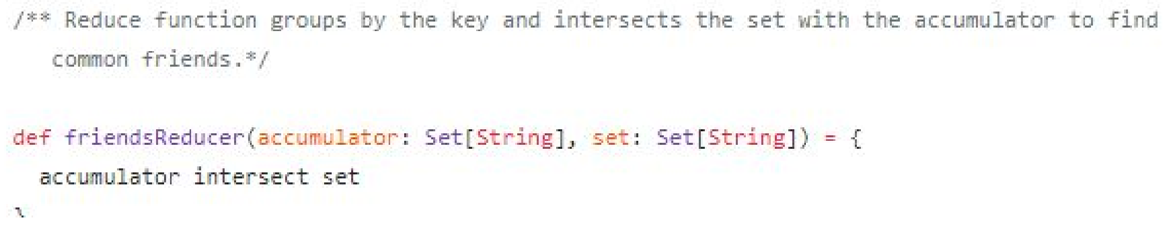
Workflow:

Here the friendMapper Function that perform mapping.



```
def friendsMapper(line: String) = {
  val words = line.split(" ")
  val key = words(0)
  val pairs = words.slice(1, words.size).map(friend => {
    if (key < friend) (key, friend) else (friend, key)
  })
  pairs.map(pair => (pair, words.slice(1, words.size).toSet))
}
```

Here is the friendReducer Function that perform reducing.



```
/** Reduce function groups by the key and intersects the set with the accumulator to find
   common friends.*/

def friendsReducer(accumulator: Set[String], set: Set[String]) = {
  accumulator intersect set
}
```

Both the function called in Spark flatmap, reducebykey along with filter and sortby transformations.

```
val file = sc.textFile("/Users/sai/Documents/GitHub/CSEE5590_BIGDATA_PROGAMMING_Fall2018/Lab3" +
  "/data/facebook_combined.txt")

val results = file.flatMap(friendsMapper)
  .reduceByKey(friendsReducer)
  .filter(!_._2.isEmpty)
  .sortByKey()

results.collect.foreach(line => {
  println(s"${line._1} ${line._2.mkString(" ")}")})

results.coalesce(1).saveAsTextFile("MutualFriends")
```

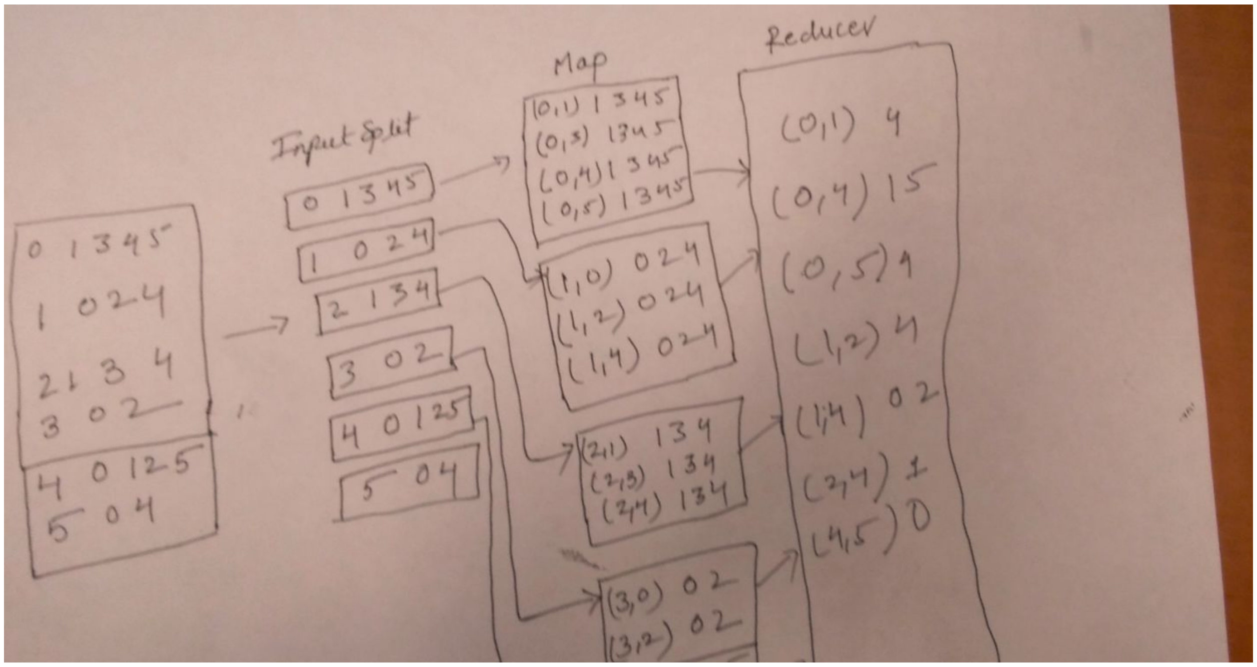
Finally, at last we are saving our result using Coalesce Action transformation in text file.

Evaluation:

Apache Sparks requires very less number of line as compare to traditional Map Reduce program and is very fast due to its framework.

Conclusion:

Spark is faster and better than traditional Map Reduce program.



Objective:

The goal of this part is to use Spark RDD, DataFrames and Spark Sql concepts.

Approaches:

To work on this problem we will take FIFA World Cup dataset and perform various step as mentioned below.

1. We will create the Data Frame from all 3 datasets for our analysis.
2. Using the all 3 dataframes we will be creating 3 temporary view on which we will be running Spark Sql.
3. Then we will perform 10 Apache Spark Sql operation and get the meaningful results.
4. For the last part we will creating the RDDs ,Dataframes and Apache Spark Sql perform various queries to see difference in result.

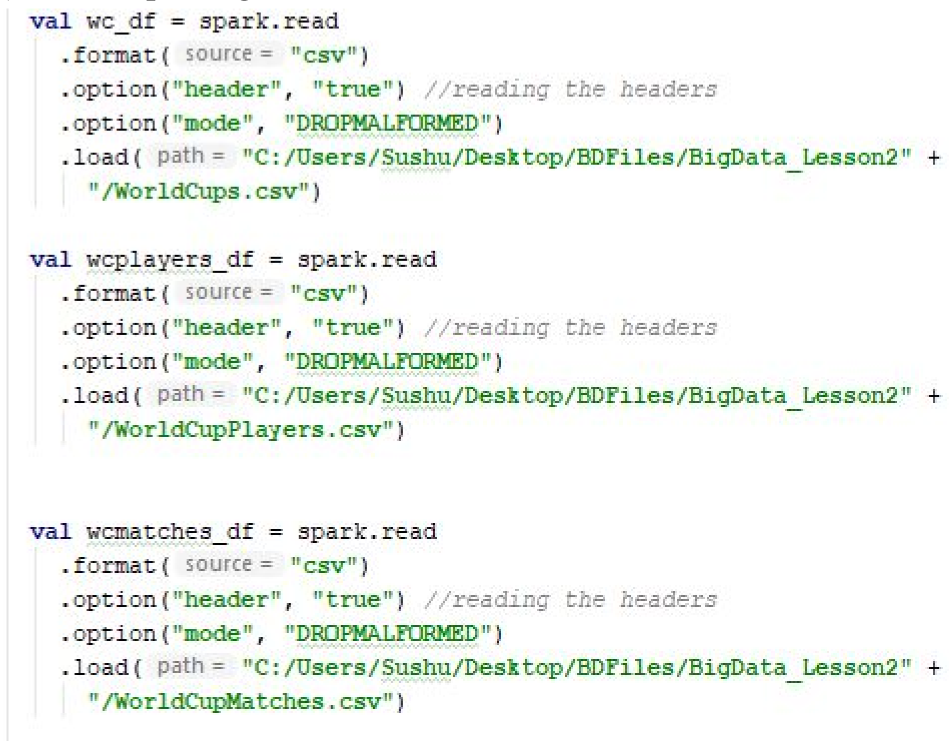
Datasets:

https://github.com/rnekadi/CSEE5590_BIGDATA_PROGAMMING_Fall2018/blob/master/Lab3/data/WorldCups.csv

Workflow:

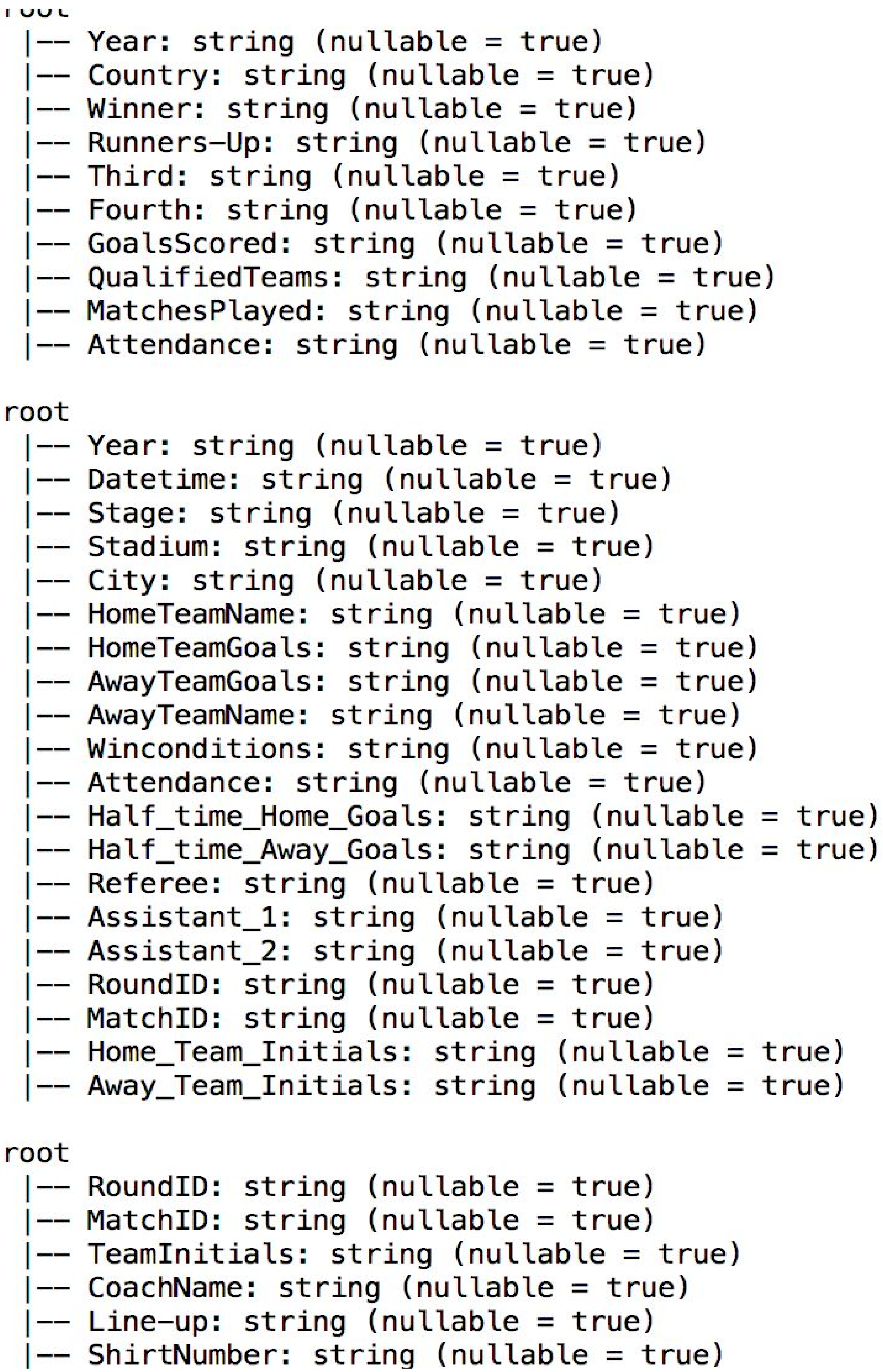
We have chosen Fifa World Cup dataset and are using all 3 Fifa datasets given in the Kaggle repository.

a) Importing the dataset and creation of datafra



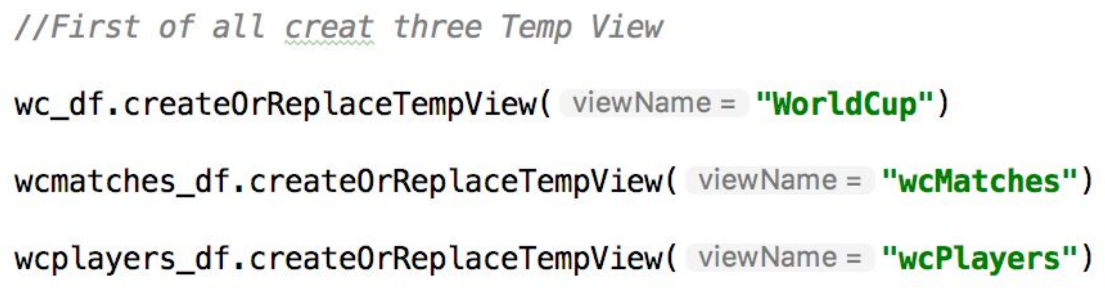
```
val wc_df = spark.read
  .format( source = "csv")
  .option("header", "true") //reading the headers
  .option("mode", "DROPMALFORMED")
  .load( path = "C:/Users/Sushu/Desktop/BDFiles/BigData_Lesson2" +
    "/WorldCups.csv")

val wcplayers_df = spark.read
  .format( source = "csv")
  .option("header", "true") //reading the headers
  .option("mode", "DROPMALFORMED")
  .load( path = "C:/Users/Sushu/Desktop/BDFiles/BigData_Lesson2" +
    "/WorldCupPlayers.csv")


val wcmatches_df = spark.read
  .format( source = "csv")
  .option("header", "true") //reading the headers
  .option("mode", "DROPMALFORMED")
  .load( path = "C:/Users/Sushu/Desktop/BDFiles/BigData_Lesson2" +
    "/WorldCupMatches.csv")
```

Printing the Schema for three datasets and Structure type.



```
 |-- Year: string (nullable = true)
 |-- Country: string (nullable = true)
 |-- Winner: string (nullable = true)
 |-- Runners-Up: string (nullable = true)
 |-- Third: string (nullable = true)
 |-- Fourth: string (nullable = true)
 |-- GoalsScored: string (nullable = true)
 |-- QualifiedTeams: string (nullable = true)
 |-- MatchesPlayed: string (nullable = true)
 |-- Attendance: string (nullable = true)

root
 |-- Year: string (nullable = true)
 |-- Datetime: string (nullable = true)
 |-- Stage: string (nullable = true)
 |-- Stadium: string (nullable = true)
 |-- City: string (nullable = true)
 |-- HomeTeamName: string (nullable = true)
 |-- HomeTeamGoals: string (nullable = true)
 |-- AwayTeamGoals: string (nullable = true)
 |-- AwayTeamName: string (nullable = true)
 |-- Winconditions: string (nullable = true)
 |-- Attendance: string (nullable = true)
 |-- Half_time_Home_Goals: string (nullable = true)
 |-- Half_time_Away_Goals: string (nullable = true)
 |-- Referee: string (nullable = true)
 |-- Assistant_1: string (nullable = true)
 |-- Assistant_2: string (nullable = true)
 |-- RoundID: string (nullable = true)
 |-- MatchID: string (nullable = true)
 |-- Home_Team_Initials: string (nullable = true)
 |-- Away_Team_Initials: string (nullable = true)

root
 |-- RoundID: string (nullable = true)
 |-- MatchID: string (nullable = true)
 |-- TeamInitials: string (nullable = true)
 |-- CoachName: string (nullable = true)
 |-- Line-up: string (nullable = true)
 |-- ShirtNumber: string (nullable = true)
```

b) Creating the tempView on Apache Spark using the above dataframes.

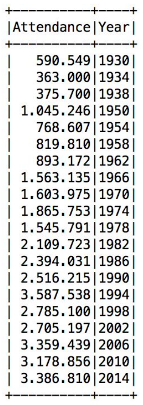


```
//First of all creat three Temp View

wc_df.createOrReplaceTempView( viewName = "WorldCup")

wcmatches_df.createOrReplaceTempView( viewName = "wcMatches")

wcplayers_df.createOrReplaceTempView( viewName = "wcPlayers")
```

c) Performing 10 Apache Sql Queries on view

1. Find the attendance by years using Worldcup view

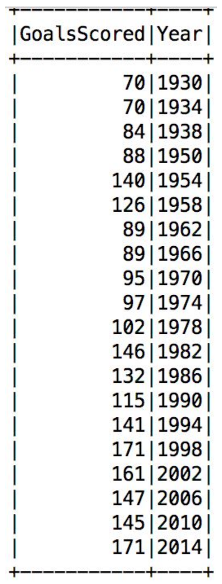
Query: **val** wcAtd = spark.sql(**"select Attendance,Year from WorldCup Order By Year"**)
wcAtd.show()



```
+----------+----+
|Attendance|Year|
+----------+----+
|   590.549|1930|
|   363.000|1934|
|   375.700|1938|
| 1.045.246|1950|
|   768.607|1954|
|   819.810|1958|
|   893.172|1962|
| 1.563.135|1966|
| 1.603.975|1970|
| 1.865.753|1974|
| 1.545.791|1978|
| 2.109.723|1982|
| 2.394.031|1986|
| 2.516.215|1990|
| 3.587.538|1994|
| 2.785.100|1998|
| 2.705.197|2002|
| 3.359.439|2006|
| 3.178.856|2010|
| 3.386.810|2014|
+----------+----+
```

2. Find the goals by years using WorldCup view

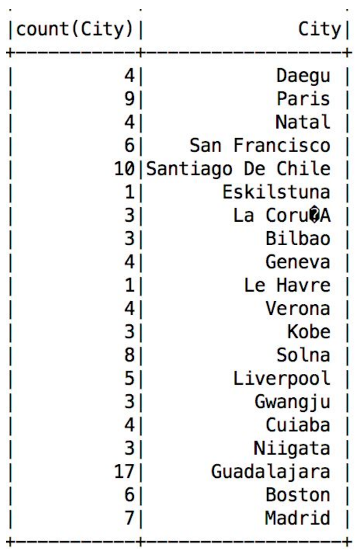
Query: **val** wcgoal = spark.sql(**"select GoalsScored,Year from WorldCup Order By Year"**)
wcgoal.show()



```
+-----------+----+
|GoalsScored|Year|
+-----------+----+
|         70|1930|
|         70|1934|
|         84|1938|
|         88|1950|
|        140|1954|
|        126|1958|
|         89|1962|
|         89|1966|
|         95|1970|
|         97|1974|
|        102|1978|
|        146|1982|
|        132|1986|
|        115|1990|
|        141|1994|
|        171|1998|
|        161|2002|
|        147|2006|
|        145|2010|
|        171|2014|
+-----------+----+
```

3. Find cities that hosted highest WorldCup matches

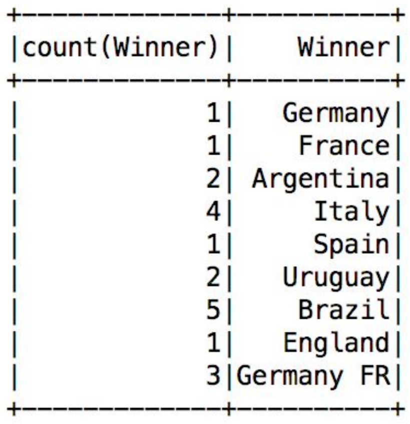
Query: **val** cityCount = spark.sql("**select Count(City),City from wcMatches GroupBy City**")
cityCount.show()



```
|count(City)|               City|
+-----------+-------------------+
|          4|             Daegu |
|          9|             Paris |
|          4|             Natal |
|          6|     San Francisco |
|         10|Santiago De Chile |
|          1|        Eskilstuna |
|          3|         La Coru�A |
|          3|            Bilbao |
|          4|            Geneva |
|          1|          Le Havre |
|          4|            Verona |
|          3|              Kobe |
|          8|             Solna |
|          5|         Liverpool |
|          3|           Gwangju |
|          4|            Cuiaba |
|          3|           Niigata |
|         17|       Guadalajara |
|          6|            Boston |
|          7|            Madrid |
+-----------+-------------------+
```

4. Teams with the most World Cup final victories on WorldCup view

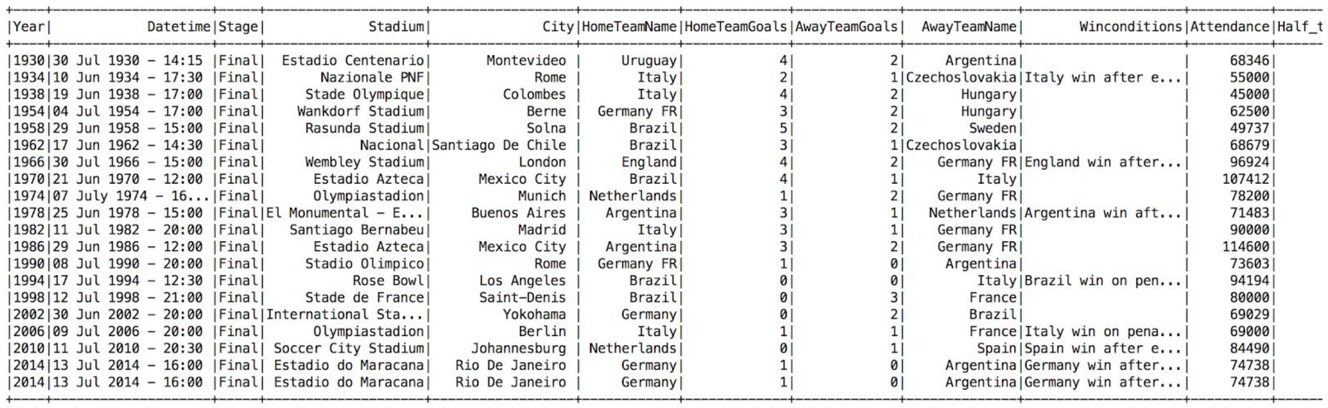
Query: **val** CountryWin = spark.sql(**"select Count(Winner),Winner from WorldCup Group By Winner"**)
CountryWin.show()



```
+-------------+----------+
|count(Winner)|    Winner|
+-------------+----------+
|            1|   Germany|
|            1|    France|
|            2| Argentina|
|            4|     Italy|
|            1|     Spain|
|            2|   Uruguay|
|            5|    Brazil|
|            1|   England|
|            3|Germany FR|
+-------------+----------+
```

5. Display all Stage Final Matches

Query: **val** FinalDF = spark.sql(**"select * from wcMatches where Stage='Final'"**)
FinalDF.show()

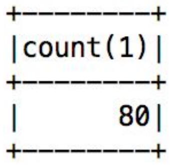


```
+----+--------------------+-----+--------------------+------------------+------------+-------------+-------------+--------------+--------------------+----------+------
|Year|            Datetime|Stage|             Stadium|              City|HomeTeamName|HomeTeamGoals|AwayTeamGoals|  AwayTeamName|       Winconditions|Attendance|Half_t
+----+--------------------+-----+--------------------+------------------+------------+-------------+-------------+--------------+--------------------+----------+------
|1930|30 Jul 1930 - 14:15 |Final|  Estadio Centenario|       Montevideo |     Uruguay|            4|            2|     Argentina|                    |     68346|
|1934|10 Jun 1934 - 17:30 |Final|       Nazionale PNF|             Rome |       Italy|            2|            1|Czechoslovakia|Italy win after e...|     55000|
|1938|19 Jun 1938 - 17:00 |Final|     Stade Olympique|         Colombes |       Italy|            4|            2|       Hungary|                    |     45000|
|1954|04 Jul 1954 - 17:00 |Final|    Wankdorf Stadium|            Berne |  Germany FR|            3|            2|       Hungary|                    |     62500|
|1958|29 Jun 1958 - 15:00 |Final|     Rasunda Stadium|            Solna |      Brazil|            5|            2|        Sweden|                    |     49737|
|1962|17 Jun 1962 - 14:30 |Final|            Nacional|Santiago De Chile |      Brazil|            3|            1|Czechoslovakia|                    |     68679|
|1966|30 Jul 1966 - 15:00 |Final|     Wembley Stadium|           London |     England|            4|            2|    Germany FR|England win after...|     96924|
|1970|21 Jun 1970 - 12:00 |Final|      Estadio Azteca|      Mexico City |      Brazil|            4|            1|         Italy|                    |    107412|
|1974|07 July 1974 - 16...|Final|      Olympiastadion|           Munich | Netherlands|            1|            2|    Germany FR|                    |     78200|
|1978|25 Jun 1978 - 15:00 |Final|El Monumental - E...|     Buenos Aires |   Argentina|            3|            1|   Netherlands|Argentina win aft...|     71483|
|1982|11 Jul 1982 - 20:00 |Final|   Santiago Bernabeu|           Madrid |       Italy|            3|            1|    Germany FR|                    |     90000|
|1986|29 Jun 1986 - 12:00 |Final|      Estadio Azteca|      Mexico City |   Argentina|            3|            2|    Germany FR|                    |    114600|
|1990|08 Jul 1990 - 20:00 |Final|     Stadio Olimpico|             Rome |  Germany FR|            1|            0|     Argentina|                    |     73603|
|1994|17 Jul 1994 - 12:30 |Final|           Rose Bowl|      Los Angeles |      Brazil|            0|            0|         Italy|Brazil win on pen...|     94194|
|1998|12 Jul 1998 - 21:00 |Final|     Stade de France|      Saint-Denis |      Brazil|            0|            3|        France|                    |     80000|
|2002|30 Jun 2002 - 20:00 |Final|International Sta...|         Yokohama |     Germany|            0|            2|        Brazil|                    |     69029|
|2006|09 Jul 2006 - 20:00 |Final|      Olympiastadion|           Berlin |       Italy|            1|            1|        France|Italy win on pena...|     69000|
|2010|11 Jul 2010 - 20:30 |Final| Soccer City Stadium|     Johannesburg | Netherlands|            0|            1|         Spain|Spain win after e...|     84490|
|2014|13 Jul 2014 - 16:00 |Final| Estadio do Maracana|   Rio De Janeiro |     Germany|            1|            0|     Argentina|Germany win after...|     74738|
|2014|13 Jul 2014 - 16:00 |Final| Estadio do Maracana|   Rio De Janeiro |     Germany|            1|            0|     Argentina|Germany win after...|     74738|
+----+--------------------+-----+--------------------+------------------+------------+-------------+-------------+--------------+--------------------+----------+------
```

6. Number of matches in year 2014

Query: **val** match2014 = spark.sql(**"select count(*) from wcMatches where year=2014"**)

match2014.show()

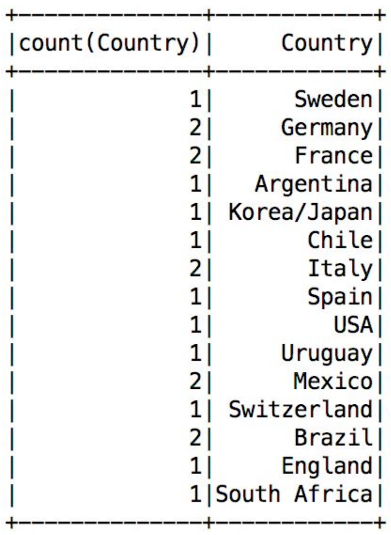


```
+--------+
|count(1)|
+--------+
|      80|
+--------+
```

7. Country which hosted WorldCup highest number of times

Query: **val** CountHost = spark.sql(**"select Count(Country),Country from WorldCup Group by Country"**)

CountHost.show()

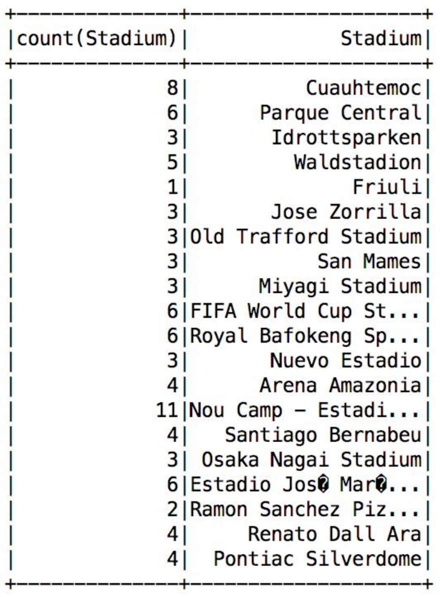


```
+--------------+------------+
|count(Country)|     Country|
+--------------+------------+
|             1|      Sweden|
|             2|     Germany|
|             2|      France|
|             1|   Argentina|
|             1| Korea/Japan|
|             1|       Chile|
|             2|       Italy|
|             1|       Spain|
|             1|         USA|
|             1|     Uruguay|
|             2|      Mexico|
|             1| Switzerland|
|             2|      Brazil|
|             1|     England|
|             1|South Africa|
+--------------+------------+
```

8. Stadium with highest number of Matches

Query: **val** StadmatchCount = spark.sql(**"select Count(Stadium),Stadium from wcMatches Group By Stadium"**)

StadmatchCount.show()

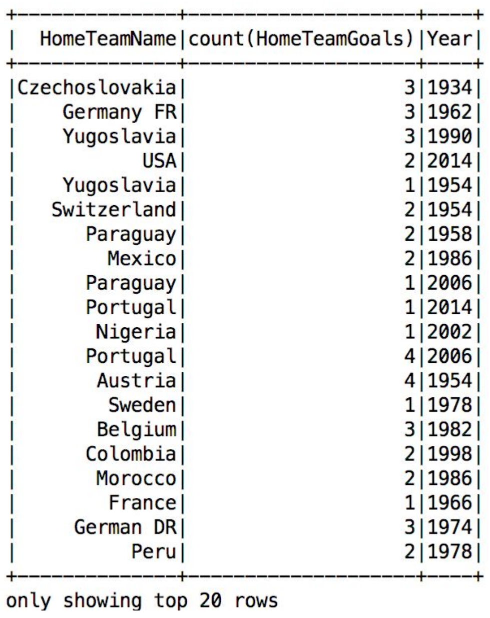


```
+--------------+--------------------+
|count(Stadium)|             Stadium|
+--------------+--------------------+
|             8|          Cuauhtemoc|
|             6|      Parque Central|
|             3|       Idrottsparken|
|             5|          Waldstadion|
|             1|              Friuli|
|             3|       Jose Zorrilla|
|             3|Old Trafford Stadium|
|             3|           San Mames|
|             3|      Miyagi Stadium|
|             6|FIFA World Cup St...|
|             6|Royal Bafokeng Sp...|
|             3|        Nuevo Estadio|
|             4|      Arena Amazonia|
|            11|Nou Camp - Estadi...|
|             4|    Santiago Bernabeu|
|             3| Osaka Nagai Stadium|
|             6|Estadio Jos� Mar�...|
|             2|Ramon Sanchez Piz...|
|             4|     Renato Dall Ara|
|             4|   Pontiac Silverdome|
+--------------+--------------------+
```

9. Home Team Goals count and Home Team Names by Years

Query: **val** homeGoals = spark.sql(**"select HomeTeamName,Count(HomeTeamGoals),Year from wcMatches Group By Year,HomeTeamName"**)

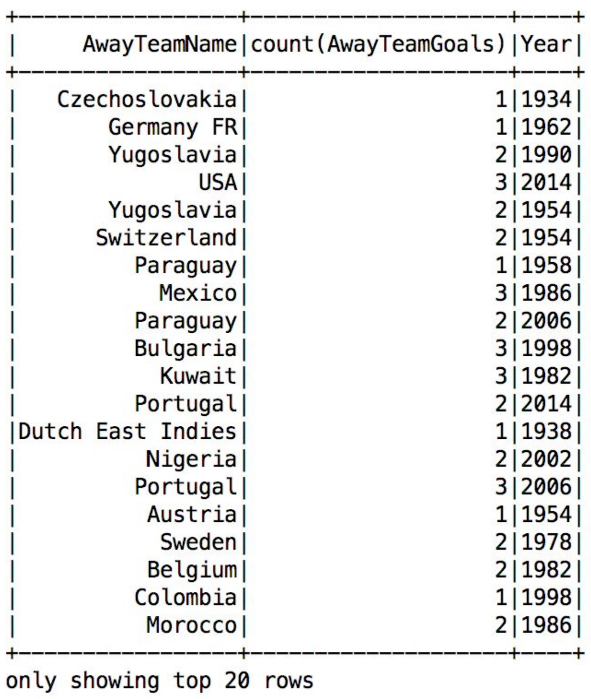
homeGoals.show()



```
+--------------+--------------------+----+
|  HomeTeamName|count(HomeTeamGoals)|Year|
+--------------+--------------------+----+
|Czechoslovakia|                   3|1934|
|    Germany FR|                   3|1962|
|    Yugoslavia|                   3|1990|
|           USA|                   2|2014|
|    Yugoslavia|                   1|1954|
|   Switzerland|                   2|1954|
|      Paraguay|                   2|1958|
|        Mexico|                   2|1986|
|      Paraguay|                   1|2006|
|      Portugal|                   1|2014|
|       Nigeria|                   1|2002|
|      Portugal|                   4|2006|
|       Austria|                   4|1954|
|        Sweden|                   1|1978|
|       Belgium|                   3|1982|
|      Colombia|                   2|1998|
|       Morocco|                   2|1986|
|        France|                   1|1966|
|     German DR|                   3|1974|
|          Peru|                   2|1978|
+--------------+--------------------+----+
only showing top 20 rows
```

10. Away Team Goals count by Years

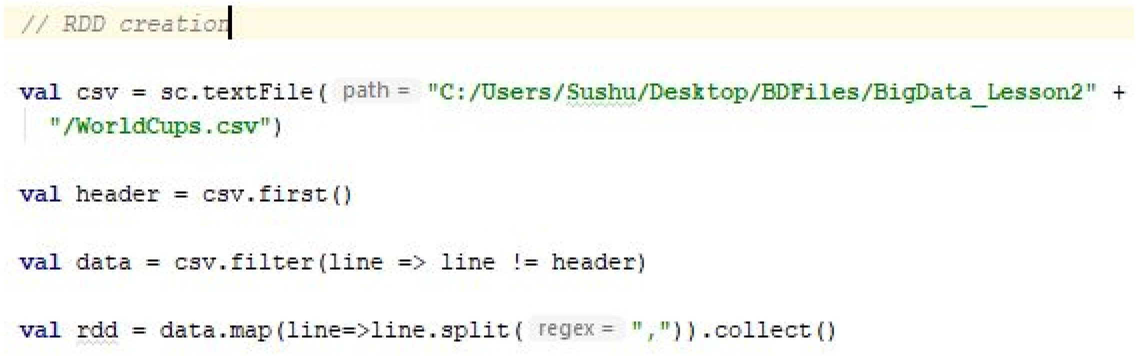
Query:
```
val awayTeamGoals = spark.sql("select
    AwayTeamName,Count(AwayTeamGoals),Year from wcMatches
    Group By Year,AwayTeamName")

    awayTeamGoals.show()
```



```
+-----------------+--------------------+----+
|     AwayTeamName|count(AwayTeamGoals)|Year|
+-----------------+--------------------+----+
|   Czechoslovakia|                   1|1934|
|       Germany FR|                   1|1962|
|       Yugoslavia|                   2|1990|
|              USA|                   3|2014|
|       Yugoslavia|                   2|1954|
|      Switzerland|                   2|1954|
|         Paraguay|                   1|1958|
|           Mexico|                   3|1986|
|         Paraguay|                   2|2006|
|         Bulgaria|                   3|1998|
|           Kuwait|                   3|1982|
|         Portugal|                   2|2014|
|Dutch East Indies|                   1|1938|
|          Nigeria|                   2|2002|
|         Portugal|                   3|2006|
|          Austria|                   1|1954|
|           Sweden|                   2|1978|
|          Belgium|                   2|1982|
|         Colombia|                   1|1998|
|          Morocco|                   2|1986|
+-----------------+--------------------+----+
only showing top 20 rows
```

b) Perform 5 queries in Spark's RDD and Spark DataFrames

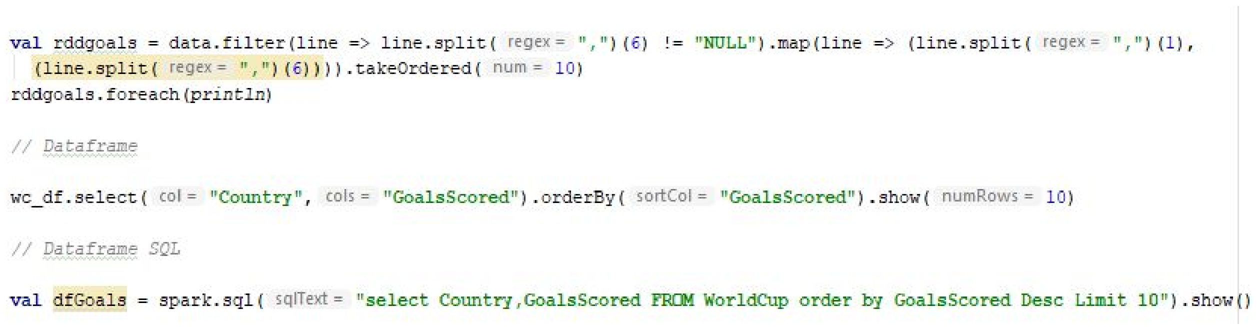
We have first created RDD as follows



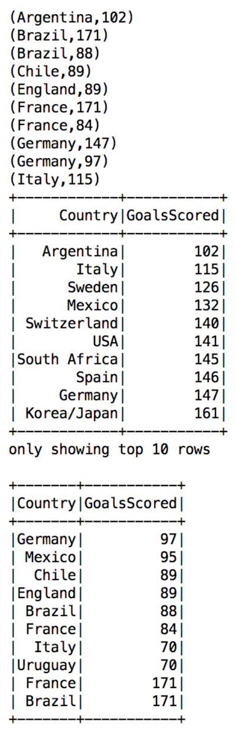
```
// RDD creation

val csv = sc.textFile( path = "C:/Users/Sushu/Desktop/BDFiles/BigData_Lesson2" +
  "/WorldCups.csv")

val header = csv.first()

val data = csv.filter(line => line != header)

val rdd = data.map(line=>line.split( regex = ",")).collect()
```

1. Find Highest Number of Goals

Query

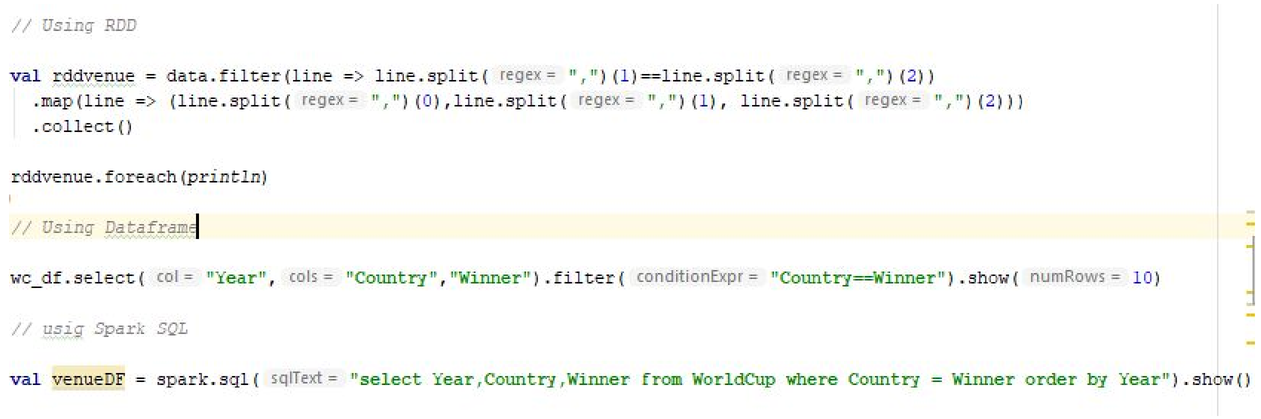


```
val rddgoals = data.filter(line => line.split( regex = ",")(6)  != "NULL").map(line => (line.split( regex = ",")(1),
  (line.split( regex = ",")(6)))).takeOrdered( num = 10)
rddgoals.foreach(println)

// Dataframe

wc_df.select( col = "Country", cols = "GoalsScored").orderBy( sortCol = "GoalsScored").show( numRows = 10)

// Dataframe SQL

val dfGoals = spark.sql( sqlText = "select Country,GoalsScored FROM WorldCup order by GoalsScored Desc Limit 10").show()
```

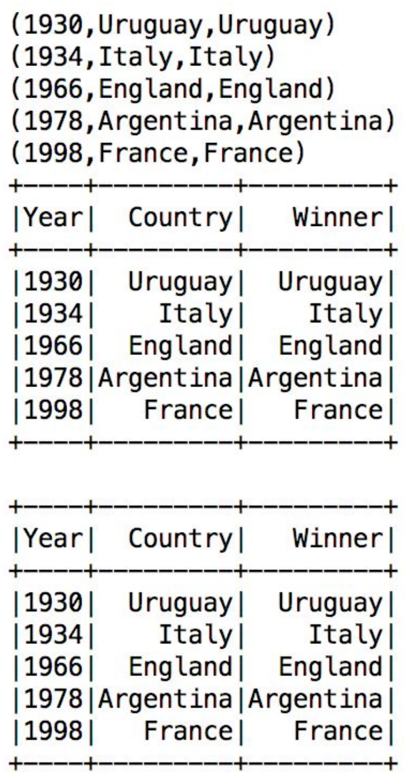


```
(Argentina,102)
(Brazil,171)
(Brazil,88)
(Chile,89)
(England,89)
(France,171)
(France,84)
(Germany,147)
(Germany,97)
(Italy,115)
+------------+-----------+
|     Country|GoalsScored|
+------------+-----------+
|   Argentina|        102|
|       Italy|        115|
|      Sweden|        126|
|      Mexico|        132|
| Switzerland|        140|
|         USA|        141|
|South Africa|        145|
|       Spain|        146|
|     Germany|        147|
| Korea/Japan|        161|
+------------+-----------+
only showing top 10 rows

+-------+-----------+
|Country|GoalsScored|
+-------+-----------+
|Germany|         97|
| Mexico|         95|
|  Chile|         89|
|England|         89|
| Brazil|         88|
| France|         84|
|  Italy|         70|
|Uruguay|         70|
| France|        171|
| Brazil|        171|
+-------+-----------+
```

2. Retrieve all the hosting countries who are winning countries along with the year. Query:



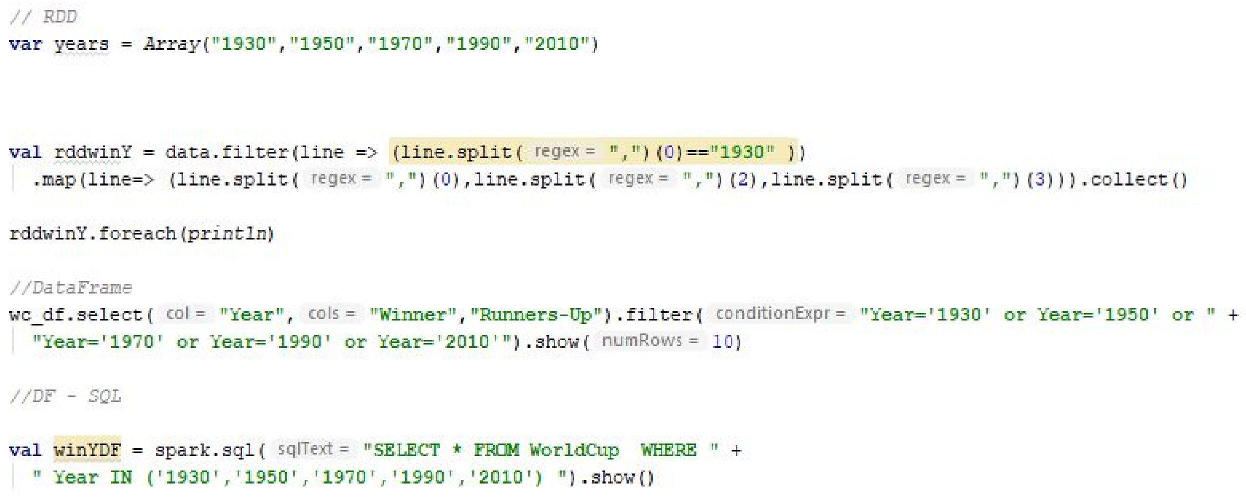
```
// Using RDD

val rddvenue = data.filter(line => line.split( regex = ",")(1)==line.split( regex = ",")(2))
  .map(line => (line.split( regex = ",")(0),line.split( regex = ",")(1), line.split( regex = ",")(2)))
  .collect()

rddvenue.foreach(println)

// Using Dataframe

wc_df.select( col = "Year", cols = "Country","Winner").filter( conditionExpr = "Country==Winner").show( numRows = 10)

// usig Spark SQL

val venueDF = spark.sql( sqlText = "select Year,Country,Winner from WorldCup where Country = Winner order by Year").show()
```



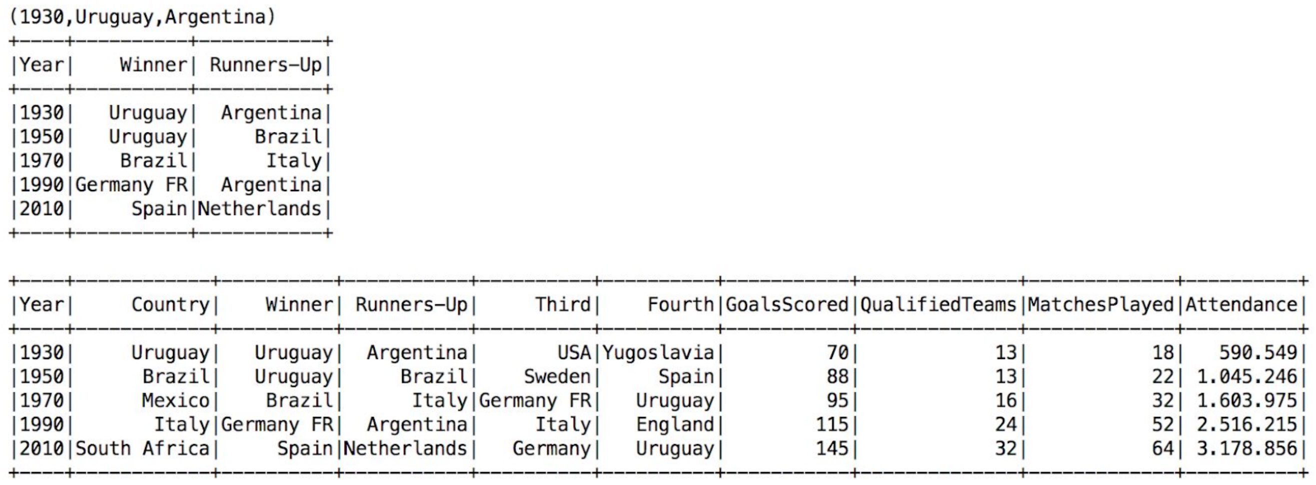
```
(1930,Uruguay,Uruguay)
(1934,Italy,Italy)
(1966,England,England)
(1978,Argentina,Argentina)
(1998,France,France)
+----+---------+---------+
|Year|  Country|   Winner|
+----+---------+---------+
|1930|  Uruguay|  Uruguay|
|1934|    Italy|    Italy|
|1966|  England|  England|
|1978|Argentina|Argentina|
|1998|   France|   France|
+----+---------+---------+

+----+---------+---------+
|Year|  Country|   Winner|
+----+---------+---------+
|1930|  Uruguay|  Uruguay|
|1934|    Italy|    Italy|
|1966|  England|  England|
|1978|Argentina|Argentina|
|1998|   France|   France|
+----+---------+---------+
```

3. Details of years ending with zero

Query:



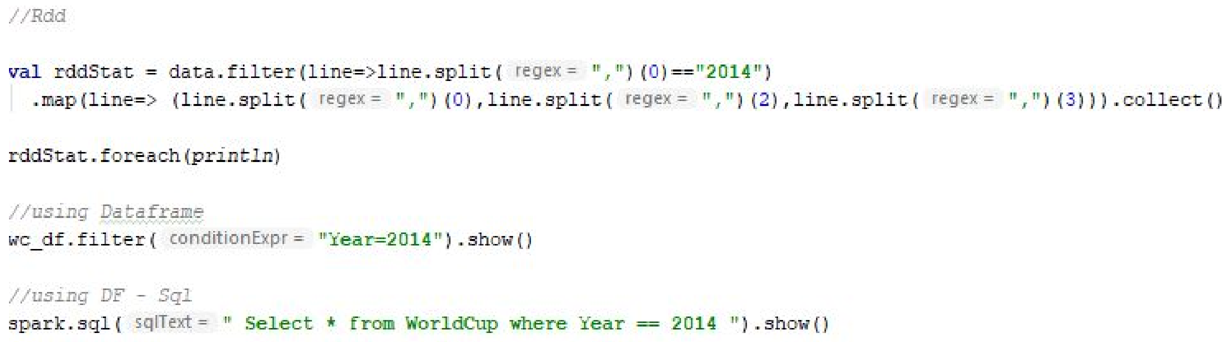
```
// RDD
var years = Array("1930","1950","1970","1990","2010")



val rddwinY = data.filter(line => (line.split( regex = ",")(0)=="1930" ))
  .map(line=> (line.split( regex = ",")(0),line.split( regex = ",")(2),line.split( regex = ",")(3))).collect()

rddwinY.foreach(println)

//DataFrame
wc_df.select( col = "Year", cols = "Winner","Runners-Up").filter( conditionExpr = "Year='1930' or Year='1950' or " +
  "Year='1970' or Year='1990' or Year='2010'").show( numRows = 10)

//DF - SQL

val winYDF = spark.sql( sqlText = "SELECT * FROM WorldCup  WHERE " +
  " Year IN ('1930','1950','1970','1990','2010') ").show()
```



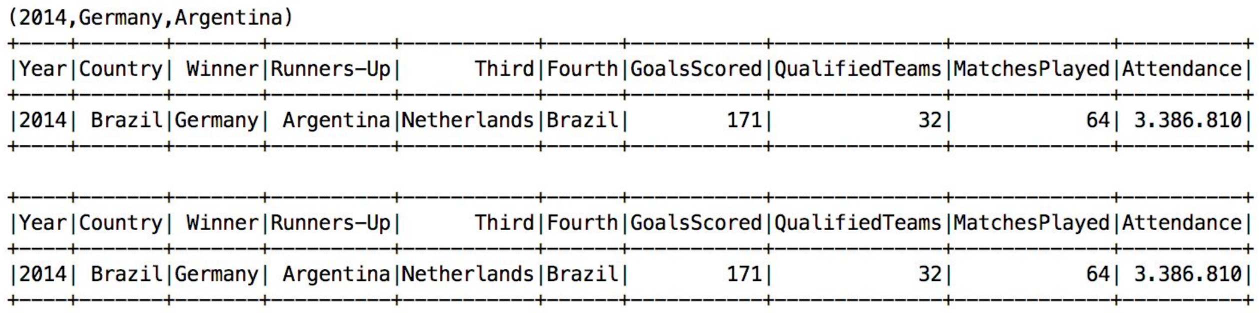
```
(1930,Uruguay,Argentina)
+----+----------+-----------+
|Year|    Winner| Runners-Up|
+----+----------+-----------+
|1930|   Uruguay|  Argentina|
|1950|   Uruguay|     Brazil|
|1970|    Brazil|      Italy|
|1990|Germany FR|  Argentina|
|2010|     Spain|Netherlands|
+----+----------+-----------+

+----+------------+----------+-----------+----------+----------+-----------+--------------+-------------+----------+
|Year|     Country|    Winner| Runners-Up|     Third|    Fourth|GoalsScored|QualifiedTeams|MatchesPlayed|Attendance|
+----+------------+----------+-----------+----------+----------+-----------+--------------+-------------+----------+
|1930|     Uruguay|   Uruguay|  Argentina|       USA|Yugoslavia|         70|            13|           18|   590.549|
|1950|      Brazil|   Uruguay|     Brazil|    Sweden|     Spain|         88|            13|           22| 1.045.246|
|1970|      Mexico|    Brazil|      Italy|Germany FR|   Uruguay|         95|            16|           32| 1.603.975|
|1990|       Italy|Germany FR|  Argentina|     Italy|   England|        115|            24|           52| 2.516.215|
|2010|South Africa|     Spain|Netherlands|   Germany|   Uruguay|        145|            32|           64| 3.178.856|
+----+------------+----------+-----------+----------+----------+-----------+--------------+-------------+----------+
```

4. Retrieve all the details of the World Cup match organised in 2014

Query:



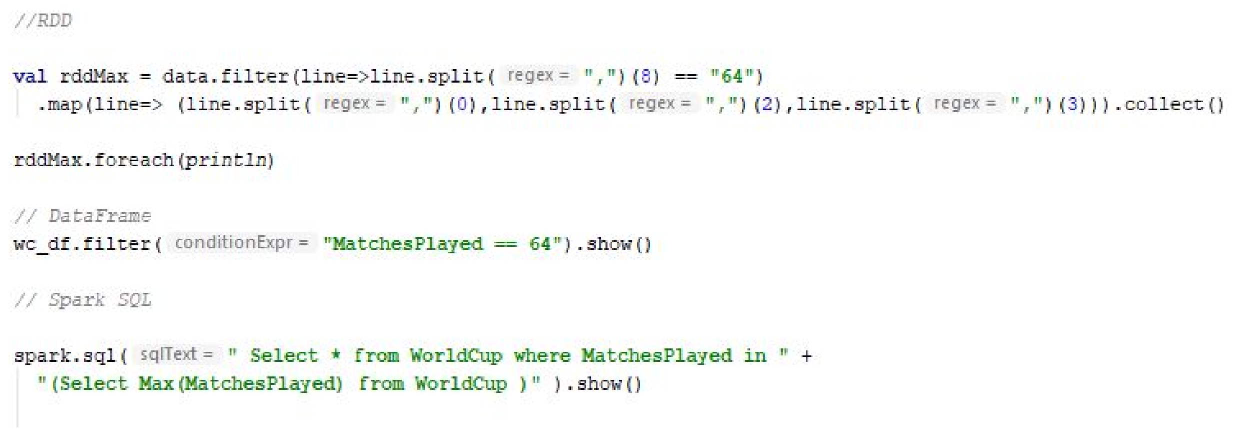
```
//Rdd

val rddStat = data.filter(line=>line.split( regex = ",")(0)=="2014")
  .map(line=>  (line.split( regex = ",")(0),line.split( regex = ",")(2),line.split( regex = ",")(3))).collect()

rddStat.foreach(println)

//using Dataframe
wc_df.filter( conditionExpr = "Year=2014").show()

//using DF - Sql
spark.sql( sqlText = " Select * from WorldCup where Year == 2014 ").show()
```



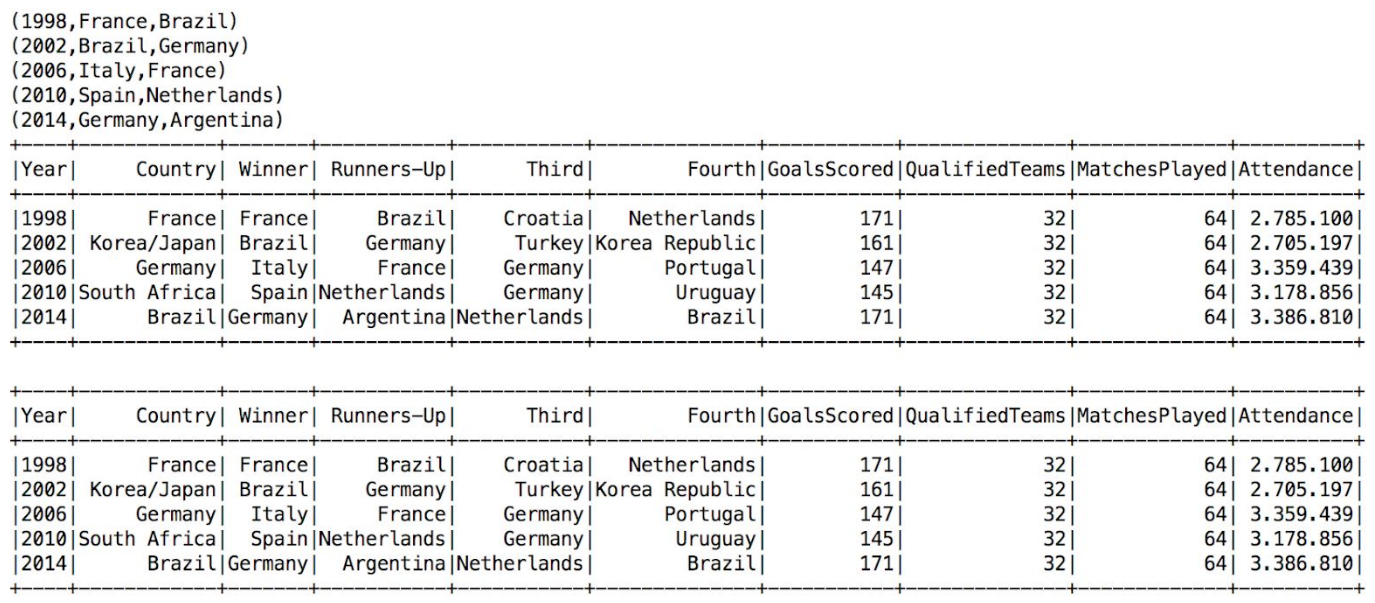
```
(2014,Germany,Argentina)
+----+-------+-------+----------+-----------+------+-----------+--------------+-------------+----------+
|Year|Country| Winner|Runners-Up|      Third|Fourth|GoalsScored|QualifiedTeams|MatchesPlayed|Attendance|
+----+-------+-------+----------+-----------+------+-----------+--------------+-------------+----------+
|2014| Brazil|Germany| Argentina|Netherlands|Brazil|        171|            32|           64| 3.386.810|
+----+-------+-------+----------+-----------+------+-----------+--------------+-------------+----------+

+----+-------+-------+----------+-----------+------+-----------+--------------+-------------+----------+
|Year|Country| Winner|Runners-Up|      Third|Fourth|GoalsScored|QualifiedTeams|MatchesPlayed|Attendance|
+----+-------+-------+----------+-----------+------+-----------+--------------+-------------+----------+
|2014| Brazil|Germany| Argentina|Netherlands|Brazil|        171|            32|           64| 3.386.810|
+----+-------+-------+----------+-----------+------+-----------+--------------+-------------+----------+
```

5. Maximum Matches Played

Query:



```
//RDD

val rddMax = data.filter(line=>line.split( regex = ",")(8) == "64")
  .map(line=> (line.split( regex = ",")(0),line.split( regex = ",")(2),line.split( regex = ",")(3))).collect()

rddMax.foreach(println)

// DataFrame
wc_df.filter( conditionExpr = "MatchesPlayed == 64").show()

// Spark SQL

spark.sql( sqlText = " Select * from WorldCup where MatchesPlayed in " +
  "(Select Max(MatchesPlayed) from WorldCup )" ).show()
```



```
(1998,France,Brazil)
(2002,Brazil,Germany)
(2006,Italy,France)
(2010,Spain,Netherlands)
(2014,Germany,Argentina)
+----+------------+-------+-----------+-----------+--------------+-----------+--------------+-------------+----------+
|Year|     Country| Winner| Runners-Up|      Third|        Fourth|GoalsScored|QualifiedTeams|MatchesPlayed|Attendance|
+----+------------+-------+-----------+-----------+--------------+-----------+--------------+-------------+----------+
|1998|      France| France|     Brazil|    Croatia|   Netherlands|        171|            32|           64| 2.785.100|
|2002| Korea/Japan| Brazil|    Germany|     Turkey|Korea Republic|        161|            32|           64| 2.705.197|
|2006|     Germany|  Italy|     France|    Germany|      Portugal|        147|            32|           64| 3.359.439|
|2010|South Africa|  Spain|Netherlands|    Germany|       Uruguay|        145|            32|           64| 3.178.856|
|2014|      Brazil|Germany|  Argentina|Netherlands|        Brazil|        171|            32|           64| 3.386.810|
+----+------------+-------+-----------+-----------+--------------+-----------+--------------+-------------+----------+

+----+------------+-------+-----------+-----------+--------------+-----------+--------------+-------------+----------+
|Year|     Country| Winner| Runners-Up|      Third|        Fourth|GoalsScored|QualifiedTeams|MatchesPlayed|Attendance|
+----+------------+-------+-----------+-----------+--------------+-----------+--------------+-------------+----------+
|1998|      France| France|     Brazil|    Croatia|   Netherlands|        171|            32|           64| 2.785.100|
|2002| Korea/Japan| Brazil|    Germany|     Turkey|Korea Republic|        161|            32|           64| 2.705.197|
|2006|     Germany|  Italy|     France|    Germany|      Portugal|        147|            32|           64| 3.359.439|
|2010|South Africa|  Spain|Netherlands|    Germany|       Uruguay|        145|            32|           64| 3.178.856|
|2014|      Brazil|Germany|  Argentina|Netherlands|        Brazil|        171|            32|           64| 3.386.810|
+----+------------+-------+-----------+-----------+--------------+-----------+--------------+-------------+----------+
```

Evaluation:

We can see from the execution of above queries that Dataframe and Apache Spark Sql provides better performance and faster query results compared to RDD.

Below are the Comparison of RDD and Dataframe

1. **Optimization :** RDD doesn't provide built in optimization while Dataframe does using Catalyst optimizer.
2. **Garbage Collection :** There is overhead of garbage Collection associated with RDD, Dataframe avoid same while object creation phase.
3. **Type Safety :** RDD provide Compile type Dataframe provide Runtime Safety.
4. **Aggregation :** Slower in RDD while faster in Dataframe

5. **Interopertability :** RDD get converted to Data Frames while Dataframe can not be converted to RDD.

Conclusion:

DataFrame and Apache Spark Sql are better the RDD for doing Query Analysis.

References:

https://data-flair.training/blogs/spark-rdd-operations-transformations-actions/

https://datascienceplus.com/dataframes-vs-rdds-in-spark-part-1/

https://medium.com/@joydeepubuntu/create-dataframes-in-spark-using-scala-6a33dd4bf15e