UNIVERSITY OF MISSOURI-KANSAS CITY

Bigdata Hadoop Programming

# Lab 3 Assignment

Team Members:

Pragathi Thammaneni

Sridevi Mallipudi

Introduction:

The main core concept for executing the Lab 3 Assignment is to implement the MapReduce Algorithm for finding Facebook common friends problem and run the MapReduce job on Apache Spark. And also, to implement the Spark Data frames and including intuitive queries like pattern recognition, topic discussion, most important terms etc.

Objectives:

To code for the 2 questions the below concepts are implemented.

Map reduce Algorithm for Facebook common friend's problem

Apache Spark

Spark Data frames

Datasets -queries

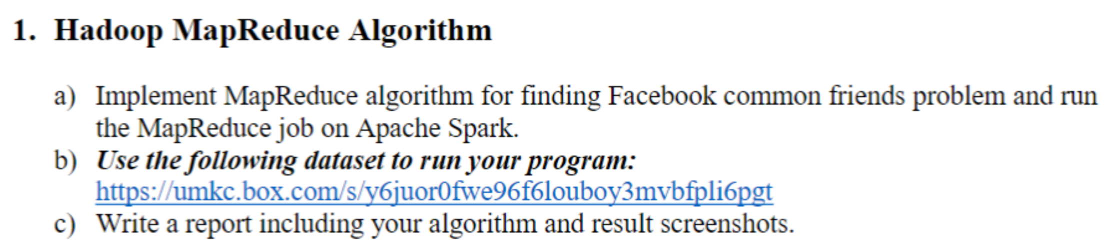
Approaches /Methods:

Apache Spark

Workflow &Datasets/Parameters and Evaluation:

The below each question will follow different approaches to solve. Coding is done to perform the evaluation of each individual snippet to execute the datasets which are provided as the input parameters.

Question 1:

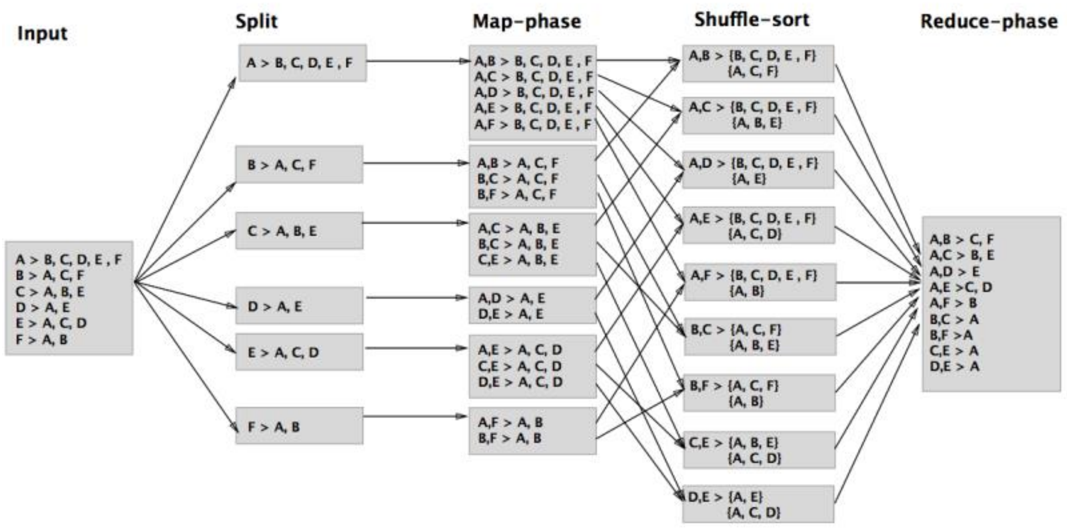


1. Hadoop MapReduce Algorithm

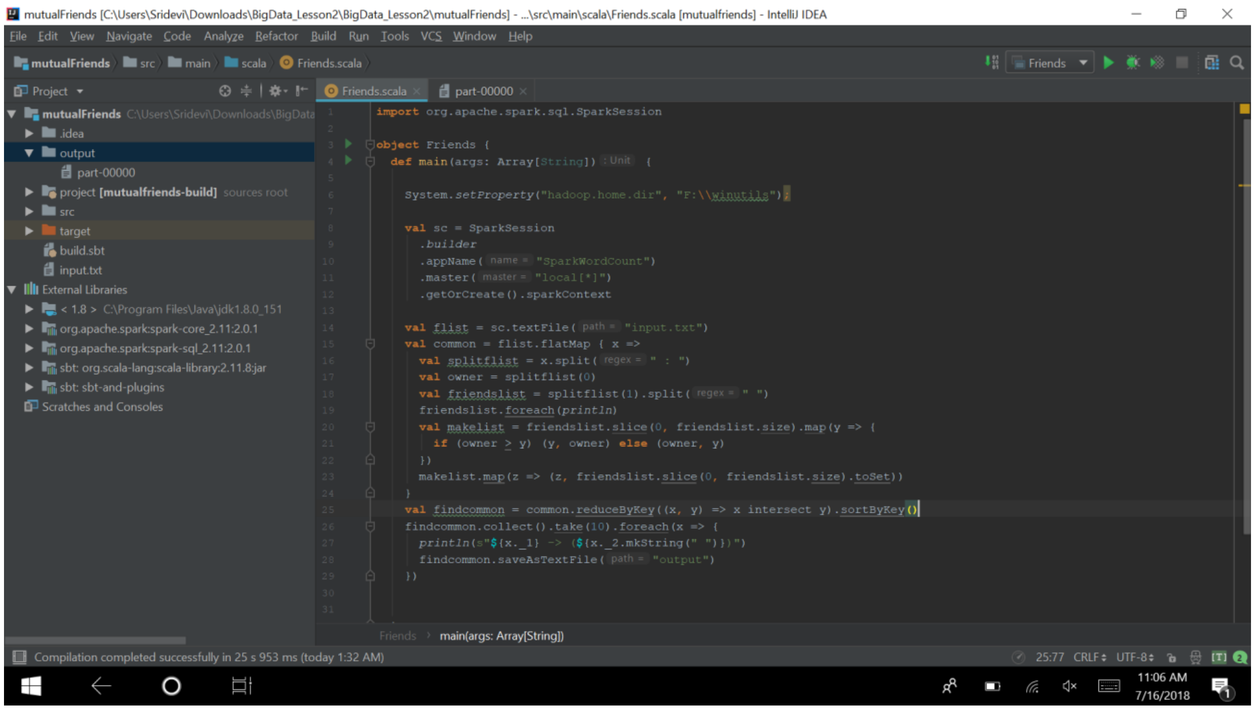
a) Implement MapReduce algorithm for finding Facebook common friends problem and run the MapReduce job on Apache Spark.
b) ***Use the following dataset to run your program:*** https://umkc.box.com/s/y6juor0fwe96f6louboy3mvbfpli6pgt
c) Write a report including your algorithm and result screenshots.

For solving the below question a map reduce algorithm is used to find out the face book common friend problem. Below is the sample use case for implementing the Map reduce.

Use Case Diagram:



Below is the screen shot for the Map Reduce Algorithm

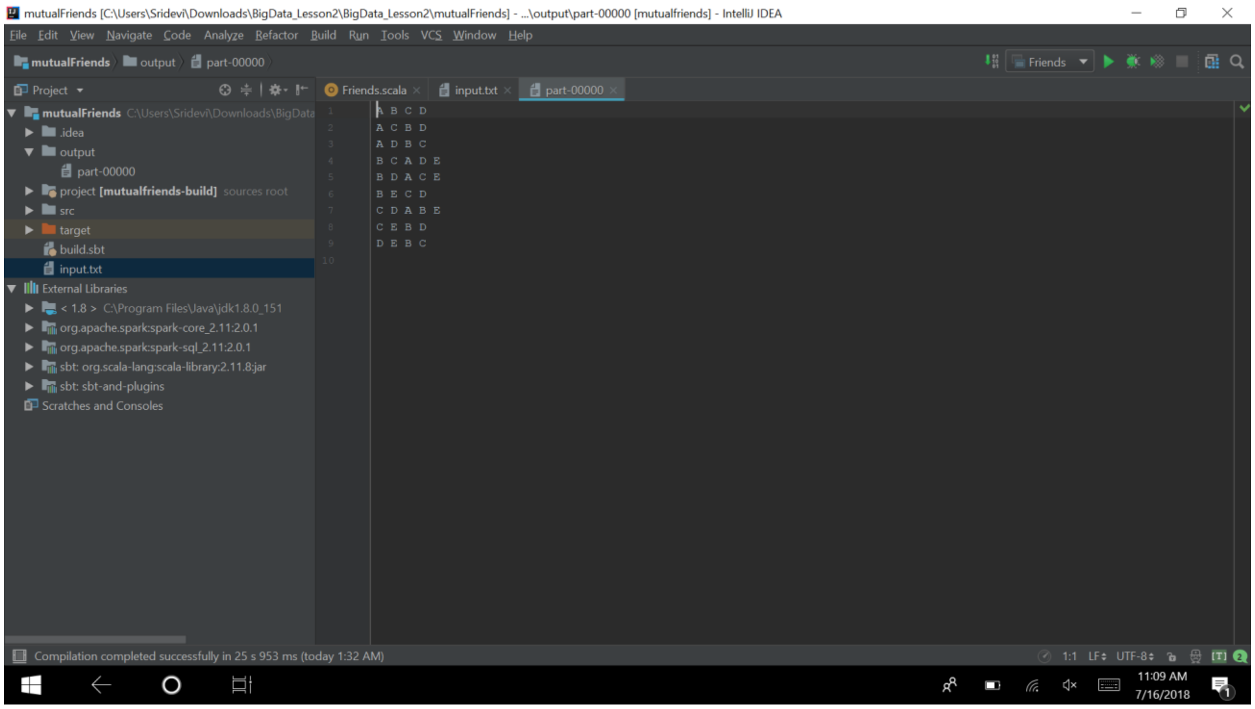


mutualFriends [C:\Users\Sridevi\Downloads\BigData_Lesson2\BigData_Lesson2\mutualFriends] - ...\src\main\scala\Friends.scala [mutualfriends] - IntelliJ IDEA

```scala
import org.apache.spark.sql.SparkSession

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

    System.setProperty("hadoop.home.dir", "F:\\winutils");

    val sc = SparkSession
      .builder
      .appName( name = "SparkWordCount")
      .master( master = "local[*]")
      .getOrCreate().sparkContext

    val flist = sc.textFile( path = "input.txt")
    val common = flist.flatMap { x =>
      val splitflist = x.split( regex = " : ")
      val owner = splitflist(0)
      val friendslist = splitflist(1).split( regex = " ")
      friendslist.foreach(println)
      val makelist = friendslist.slice(0, friendslist.size).map(y => {
        if (owner > y) (y, owner) else (owner, y)
      })
      makelist.map(z => (z, friendslist.slice(0, friendslist.size).toSet))
    }
    val findcommon = common.reduceByKey((x, y) => x intersect y).sortByKey()
    findcommon.collect().take(10).foreach(x => {
      println(s"${x._1} -> (${x._2.mkString(" ")})")
      findcommon.saveAsTextFile( path = "output")
    })
```

Compilation completed successfully in 25 s 953 ms (today 1:32 AM)

Output Screenshot:


mutualFriends [C:\Users\Sridevi\Downloads\BigData_Lesson2\BigData_Lesson2\mutualFriends] - ...\output\part-00000 [mutualfriends] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
mutualFriends output part-00000
Friends
Project
Friends.scala
input.txt
part-00000
mutualFriends C:\Users\Sridevi\Downloads\BigData
.idea
output
part-00000
project [mutualfriends-build] sources root
src
target
build.sbt
input.txt
External Libraries
< 1.8 > C:\Program Files\Java\jdk1.8.0_151
org.apache.spark:spark-core_2.11:2.0.1
org.apache.spark:spark-sql_2.11:2.0.1
sbt: org.scala-lang:scala-library:2.11.8:jar
sbt: sbt-and-plugins
Scratches and Consoles
1:1 LF UTF-8
11:09 AM
7/16/2018

```
A B C D
A C B D
A D B C
B C A D E
B D A C E
B E C D
C D A B E
C E B D
D E B C
```

Compilation completed successfully in 25 s 953 ms (today 1:32 AM)

Question 2:

2. **Spark Data Frames**

Datasets:

1. **FIFA World Cup:**
   https://www.kaggle.com/abecklas/fifa-world-cup#WorldCupMatches.csv
2. **Kickstarter Projects**
   https://www.kaggle.com/kemical/kickstarter-projects
3. **Google-Landmarks Dataset**
   https://www.kaggle.com/google/google-landmarks-dataset

   a. **Create a Spark DataFrame using one of datasets, trying to use all different StructType.**
   b. **Perform 10 intuitive questions in Dataset (e.g.: pattern recognition, topic discussion, most important terms, etc.). Use your innovation to think out of box.**
   c. **Perform any 5 queries in Spark RDD's and Spark Data Frames. Compare the results**

Queries implemented:

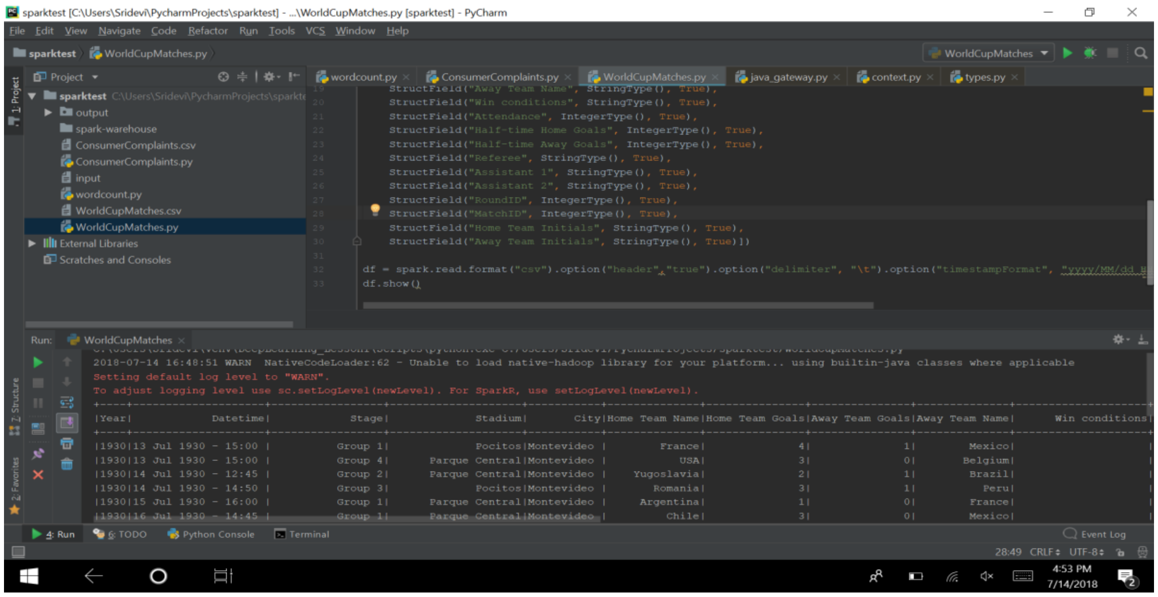
Below are the queries that has been implemented on the data sets FIFA World Cup.

1. Creation of Spark Data Frame using FIFA World Cup.
2. Queries regarding pattern recognition, topic discussion and most important terms has been performed.

3. Performed queries using Spark RDD’s and Spark Data Frames.

Please refer the source code folder for the queries description

Struct Type:

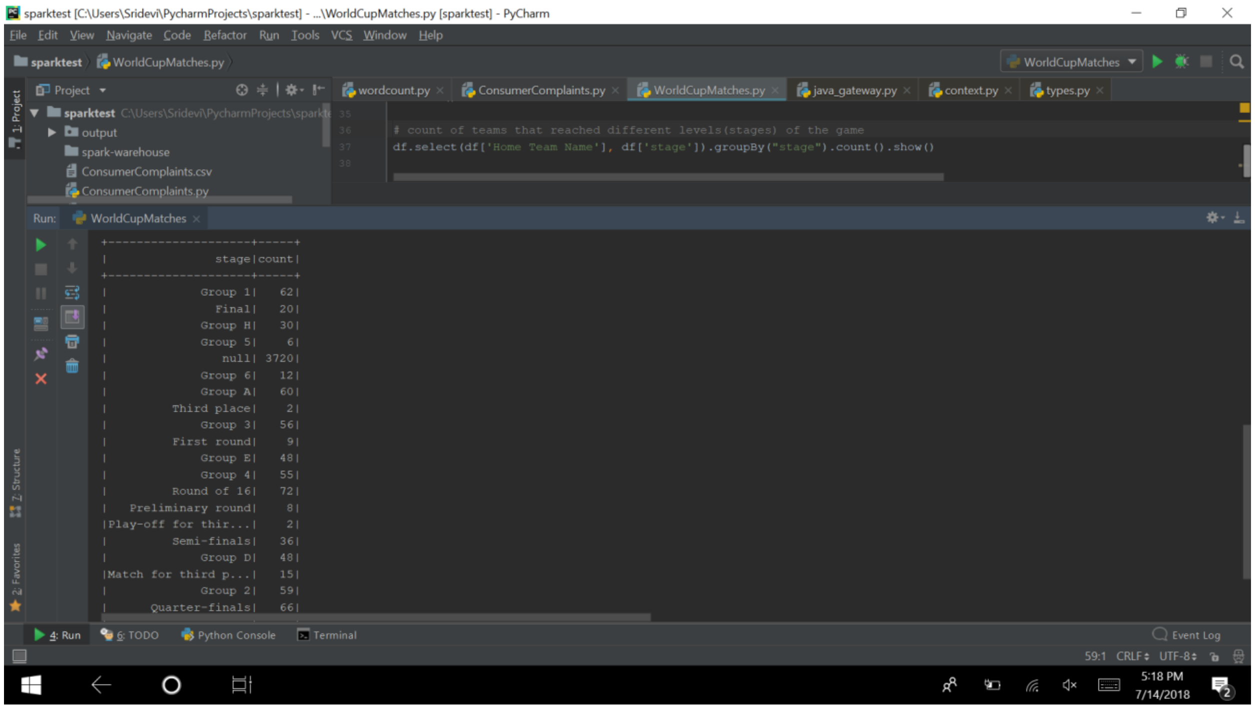


```
    StructField("Away Team Name", StringType(), True),
    StructField("Win conditions", StringType(), True),
    StructField("Attendance", IntegerType(), True),
    StructField("Half-time Home Goals", IntegerType(), True),
    StructField("Half-time Away Goals", IntegerType(), True),
    StructField("Referee", StringType(), True),
    StructField("Assistant 1", StringType(), True),
    StructField("Assistant 2", StringType(), True),
    StructField("RoundID", IntegerType(), True),
    StructField("MatchID", IntegerType(), True),
    StructField("Home Team Initials", StringType(), True),
    StructField("Away Team Initials", StringType(), True)])

df = spark.read.format("csv").option("header","true").option("delimiter", "\t").option("timestampFormat", "yyyy/MM/dd H
df.show()
```

```
2018-07-14 16:48:51 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
+----+------------------+---------------+-------------------+----------+---------------+---------------+---------------+---------------+----------------
|Year|          Datetime|          Stage|            Stadium|      City|Home Team Name|Home Team Goals|Away Team Goals|Away Team Name|    Win conditions|
+----+------------------+---------------+-------------------+----------+---------------+---------------+---------------+---------------+----------------
|1930|13 Jul 1930 - 15:00 |        Group 1|            Pocitos|Montevideo |         France|              4|              1|         Mexico|                |
|1930|13 Jul 1930 - 15:00 |        Group 4|     Parque Central|Montevideo |            USA|              3|              0|        Belgium|                |
|1930|14 Jul 1930 - 12:45 |        Group 2|     Parque Central|Montevideo |     Yugoslavia|              2|              1|         Brazil|                |
|1930|14 Jul 1930 - 14:50 |        Group 3|            Pocitos|Montevideo |        Romania|              3|              1|           Peru|                |
|1930|15 Jul 1930 - 16:00 |        Group 1|     Parque Central|Montevideo |      Argentina|              1|              0|         France|                |
|1930|16 Jul 1930 - 14:45 |        Group 1|     Parque Central|Montevideo |          Chile|              3|              0|         Mexico|                |
```

10 intuitive queries:

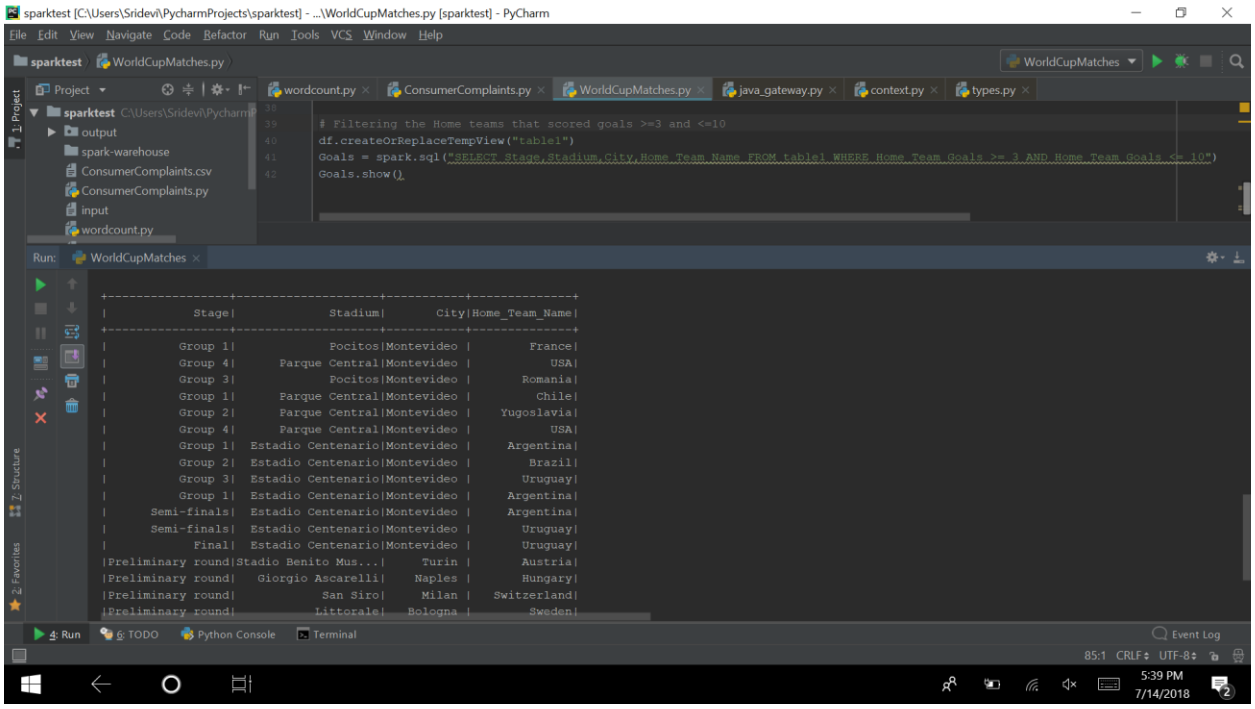
Query 1:Count of teams that reached different levels(stages) of the game



```
# count of teams that reached different levels(stages) of the game
df.select(df['Home Team Name'], df['stage']).groupBy("stage").count().show()
```

```
+--------------------+-----+
|               stage|count|
+--------------------+-----+
|             Group 1|   62|
|               Final|   20|
|             Group H|   30|
|             Group 5|    6|
|                null| 3720|
|             Group 6|   12|
|             Group A|   60|
|         Third place|    2|
|             Group 3|   56|
|         First round|    9|
|             Group E|   48|
|             Group 4|   55|
|         Round of 16|   72|
|   Preliminary round|    8|
|Play-off for thir...|    2|
|         Semi-finals|   36|
|             Group D|   48|
|Match for third p...|   15|
|             Group 2|   59|
|      Quarter-finals|   66|
```

Query 2:Filtering the Home teams that scored goals >=3 and <=10

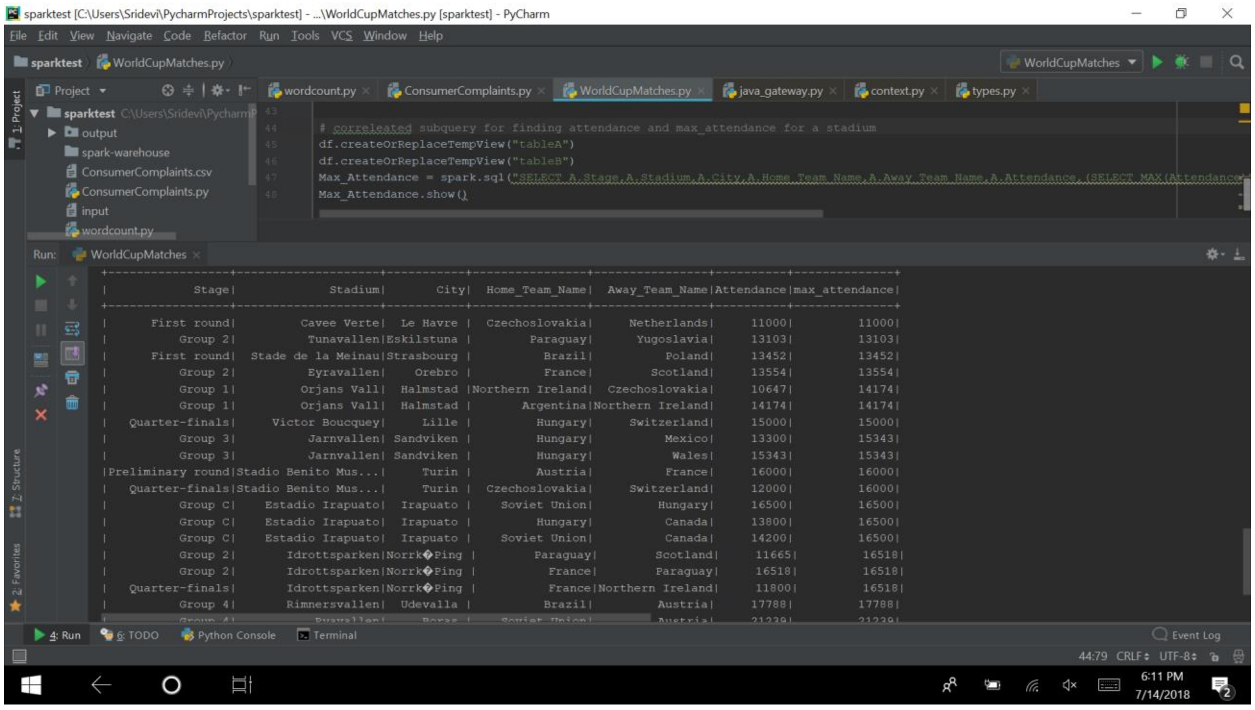


sparktest [C:\Users\Sridevi\PycharmProjects\sparktest] - ...\WorldCupMatches.py [sparktest] - PyCharm

```
# Filtering the Home teams that scored goals >=3 and <=10
df.createOrReplaceTempView("table1")
Goals = spark.sql("SELECT Stage,Stadium,City,Home_Team_Name FROM table1 WHERE Home_Team_Goals >= 3 AND Home_Team_Goals <= 10")
Goals.show()
```

```
+-----------------+--------------------+-----------+--------------+
|            Stage|             Stadium|       City|Home_Team_Name|
+-----------------+--------------------+-----------+--------------+
|          Group 1|             Pocitos|Montevideo |        France|
|          Group 4|      Parque Central|Montevideo |           USA|
|          Group 3|             Pocitos|Montevideo |       Romania|
|          Group 1|      Parque Central|Montevideo |         Chile|
|          Group 2|      Parque Central|Montevideo |    Yugoslavia|
|          Group 4|      Parque Central|Montevideo |           USA|
|          Group 1|  Estadio Centenario|Montevideo |     Argentina|
|          Group 2|  Estadio Centenario|Montevideo |        Brazil|
|          Group 3|  Estadio Centenario|Montevideo |       Uruguay|
|          Group 1|  Estadio Centenario|Montevideo |     Argentina|
|      Semi-finals|  Estadio Centenario|Montevideo |     Argentina|
|      Semi-finals|  Estadio Centenario|Montevideo |       Uruguay|
|            Final|  Estadio Centenario|Montevideo |       Uruguay|
|Preliminary round|Stadio Benito Mus...|     Turin |       Austria|
|Preliminary round|   Giorgio Ascarelli|    Naples |       Hungary|
|Preliminary round|            San Siro|     Milan |   Switzerland|
|Preliminary round|            Littorale|   Bologna |        Sweden|
```

Query 3: Correlated subquery for finding attendance and max_attendance for a stadium

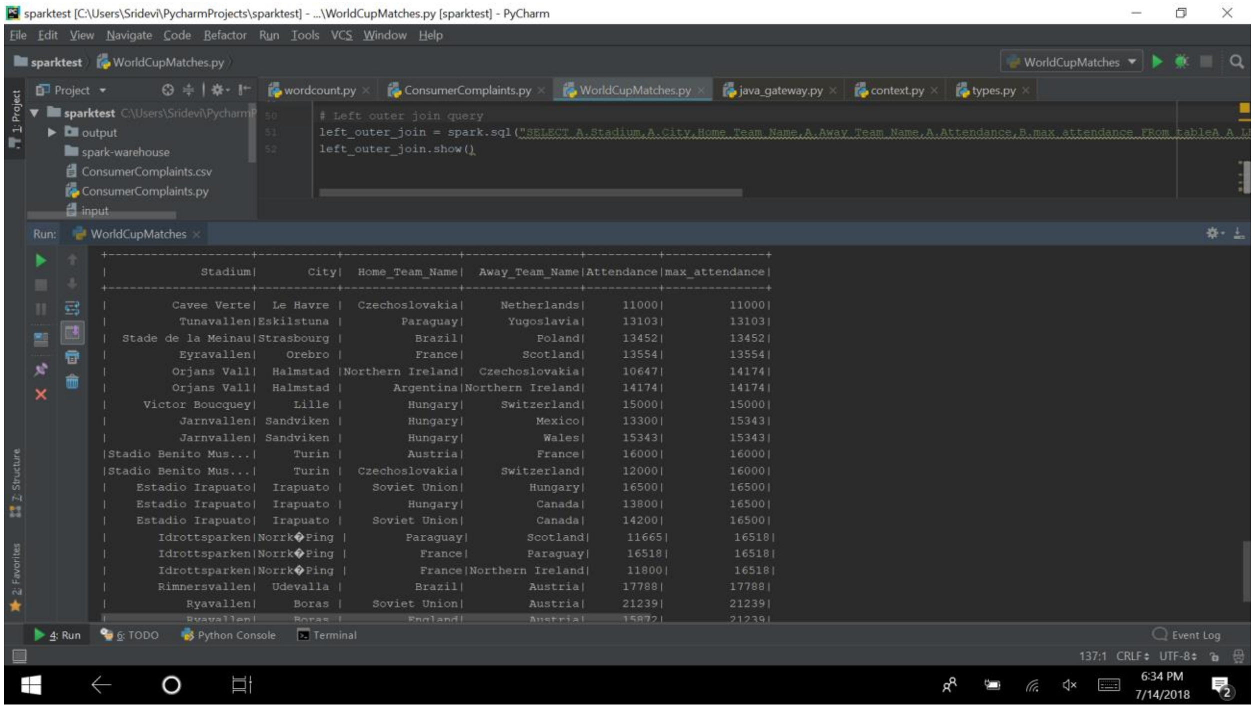


sparktest [C:\Users\Sridevi\PycharmProjects\sparktest] - ...\WorldCupMatches.py [sparktest] - PyCharm

```
# correleated subquery for finding attendance and max_attendance for a stadium
df.createOrReplaceTempView("tableA")
df.createOrReplaceTempView("tableB")
Max_Attendance = spark.sql("SELECT A.Stage,A.Stadium,A.City,A.Home_Team_Name,A.Away_Team_Name,A.Attendance,(SELECT MAX(Attendance
Max_Attendance.show()
```

```
+-----------------+--------------------+-----------+-----------------+-----------------+----------+--------------+
|            Stage|             Stadium|       City|   Home_Team_Name|   Away_Team_Name|Attendance|max_attendance|
+-----------------+--------------------+-----------+-----------------+-----------------+----------+--------------+
|      First round|         Cavee Verte|  Le Havre |   Czechoslovakia|      Netherlands|     11000|         11000|
|          Group 2|          Tunavallen|Eskilstuna |         Paraguay|       Yugoslavia|     13103|         13103|
|      First round| Stade de la Meinau|Strasbourg |           Brazil|           Poland|     13452|         13452|
|          Group 2|          Eyravallen|    Orebro |           France|         Scotland|     13554|         13554|
|          Group 1|          Orjans Vall|  Halmstad |Northern Ireland|   Czechoslovakia|     10647|         14174|
|          Group 1|          Orjans Vall|  Halmstad |        Argentina|Northern Ireland|     14174|         14174|
|   Quarter-finals|      Victor Boucquey|     Lille |          Hungary|      Switzerland|     15000|         15000|
|          Group 3|          Jarnvallen| Sandviken |          Hungary|           Mexico|     13300|         15343|
|          Group 3|          Jarnvallen| Sandviken |          Hungary|            Wales|     15343|         15343|
|Preliminary round|Stadio Benito Mus...|     Turin |          Austria|           France|     16000|         16000|
|   Quarter-finals|Stadio Benito Mus...|     Turin |   Czechoslovakia|      Switzerland|     12000|         16000|
|          Group C|     Estadio Irapuato|  Irapuato |     Soviet Union|          Hungary|     16500|         16500|
|          Group C|     Estadio Irapuato|  Irapuato |          Hungary|           Canada|     13800|         16500|
|          Group C|     Estadio Irapuato|  Irapuato |     Soviet Union|           Canada|     14200|         16500|
|          Group 2|        Idrottsparken|Norrk�Ping |         Paraguay|         Scotland|     11665|         16518|
|          Group 2|        Idrottsparken|Norrk�Ping |           France|         Paraguay|     16518|         16518|
|   Quarter-finals|        Idrottsparken|Norrk�Ping |           France|Northern Ireland|     11800|         16518|
|          Group 4|        Rimnersvallen|  Udevalla |           Brazil|          Austria|     17788|         17788|
|          Group 4|          Ryavallen|     Boras |     Soviet Union|          Austria|     21239|         21239|
```

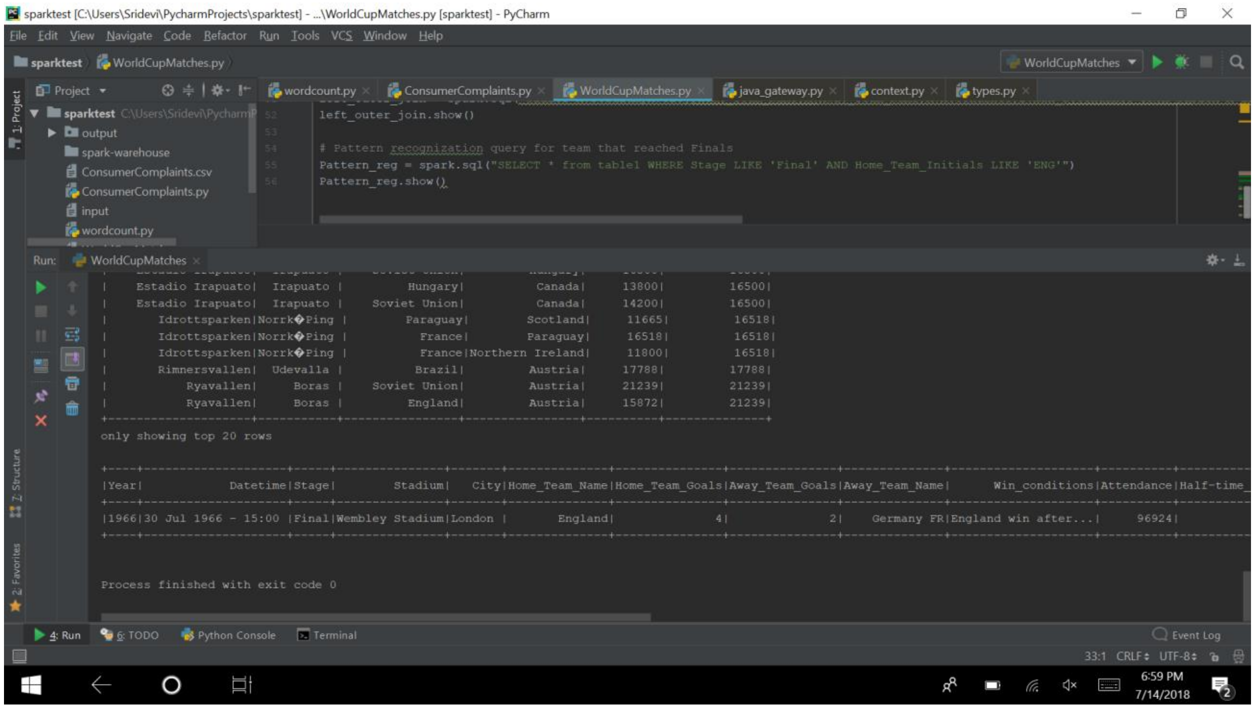
Query 4: Left outer join query



```
# Left outer join query
left_outer_join = spark.sql("SELECT A.Stadium,A.City,Home_Team_Name,A.Away_Team_Name,A.Attendance,B.max_attendance FROm tableA A LE
left_outer_join.show()
```

```
+--------------------+-----------+----------------+-----------------+----------+--------------+
|             Stadium|       City|  Home_Team_Name|   Away_Team_Name|Attendance|max_attendance|
+--------------------+-----------+----------------+-----------------+----------+--------------+
|         Cavee Verte|  Le Havre | Czechoslovakia|      Netherlands|     11000|         11000|
|          Tunavallen|Eskilstuna |        Paraguay|      Yugoslavia|     13103|         13103|
|  Stade de la Meinau|Strasbourg |          Brazil|           Poland|     13452|         13452|
|          Eyravallen|    Orebro |          France|         Scotland|     13554|         13554|
|         Orjans Vall|  Halmstad |Northern Ireland|   Czechoslovakia|     10647|         14174|
|         Orjans Vall|  Halmstad |       Argentina|Northern Ireland|     14174|         14174|
|      Victor Boucquey|     Lille |         Hungary|      Switzerland|     15000|         15000|
|          Jarnvallen| Sandviken |         Hungary|           Mexico|     13300|         15343|
|          Jarnvallen| Sandviken |         Hungary|            Wales|     15343|         15343|
|Stadio Benito Mus...|     Turin |         Austria|           France|     16000|         16000|
|Stadio Benito Mus...|     Turin |  Czechoslovakia|      Switzerland|     12000|         16000|
|    Estadio Irapuato|  Irapuato |    Soviet Union|          Hungary|     16500|         16500|
|    Estadio Irapuato|  Irapuato |         Hungary|           Canada|     13800|         16500|
|    Estadio Irapuato|  Irapuato |    Soviet Union|           Canada|     14200|         16500|
|       Idrottsparken|Norrk�Ping |        Paraguay|         Scotland|      11665|         16518|
|       Idrottsparken|Norrk�Ping |          France|         Paraguay|      16518|         16518|
|       Idrottsparken|Norrk�Ping |          France|Northern Ireland|      11800|         16518|
|       Rimnersvallen|  Udevalla |          Brazil|          Austria|     17788|         17788|
|           Ryavallen|     Boras |    Soviet Union|          Austria|     21239|         21239|
|           Ryavallen|     Boras |         England|          Austria|     15872|         21239|
```

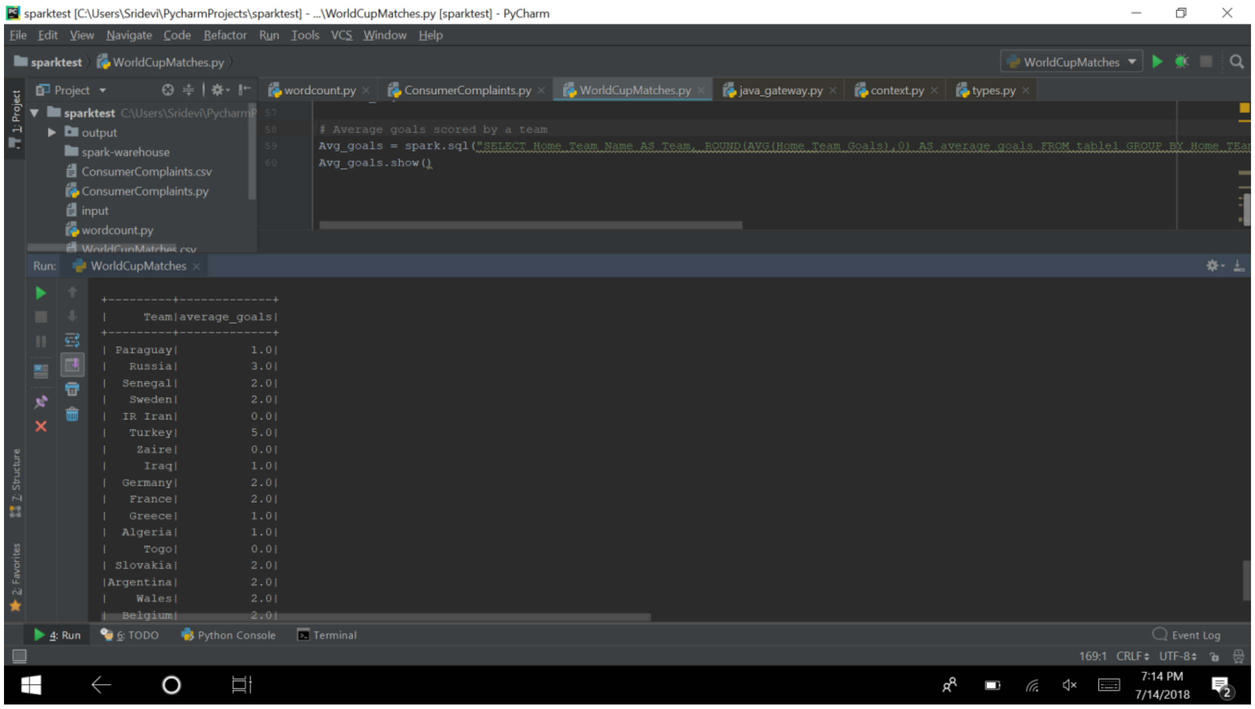
Query 5: Pattern recognization query for team that reached Finals



```
left_outer_join.show()

# Pattern recognization query for team that reached Finals
Pattern_reg = spark.sql("SELECT * from table1 WHERE Stage LIKE 'Final' AND Home_Team_Initials LIKE 'ENG'")
Pattern_reg.show()
```

```
|    Estadio Irapuato|  Irapuato |         Hungary|           Canada|     13800|         16500|
|    Estadio Irapuato|  Irapuato |    Soviet Union|           Canada|     14200|         16500|
|       Idrottsparken|Norrk�Ping |        Paraguay|         Scotland|      11665|         16518|
|       Idrottsparken|Norrk�Ping |          France|         Paraguay|      16518|         16518|
|       Idrottsparken|Norrk�Ping |          France|Northern Ireland|      11800|         16518|
|       Rimnersvallen|  Udevalla |          Brazil|          Austria|     17788|         17788|
|           Ryavallen|     Boras |    Soviet Union|          Austria|     21239|         21239|
|           Ryavallen|     Boras |         England|          Austria|     15872|         21239|
+--------------------+-----------+----------------+-----------------+----------+--------------+
only showing top 20 rows

+----+--------------------+-----+---------------+-------+--------------+---------------+---------------+--------------+--------------------+----------+---------
|Year|            Datetime|Stage|        Stadium|   City|Home_Team_Name|Home_Team_Goals|Away_Team_Goals|Away_Team_Name|      Win_conditions|Attendance|Half-time_
+----+--------------------+-----+---------------+-------+--------------+---------------+---------------+--------------+--------------------+----------+---------
|1966|30 Jul 1966 - 15:00 |Final|Wembley Stadium|London |       England|              4|              2|    Germany FR|England win after...|     96924|
+----+--------------------+-----+---------------+-------+--------------+---------------+---------------+--------------+--------------------+----------+---------


Process finished with exit code 0
```

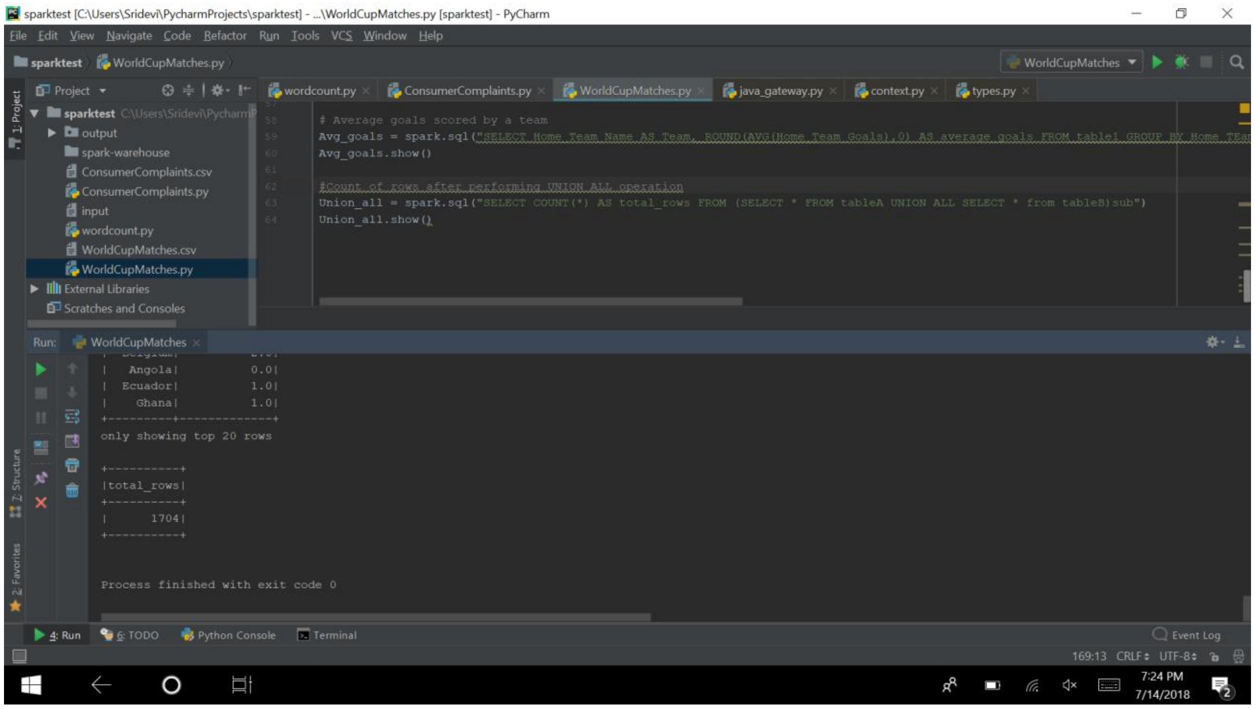
Query 6: Average goals scored by a team



```
# Average goals scored by a team
Avg_goals = spark.sql("SELECT Home Team Name AS Team, ROUND(AVG(Home Team Goals),0) AS average_goals FROM table1 GROUP BY Home TEam
Avg_goals.show()
```

```
+---------+-------------+
|     Team|average_goals|
+---------+-------------+
| Paraguay|          1.0|
|   Russia|          3.0|
|  Senegal|          2.0|
|   Sweden|          2.0|
|  IR Iran|          0.0|
|   Turkey|          5.0|
|    Zaire|          0.0|
|     Iraq|          1.0|
|  Germany|          2.0|
|   France|          2.0|
|   Greece|          1.0|
|  Algeria|          1.0|
|     Togo|          0.0|
| Slovakia|          2.0|
|Argentina|          2.0|
|    Wales|          2.0|
|  Belgium|          2.0|
```

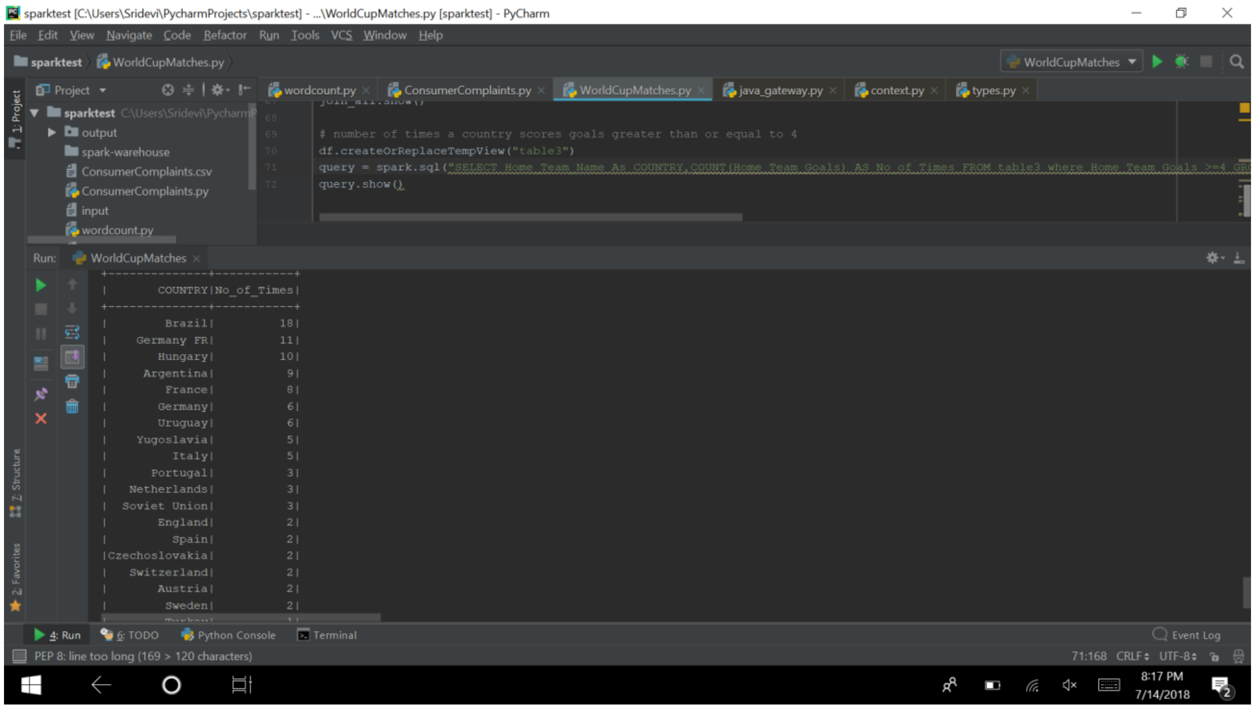
Query 7: Count of rows after performing UNION ALL operation



```
# Average goals scored by a team
Avg_goals = spark.sql("SELECT Home Team Name AS Team, ROUND(AVG(Home Team Goals),0) AS average_goals FROM table1 GROUP BY Home TEam
Avg_goals.show()

#Count of rows after performing UNION ALL operation
Union_all = spark.sql("SELECT COUNT(*) AS total_rows FROM (SELECT * FROM tableA UNION ALL SELECT * from tableB)sub")
Union_all.show()
```

```
|   Angola|          0.0|
|  Ecuador|          1.0|
|    Ghana|          1.0|
+---------+-------------+
only showing top 20 rows

+----------+
|total_rows|
+----------+
|      1704|
+----------+


Process finished with exit code 0
```

Query 8: Number of times a country scores goals greater than or equal to 4

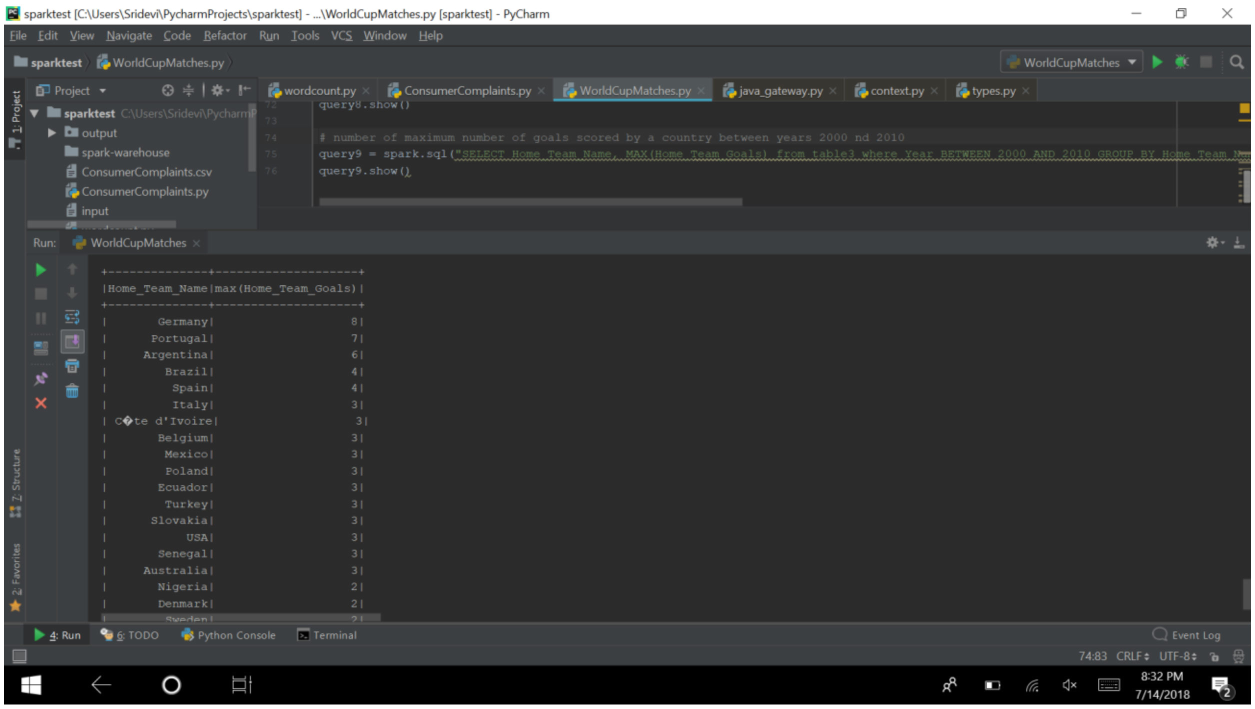


```
join_all.show()

# number of times a country scores goals greater than or equal to 4
df.createOrReplaceTempView("table3")
query = spark.sql("SELECT Home_Team_Name As COUNTRY,COUNT(Home_Team_Goals) AS No_of_Times FROM table3 where Home_Team_Goals >=4 GR
query.show()
```

```
+--------------+-----------+
|       COUNTRY|No_of_Times|
+--------------+-----------+
|        Brazil|         18|
|    Germany FR|         11|
|       Hungary|         10|
|     Argentina|          9|
|        France|          8|
|       Germany|          6|
|       Uruguay|          6|
|    Yugoslavia|          5|
|         Italy|          5|
|      Portugal|          3|
|   Netherlands|          3|
|  Soviet Union|          3|
|       England|          2|
|         Spain|          2|
|Czechoslovakia|          2|
|   Switzerland|          2|
|       Austria|          2|
|        Sweden|          2|
```

PEP 8: line too long (169 > 120 characters)

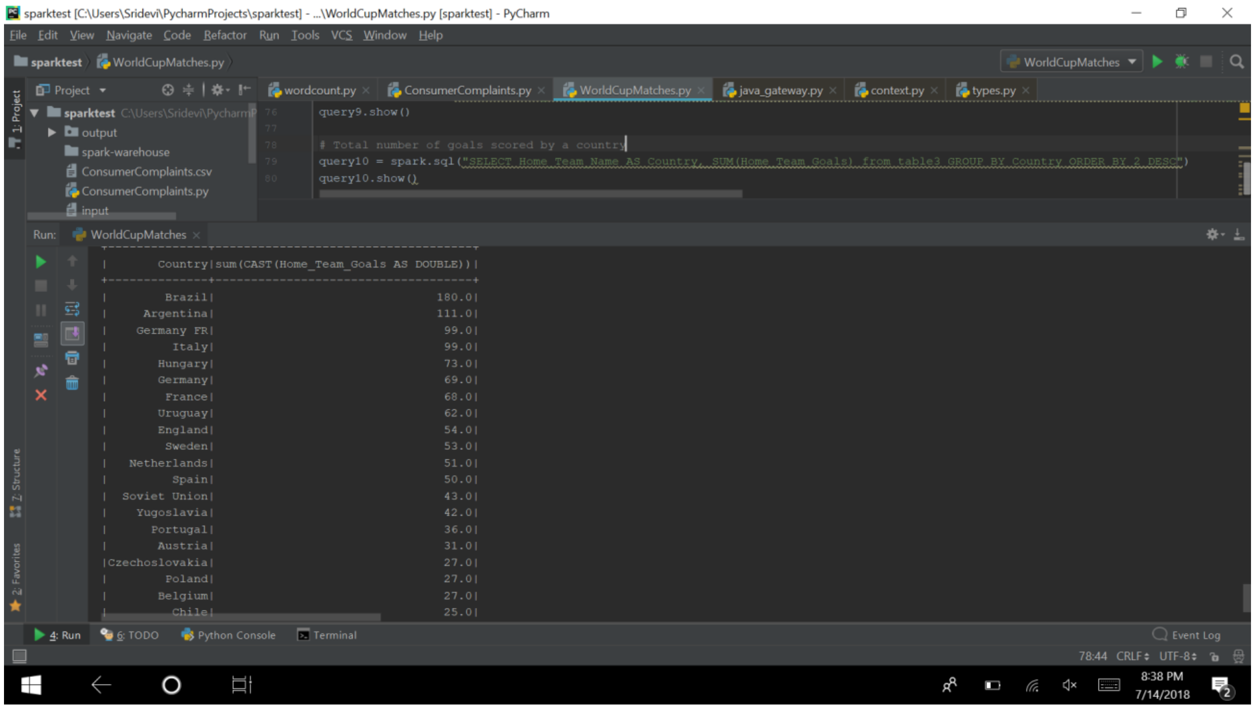
Query 9: Number of maximum number of goals scored by a country between years 2000 nd 2010



```
query8.show()

# number of maximum number of goals scored by a country between years 2000 nd 2010
query9 = spark.sql("SELECT Home_Team_Name, MAX(Home_Team_Goals) from table3 where Year BETWEEN 2000 AND 2010 GROUP BY Home_Team_Na
query9.show()
```

```
+--------------+--------------------+
|Home_Team_Name|max(Home_Team_Goals)|
+--------------+--------------------+
|       Germany|                   8|
|      Portugal|                   7|
|     Argentina|                   6|
|        Brazil|                   4|
|         Spain|                   4|
|         Italy|                   3|
| C�te d'Ivoire|                   3|
|       Belgium|                   3|
|        Mexico|                   3|
|        Poland|                   3|
|       Ecuador|                   3|
|        Turkey|                   3|
|      Slovakia|                   3|
|           USA|                   3|
|       Senegal|                   3|
|     Australia|                   3|
|       Nigeria|                   2|
|       Denmark|                   2|
```

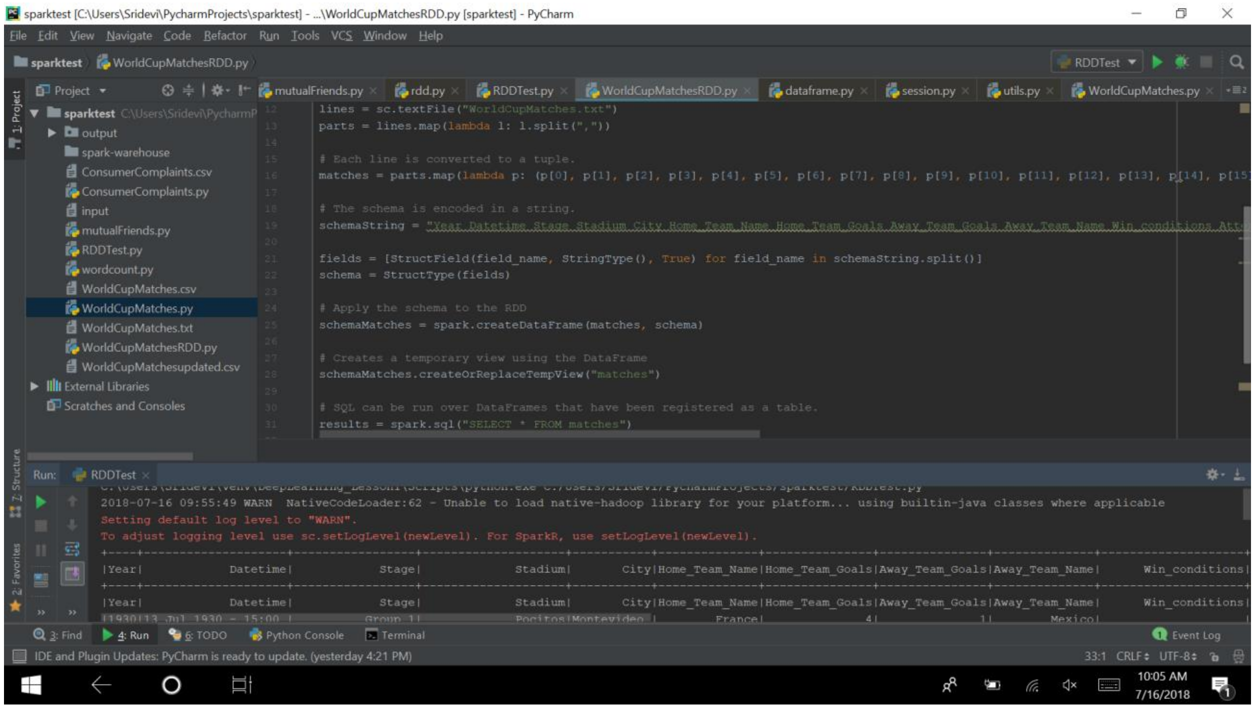
Query 10: Total number of goals scored by a country



```
query9.show()

# Total number of goals scored by a country
query10 = spark.sql("SELECT Home_Team_Name AS Country, SUM(Home_Team_Goals) from table3 GROUP BY Country ORDER BY 2 DESC")
query10.show()
```

```
+-------------+---------------------------------+
|      Country|sum(CAST(Home_Team_Goals AS DOUBLE))|
+-------------+---------------------------------+
|       Brazil|                            180.0|
|    Argentina|                            111.0|
|   Germany FR|                             99.0|
|        Italy|                             99.0|
|      Hungary|                             73.0|
|      Germany|                             69.0|
|       France|                             68.0|
|      Uruguay|                             62.0|
|      England|                             54.0|
|       Sweden|                             53.0|
|  Netherlands|                             51.0|
|        Spain|                             50.0|
| Soviet Union|                             43.0|
|   Yugoslavia|                             42.0|
|     Portugal|                             36.0|
|      Austria|                             31.0|
|Czechoslovakia|                            27.0|
|       Poland|                             27.0|
|      Belgium|                             27.0|
|        Chile|                             25.0|
```

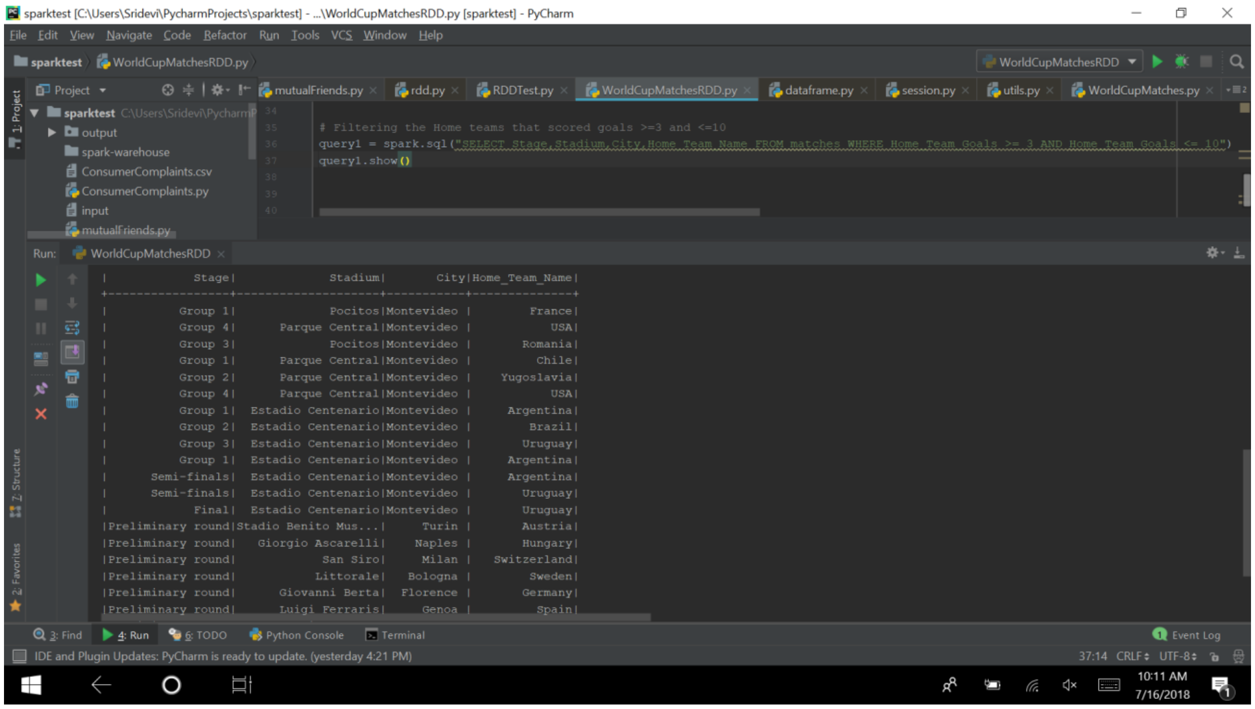
Spark RDD's: Apply the schema to the RDD



```
lines = sc.textFile("WorldCupMatches.txt")
parts = lines.map(lambda l: l.split(","))

# Each line is converted to a tuple.
matches = parts.map(lambda p: (p[0], p[1], p[2], p[3], p[4], p[5], p[6], p[7], p[8], p[9], p[10], p[11], p[12], p[13], p[14], p[15

# The schema is encoded in a string.
schemaString = "Year Datetime Stage Stadium City Home_Team_Name Home_Team_Goals Away_Team_Goals Away_Team_Name Win_conditions Att

fields = [StructField(field_name, StringType(), True) for field_name in schemaString.split()]
schema = StructType(fields)

# Apply the schema to the RDD
schemaMatches = spark.createDataFrame(matches, schema)

# Creates a temporary view using the DataFrame
schemaMatches.createOrReplaceTempView("matches")

# SQL can be run over DataFrames that have been registered as a table.
results = spark.sql("SELECT * FROM matches")
```

```
C:\Users\Sridevi\venv\DeepLearning_Lesson1\Scripts\python.exe C:/Users/Sridevi/PycharmProjects/sparktest/RDDTest.py
2018-07-16 09:55:49 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
+----+--------------------+------------------+--------------------+-------------+--------------+---------------+---------------+--------------+------------------
|Year|            Datetime|             Stage|             Stadium|         City|Home_Team_Name|Home_Team_Goals|Away_Team_Goals|Away_Team_Name|      Win_conditions|
+----+--------------------+------------------+--------------------+-------------+--------------+---------------+---------------+--------------+------------------
|Year|            Datetime|             Stage|             Stadium|         City|Home_Team_Name|Home_Team_Goals|Away_Team_Goals|Away_Team_Name|      Win_conditions|
|1930|13 Jul 1930 - 15:00 |           Group 1|             Pocitos|Montevideo |        France|              4|              1|        Mexico|
```

Query 1: Filtering the Home teams that scored goals >=3 and <=10 with RDD



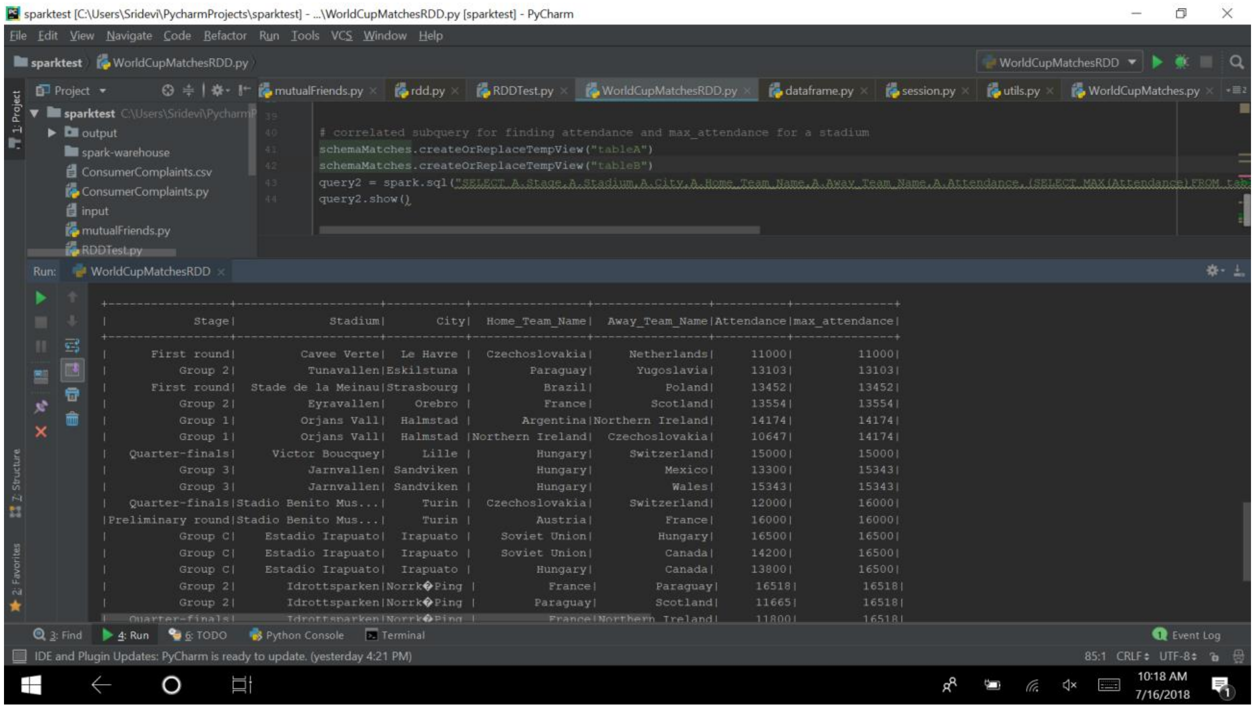
sparktest [C:\Users\Sridevi\PycharmProjects\sparktest] - ...\WorldCupMatchesRDD.py [sparktest] - PyCharm

```
# Filtering the Home teams that scored goals >=3 and <=10
query1 = spark.sql("SELECT Stage,Stadium,City,Home_Team_Name FROM matches WHERE Home_Team_Goals >= 3 AND Home_Team_Goals <= 10")
query1.show()
```

```
|           Stage|             Stadium|       City|Home_Team_Name|
+----------------+--------------------+-----------+--------------+
|         Group 1|             Pocitos|Montevideo |        France|
|         Group 4|      Parque Central|Montevideo |           USA|
|         Group 3|             Pocitos|Montevideo |       Romania|
|         Group 1|      Parque Central|Montevideo |         Chile|
|         Group 2|      Parque Central|Montevideo |    Yugoslavia|
|         Group 4|      Parque Central|Montevideo |           USA|
|         Group 1|  Estadio Centenario|Montevideo |     Argentina|
|         Group 2|  Estadio Centenario|Montevideo |        Brazil|
|         Group 3|  Estadio Centenario|Montevideo |       Uruguay|
|         Group 1|  Estadio Centenario|Montevideo |     Argentina|
|     Semi-finals|  Estadio Centenario|Montevideo |     Argentina|
|     Semi-finals|  Estadio Centenario|Montevideo |       Uruguay|
|           Final|  Estadio Centenario|Montevideo |       Uruguay|
|Preliminary round|Stadio Benito Mus...|     Turin |       Austria|
|Preliminary round|   Giorgio Ascarelli|    Naples |       Hungary|
|Preliminary round|            San Siro|     Milan |   Switzerland|
|Preliminary round|           Littorale|   Bologna |        Sweden|
|Preliminary round|      Giovanni Berta|  Florence |       Germany|
|Preliminary round|      Luigi Ferraris|     Genoa |         Spain|
```

IDE and Plugin Updates: PyCharm is ready to update. (yesterday 4:21 PM)

Query 2: Correlated subquery for finding attendance and max_attendance for a stadium with RDD



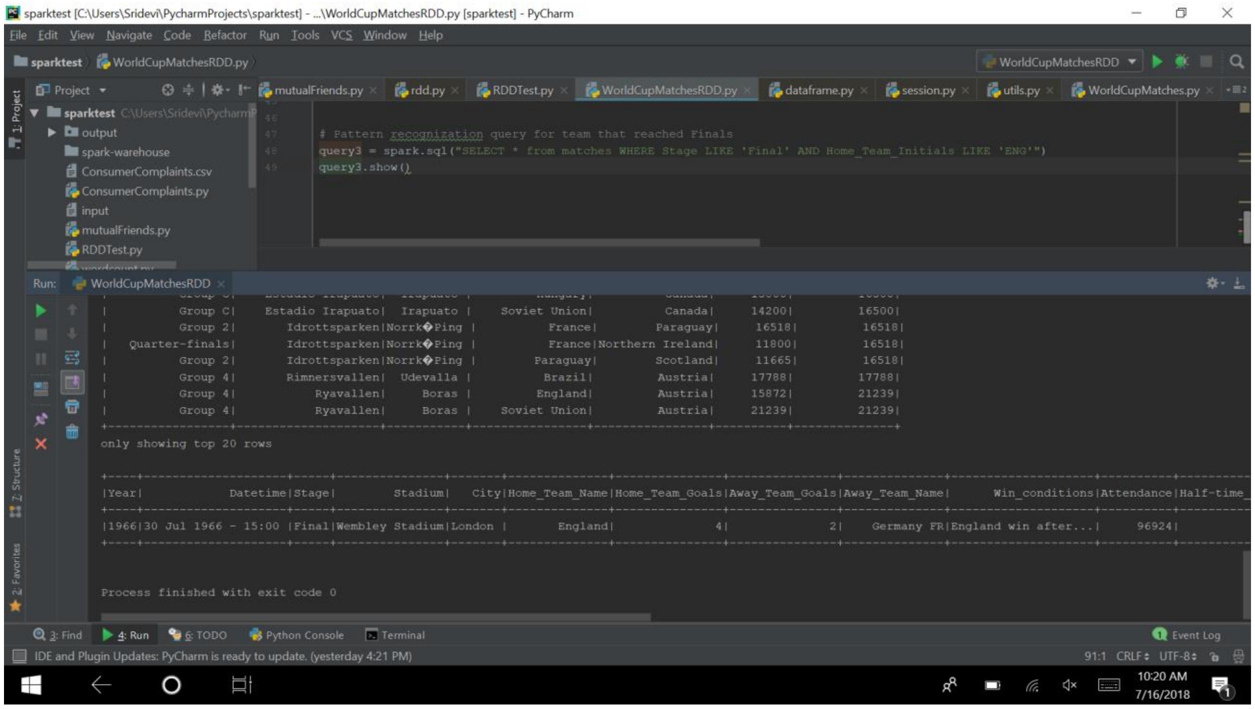
sparktest [C:\Users\Sridevi\PycharmProjects\sparktest] - ...\WorldCupMatchesRDD.py [sparktest] - PyCharm

```
# correlated subquery for finding attendance and max_attendance for a stadium
schemaMatches.createOrReplaceTempView("tableA")
schemaMatches.createOrReplaceTempView("tableB")
query2 = spark.sql("SELECT A.Stage,A.Stadium,A.City,A.Home_Team_Name,A.Away_Team_Name,A.Attendance,(SELECT MAX(Attendance)FROM tab
query2.show()
```

```
+-----------------+--------------------+-----------+----------------+----------------+----------+--------------+
|            Stage|             Stadium|       City|  Home_Team_Name|  Away_Team_Name|Attendance|max_attendance|
+-----------------+--------------------+-----------+----------------+----------------+----------+--------------+
|      First round|         Cavee Verte|  Le Havre |  Czechoslovakia|     Netherlands|     11000|         11000|
|          Group 2|          Tunavallen|Eskilstuna |        Paraguay|      Yugoslavia|     13103|         13103|
|      First round| Stade de la Meinau|Strasbourg |          Brazil|          Poland|     13452|         13452|
|          Group 2|          Eyravallen|    Orebro |          France|        Scotland|     13554|         13554|
|          Group 1|         Orjans Vall|  Halmstad |       Argentina|Northern Ireland|     14174|         14174|
|          Group 1|         Orjans Vall|  Halmstad |Northern Ireland|  Czechoslovakia|     10647|         14174|
|   Quarter-finals|      Victor Boucquey|     Lille |         Hungary|     Switzerland|     15000|         15000|
|          Group 3|          Jarnvallen| Sandviken |         Hungary|          Mexico|     13300|         15343|
|          Group 3|          Jarnvallen| Sandviken |         Hungary|           Wales|     15343|         15343|
|   Quarter-finals|Stadio Benito Mus...|     Turin |  Czechoslovakia|     Switzerland|     12000|         16000|
|Preliminary round|Stadio Benito Mus...|     Turin |         Austria|          France|     16000|         16000|
|          Group C|    Estadio Irapuato|  Irapuato |    Soviet Union|         Hungary|     16500|         16500|
|          Group C|    Estadio Irapuato|  Irapuato |    Soviet Union|          Canada|     14200|         16500|
|          Group C|    Estadio Irapuato|  Irapuato |         Hungary|          Canada|     13800|         16500|
|          Group 2|       Idrottsparken|Norrk�Ping |          France|        Paraguay|     16518|         16518|
|          Group 2|       Idrottsparken|Norrk�Ping |        Paraguay|        Scotland|     11665|         16518|
|   Quarter-finals|       Idrottsparken|Norrk�Ping |          France|Northern Ireland|     11800|         16518|
```

IDE and Plugin Updates: PyCharm is ready to update. (yesterday 4:21 PM)

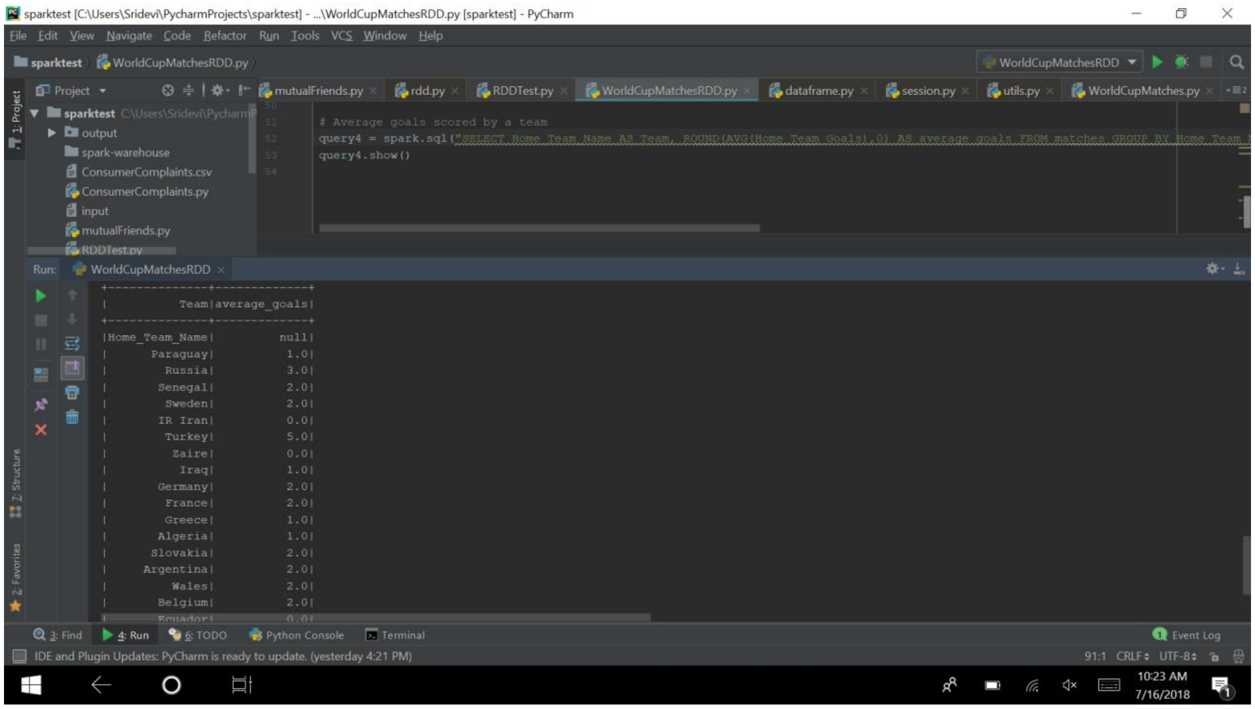
Query 3: Pattern recognition query for team that reached Finals with RDD



```
# Pattern recognization query for team that reached Finals
query3 = spark.sql("SELECT * from matches WHERE Stage LIKE 'Final' AND Home_Team_Initials LIKE 'ENG'")
query3.show()
```

```
|         Group C|   Estadio Irapuato|  Irapuato |    Soviet Union|          Canada|     14200|          16500|
|         Group 2|      Idrottsparken|Norrk�Ping |          France|        Paraguay|      16518|          16518|
|   Quarter-finals|      Idrottsparken|Norrk�Ping |          France|Northern Ireland|      11800|          16518|
|         Group 2|      Idrottsparken|Norrk�Ping |        Paraguay|        Scotland|      11665|          16518|
|         Group 4|      Rimnersvallen|  Udevalla |          Brazil|         Austria|     17788|          17788|
|         Group 4|          Ryavallen|     Boras |         England|         Austria|     15872|          21239|
|         Group 4|          Ryavallen|     Boras |    Soviet Union|         Austria|     21239|          21239|
+----------------+-------------------+-----------+----------------+----------------+----------+---------------+
only showing top 20 rows

+----+--------------------+-----+---------------+-------+--------------+---------------+---------------+--------------+--------------------+----------+----------
|Year|            Datetime|Stage|        Stadium|   City|Home_Team_Name|Home_Team_Goals|Away_Team_Goals|Away_Team_Name|      Win_conditions|Attendance|Half-time_
+----+--------------------+-----+---------------+-------+--------------+---------------+---------------+--------------+--------------------+----------+----------
|1966|30 Jul 1966 - 15:00 |Final|Wembley Stadium|London |       England|              4|              2|    Germany FR|England win after...|     96924|
+----+--------------------+-----+---------------+-------+--------------+---------------+---------------+--------------+--------------------+----------+----------


Process finished with exit code 0
```

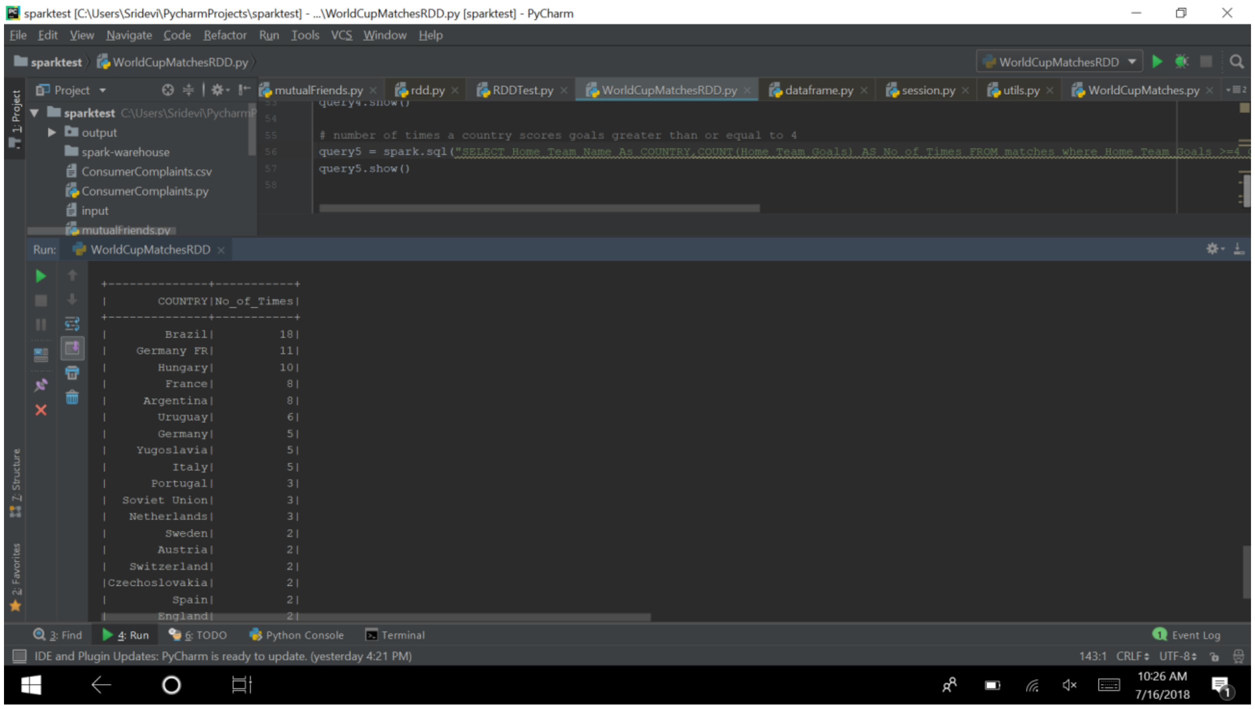
Query 4: Average goals scored by a team with RDD



```
# Average goals scored by a team
query4 = spark.sql("SELECT Home_Team_Name AS Team, ROUND(AVG(Home_Team_Goals),0) AS average_goals FROM matches GROUP BY Home_Team_
query4.show()
```

```
+--------------+-------------+
|          Team|average_goals|
+--------------+-------------+
|Home_Team_Name|         null|
|      Paraguay|          1.0|
|        Russia|          3.0|
|       Senegal|          2.0|
|        Sweden|          2.0|
|       IR Iran|          0.0|
|        Turkey|          5.0|
|         Zaire|          0.0|
|          Iraq|          1.0|
|       Germany|          2.0|
|        France|          2.0|
|        Greece|          1.0|
|       Algeria|          1.0|
|      Slovakia|          2.0|
|     Argentina|          2.0|
|         Wales|          2.0|
|       Belgium|          2.0|
|       Ecuador|          0.0|
```

Query 5: Number of times a country scores goals greater than or equal to 4 with RDD



```
query4.show()

# number of times a country scores goals greater than or equal to 4
query5 = spark.sql("SELECT Home_Team_Name As COUNTRY,COUNT(Home_Team_Goals) AS No_of_Times FROM matches where Home_Team_Goals >=4
query5.show()
```

```
+--------------+-----------+
|       COUNTRY|No_of_Times|
+--------------+-----------+
|        Brazil|         18|
|    Germany FR|         11|
|       Hungary|         10|
|        France|          8|
|     Argentina|          8|
|       Uruguay|          6|
|       Germany|          5|
|    Yugoslavia|          5|
|         Italy|          5|
|      Portugal|          3|
|  Soviet Union|          3|
|   Netherlands|          3|
|        Sweden|          2|
|       Austria|          2|
|   Switzerland|          2|
|Czechoslovakia|          2|
|         Spain|          2|
|       England|          2|
```

Comparison between Spark RDD's and the data frames:

Spark RDD's	Spark Data Frames
RDD is a fault-tolerant collection of elements that can be operated on in-parallel, also we can say RDD is the fundamental data structure of Spark	It is a distributed collection of data. Basically, data is organized into named columns in data frames.
Basically, it is read-only partition collection of records. Moreover, it supports in-memory computations on large clusters in a fault-tolerant manner	Spark also introduced **catalyst optimizer**, along with data frame. To build an extensible query optimizer, it also leverages advanced programming features.
An RDD can come easily handle data with no predefined structure.	In Spark, data frame allows developers to impose a structure onto a distributed data. It also allows higher-level abstraction.
Compile- Time Type Safety RDD- RDD Supports *object-oriented programming* style with compile-time type safety.	If we try to access any column which is not present in the table, then an attribute error may occur at runtime. Data frame will not support compile-time type safety in such case.
Spark does not compute their result right away, it evaluates RDDs lazily	Computation happens only when action appears as Spark evaluates data frame lazily

RDDs of Apache spark offers low-level functionality and control. Whereas datasets offer higher functionality. While data frame offers high-level domain-specific operations, saves space and executes at high speed. Therefore, it increases the efficiency of the system.

Conclusion:

As stated, the above workflow with certain set of parameters is followed in solving the execution by implementing the core and basic concepts Apache Spark.

Source code link:

https://github.com/PragathiThammaneni/Bigdata-Programming--Hadoop-Spark/tree/master/Lab%203

Video Link: Provided in wiki https://youtu.be/31C5DO0zQSE

Wiki Link:
https://github.com/PragathiThammaneni/Bigdata-Programming--Hadoop-Spark/wiki/Lab-3-Assignment