**BIGDATA PROGRAMMING - PROJECT 1**

**CASE-2: Spark DataFrames**

**Team Members and collaboration:**

Roshini varada - Hadoop MapReduce Algorithm

Sarika Reddy Kota - Spark DataFrames

Pallavi Arikatla - Spark streaming task

Zakari, Abdulmuhaymin - Spark Graphx task

**Idea of the project : (Question 5 part 1)**

* To Create Dataframe on given csv file and run spark queries for pattern recognition, topic discussion on DataFrames.
* To Perform any 5 queries in Spark RDD’s and Spark Data Frames.
* Perform 10 intuitive questions in Data set.

**Usage of Project: (Question 5 Part 2)**

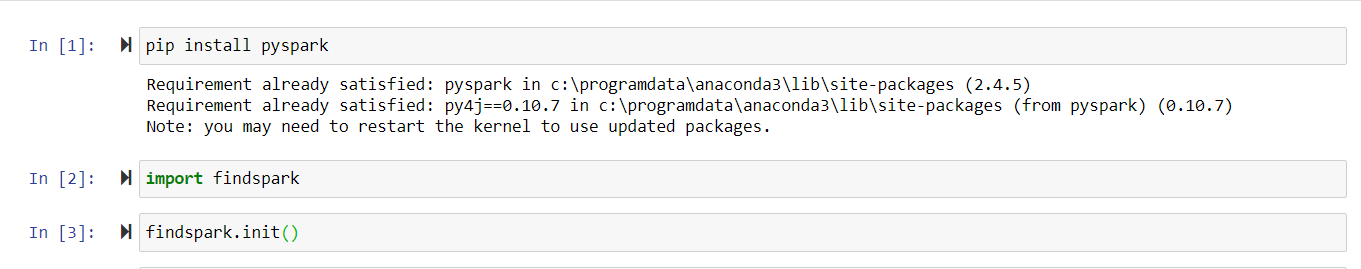
* A Spark DataFrame is a distributed collection of data organized into named columns that provides operations to filter, group, or compute aggregates, and can be used with Spark SQL.
* DataFrames can be constructed from structured data files, existing RDDs, tables in Hive, or external databases.

**Software Required:**

* Python3
* Pyspark
* Jupyter Notebook
* Github

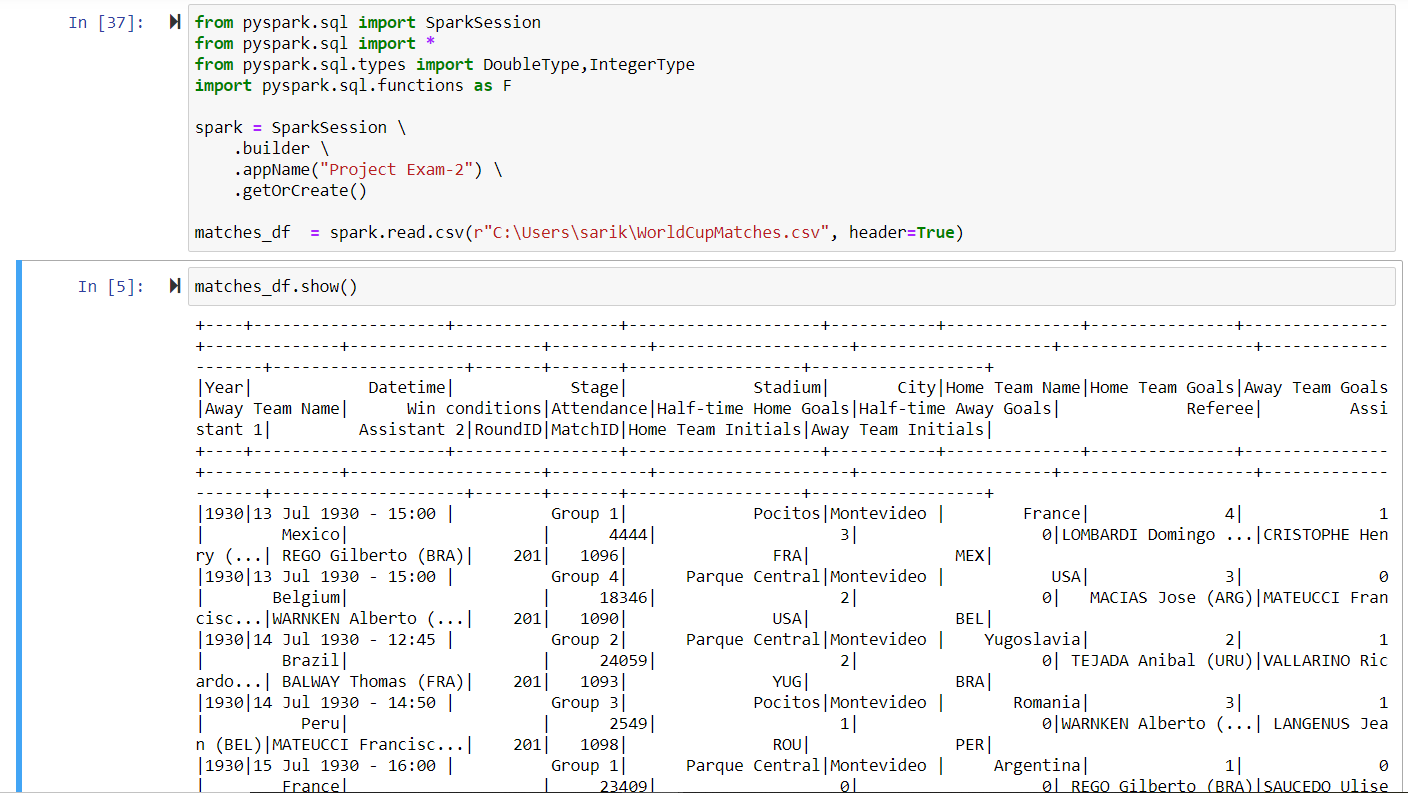
**Implementation: (Question 5 part 3)**

* For the initial setup, Install pyspark, findspark and then initialize the findspark.

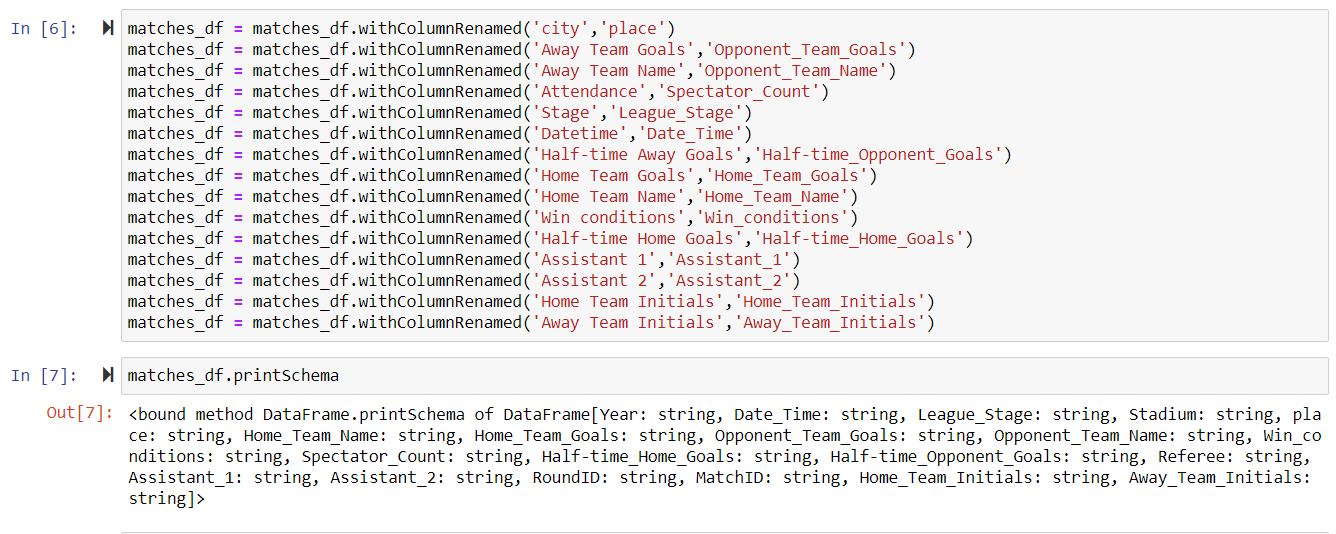


1. **Create a Spark DataFrame using one of datasetsandtry to use all different StructType**.

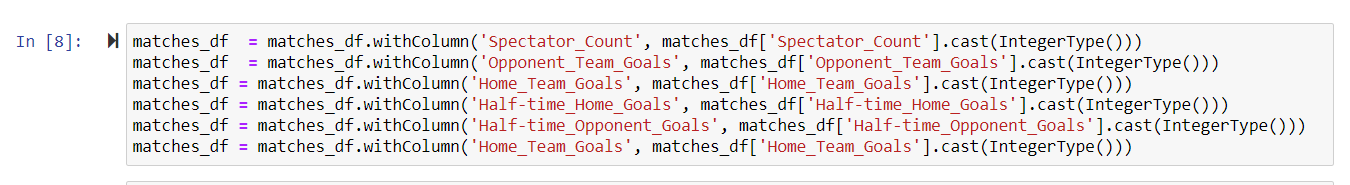
* First Create a data frame using the world cup matches data set and then rename the columns.



* Rename the columns and then printing the schema.



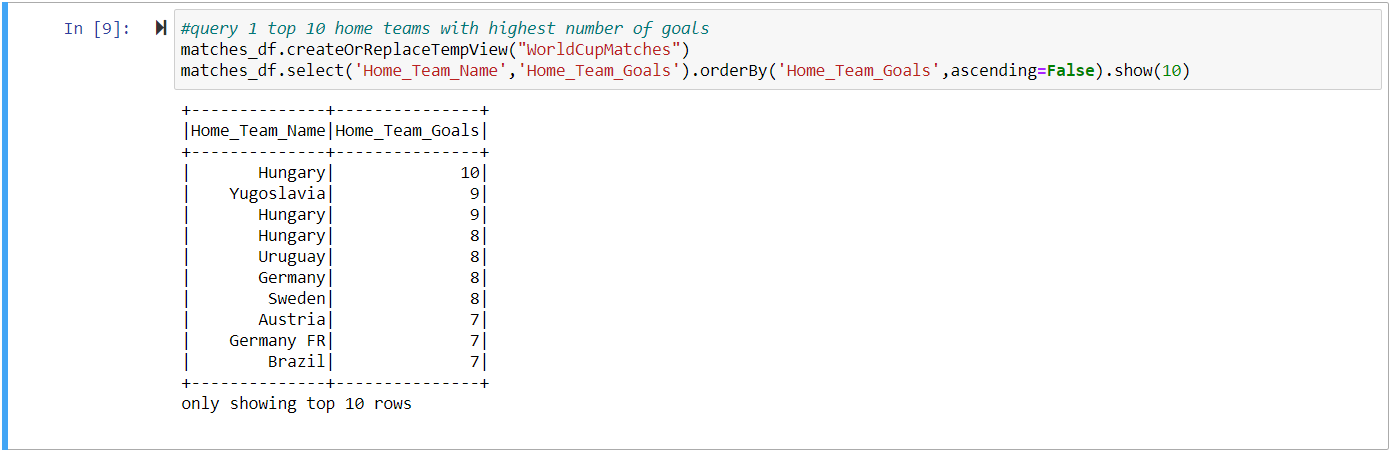
* Now Change the required columns into integer data type



1. **Perform 10 intuitive questions in Dataset**

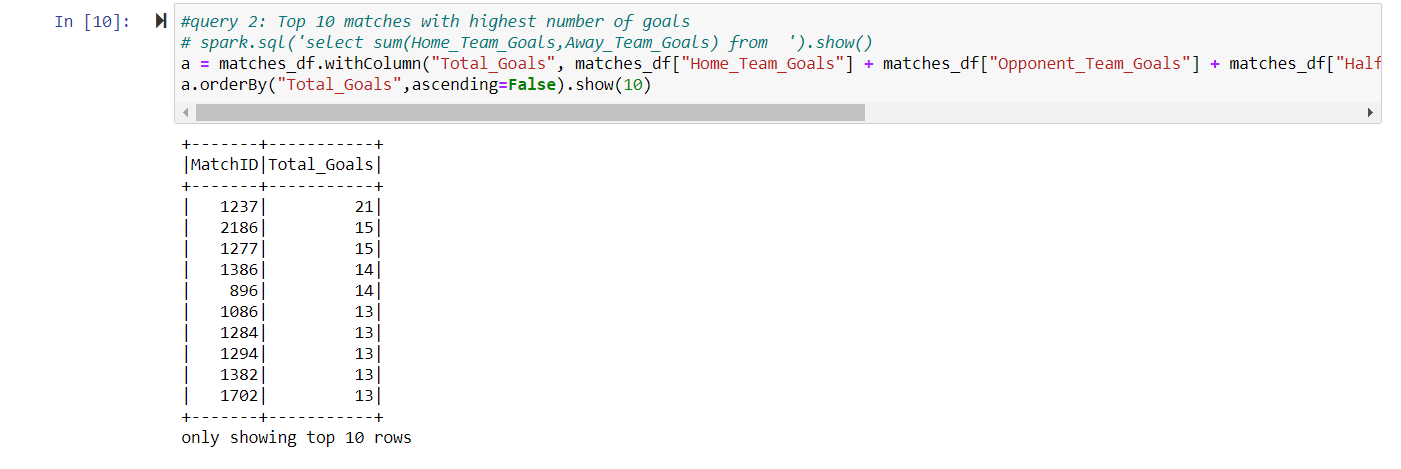
**Query 1:**

To find the top 5 home teams with highest number of goals.



**Query 2:**

To find the top 10 matches with highest number of goals.



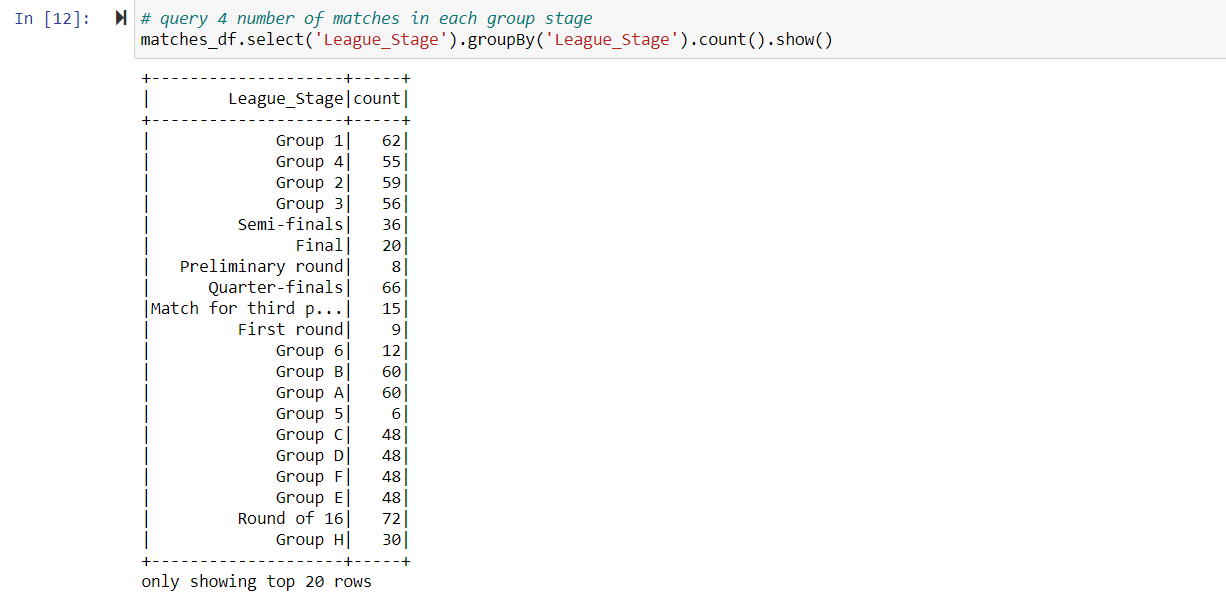
**Query 3:**

To find the number of matches held year wise.



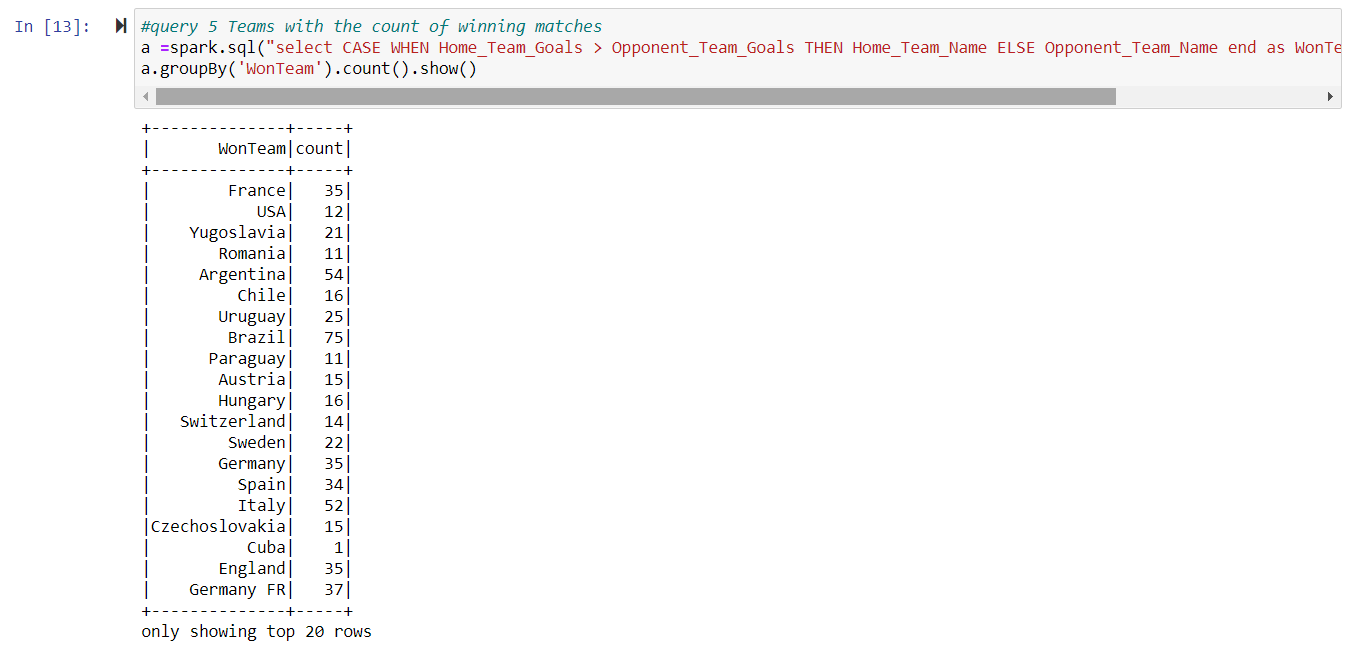
**Query 4:**

To find the number of matches in each group stage.



**Query 5:**

To find the teams with the count of winning matches.



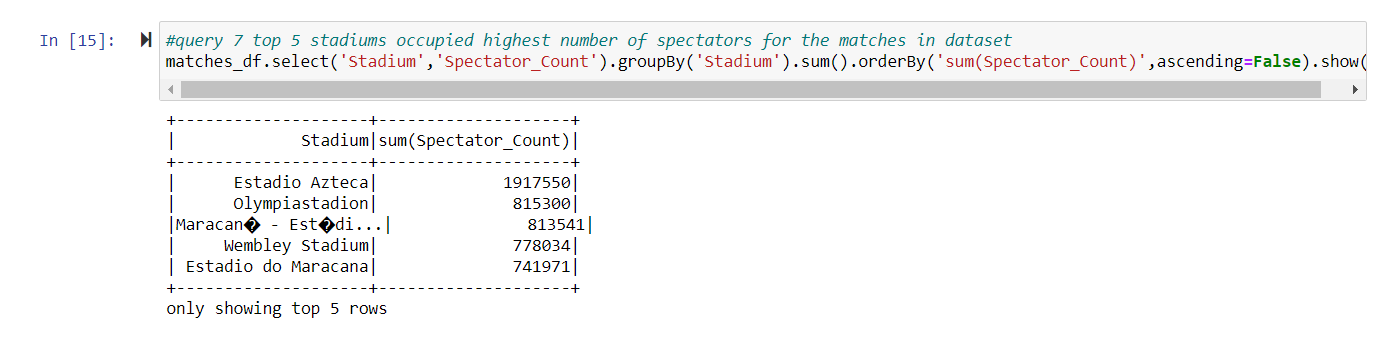
**Query 6:**

To find the matches that held with extra time given.



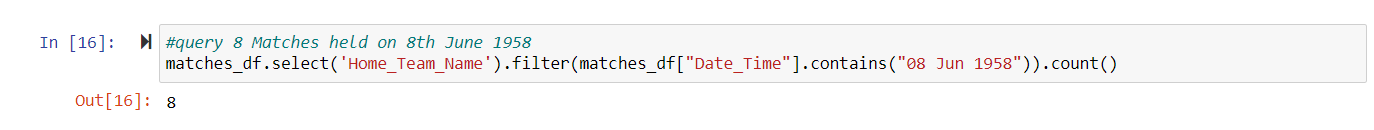
**Query 7:**

To find the top 5 stadiums occupied with the highest number of spectators for the matches.



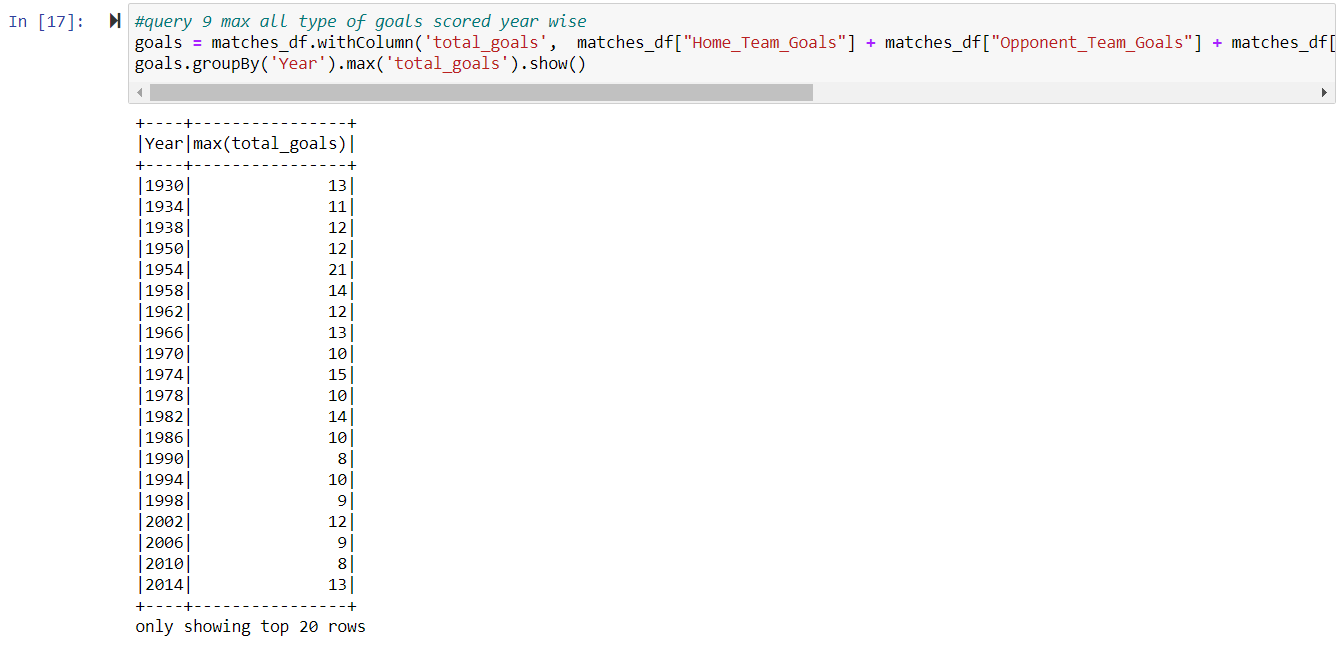
**Query 8:**

To find the matches those held on 8th June 1958.



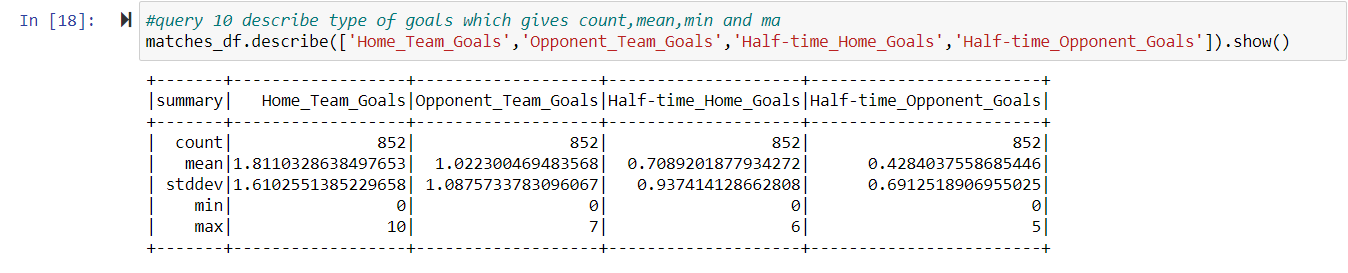
**Query 9:**

To find the maximum all type of goals which are scored year wise.



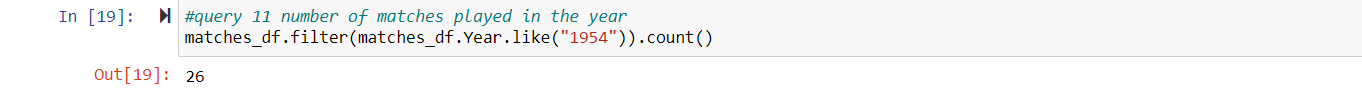
**Query 10:**

To describe type of goals which gives count, mean, min and max.



**Query 11:**

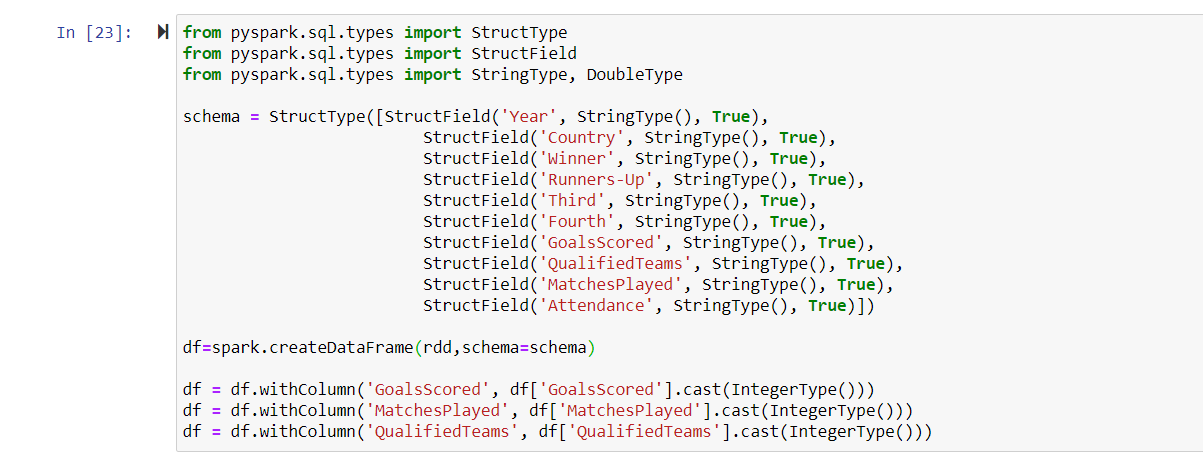
To find the number of matches played in the year 1954.



**c) Perform any 5 queries in Spark RDD’s and Spark Data Frames.**

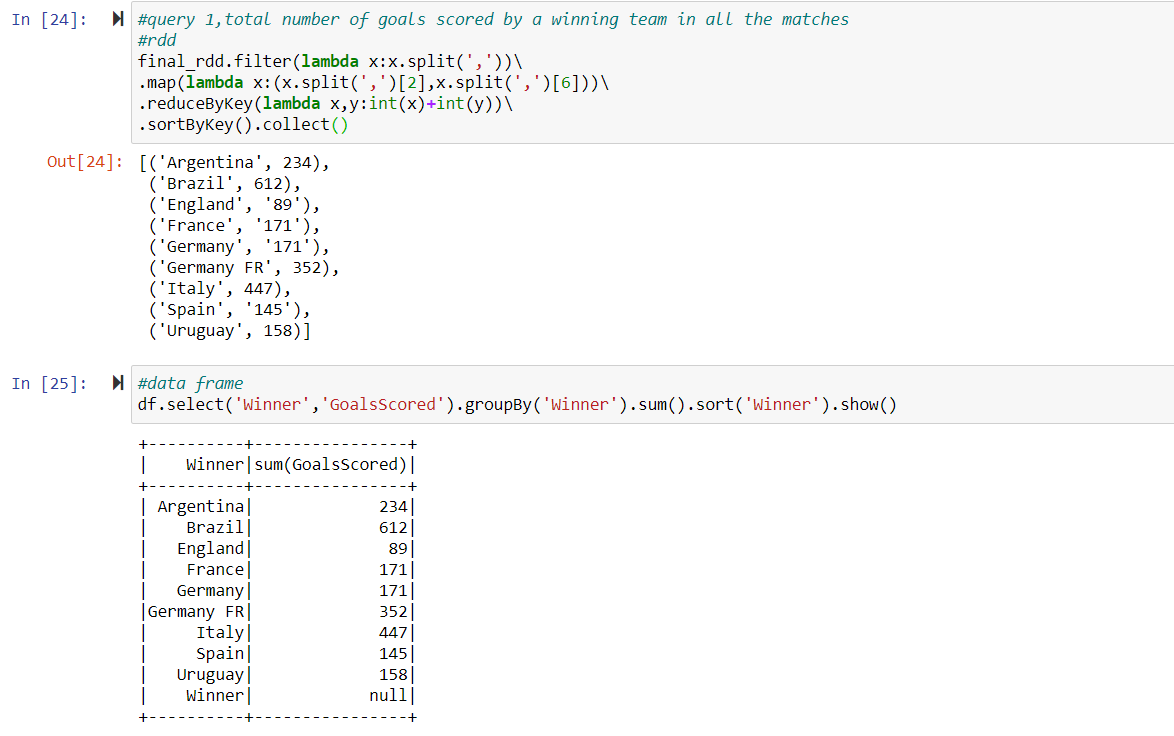
* Create RDD with the data set worldcup matches and ran queries on data frame.





**Query 1:**

To find the total number of goals scored by a winning team in all the matches.



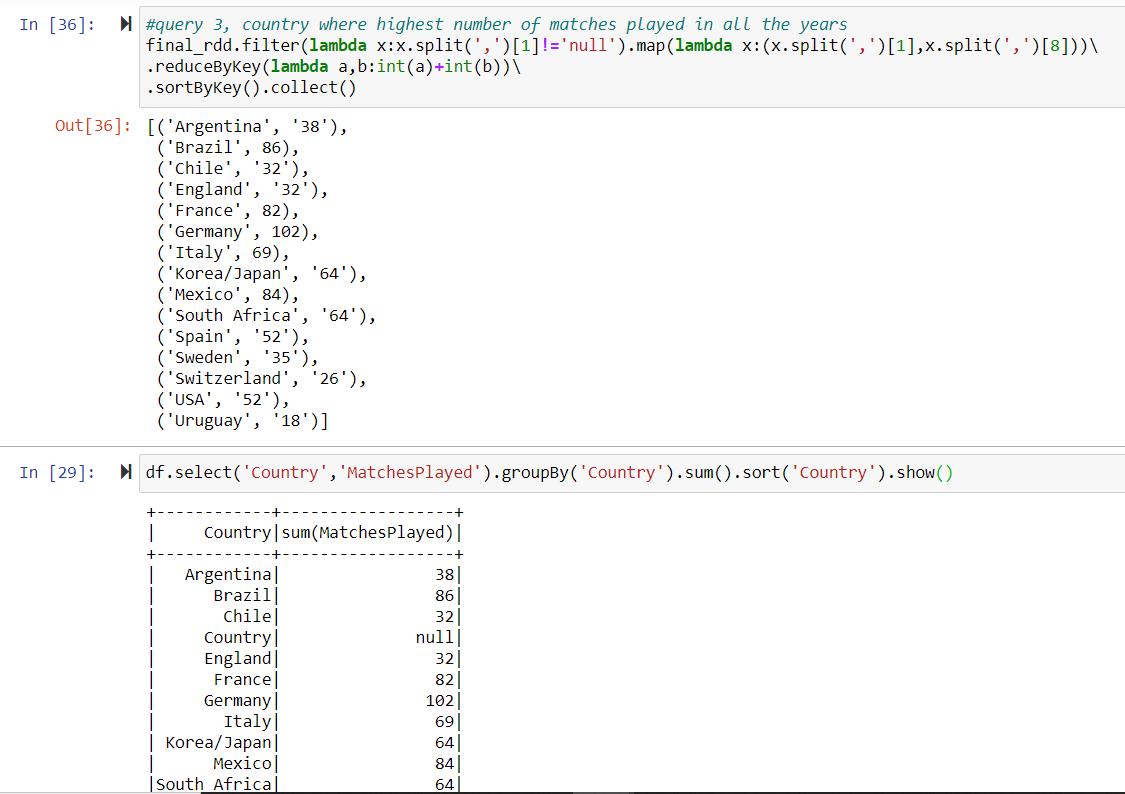
**Query 2:**

To find the number of times each country has won the cup.



**Query 3:**

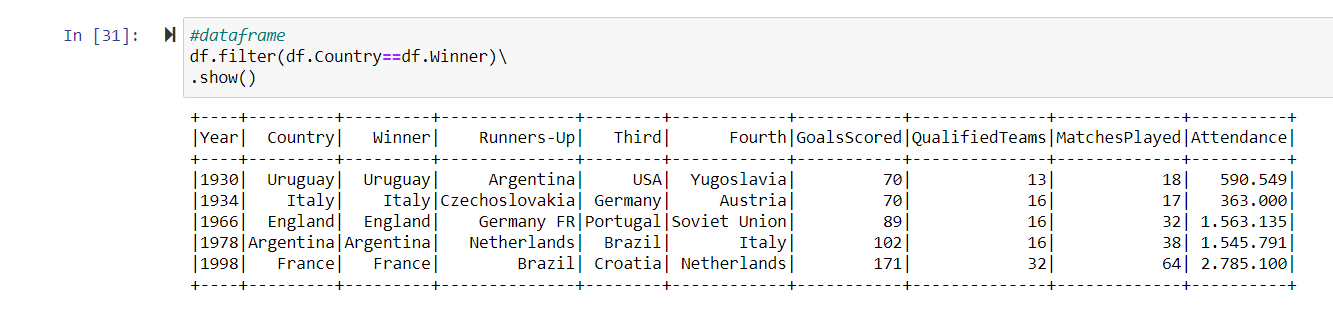
To find the countries with highest number of matches played in all the years.



**Query 4:**

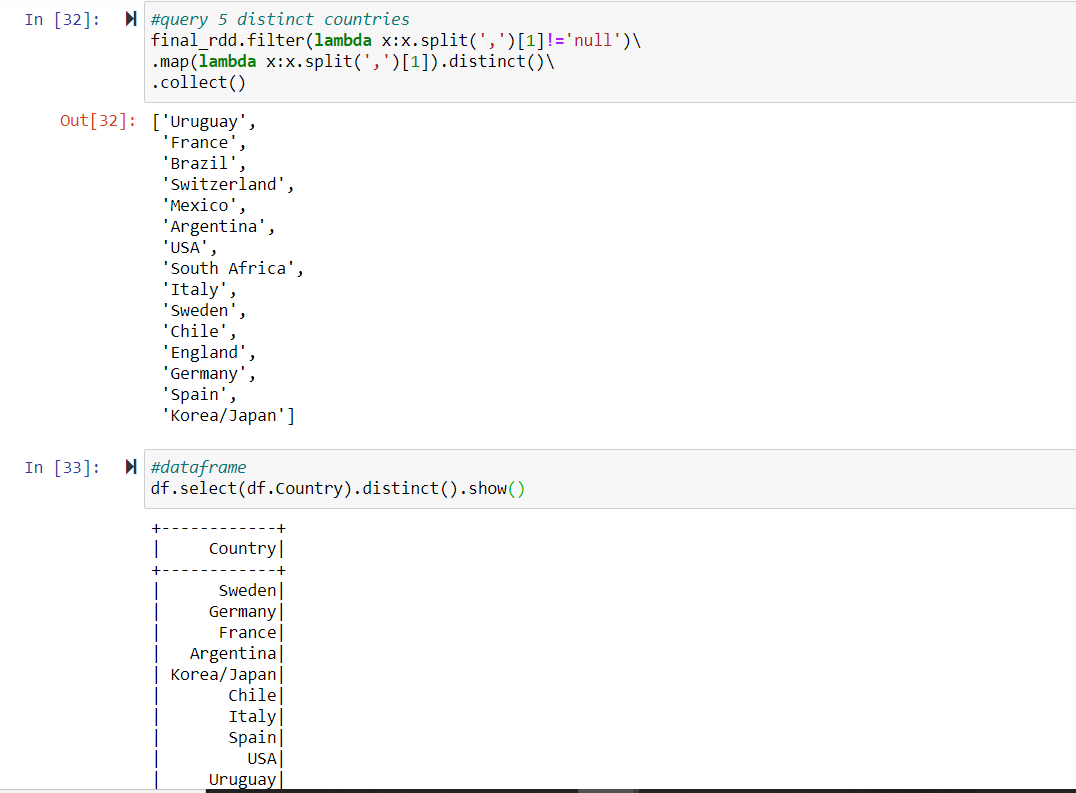
To find the records with winner and the country as same.





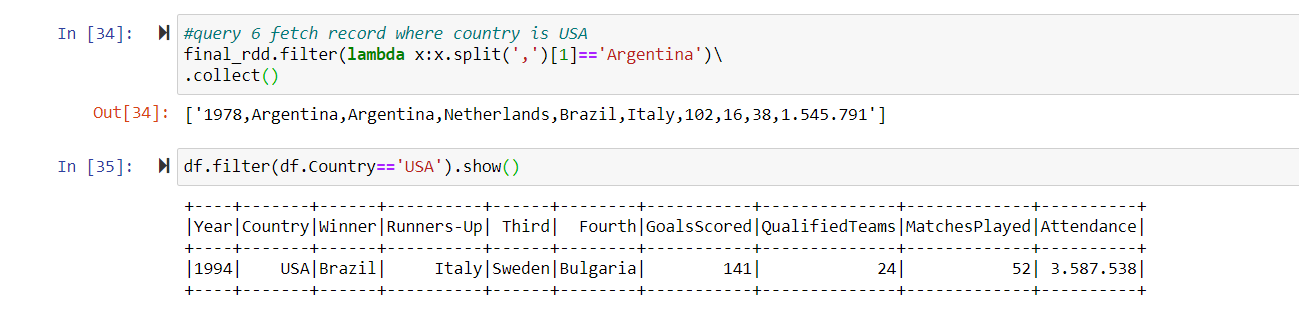
**Query 5:**

To find the distinct countries in the records.



**Query 6:**

To fetch records where country is USA.



**Challenges: (Question 5 part 4)**

It was bit challenging to write the queries which differentiate itself with other queries being used. I also tried to perform above queries which are different from one another.

**Integration: (Question 5 part 5)**

* We could successfully complete this project, which has four use cases. As there are 4 Use cases we have chosen based on our choice and worked accordingly so, it was not that difficult as we have worked independently. We finally integrated all our work in repository called BDP\_Project2

**Links of Use Cases:**

**Use-case1 - Hadoop Map reduce Algorithm**

Project-link : <https://github.com/RoshiniVarada/BDP_Project2/tree/master/CASE-1>

Wiki-link: <https://github.com/RoshiniVarada/BDP_Project2/wiki/CASE1>

**Use-case2- Spark DataFrames**

Project-link: <https://github.com/RoshiniVarada/BDP_Project2/tree/master/CASE-2>

Wiki-link: <https://github.com/RoshiniVarada/BDP_Project2/wiki/CASE2>

Video Link: <https://www.youtube.com/watch?v=Ii-m_Zb3QCs>

**Use-Case3- Spark streaming**

Project-link: <https://github.com/RoshiniVarada/BDP_Project2/tree/master/CASE-3>

Wiki-link: <https://github.com/RoshiniVarada/BDP_Project2/wiki/CASE3>

**Use-case4- Spark Graphx**

Project-link: <https://github.com/RoshiniVarada/BDP_Project2/tree/master/CASE-4>

Wiki-link: <https://github.com/RoshiniVarada/BDP_Project2/wiki/CASE4>

.