PBD PHASE – II   SUBMISSION

# Twitter Analysis Final Report

Submitted by:

Vilas Mamidyala – 16221966

Dinesh Kumar Bandam–16214040

Ranjitha Reddy BhumiReddy – 12516477

Submitted on:

04/07/2016

**Introduction:**

**First Part: IntelliJ IDEA 15**

In the second phase we have collected Streaming data using a Curl command. The collected tweets were saved as a JSON file in the local drive. Now we use the IntelliJ IDEA 15 software to write Queries in Scala by giving the input to the queries from Local drive. We now execute the query and the output of the query is stored in JSON file.

For visualization we used d3.js, which is a html, css and svg code which give the graphical visualization in the form of various graphs and charts. The query output file is converted into csv file and is given as input to the d3.js code for each visualization.

**Second Part: IBM Blue mix**

This part is developed in IBM Bluemix using the available in- built features of IBM Bluemix. We have used *insights for twitter* for collecting tweets on a specific topic. In this project we collected the tweets based on the keyword such as virat kohli, cricket and dance. These tweets were then stored to DashDB tool provided by IBM Bluemix. This Dash DB tables were connected to Apache spark instance in the IBM Bluemix. Spark SQL queries were built on the Scala Notebook and the resultant data was saved in the form of parquet file. The data which we collected was visualized using the python notebook present in IBM Bluemix with the help of matplotlib.

**Technology:**

1. Curl command from twitter developers in Ubuntu.

2. Scala and intelij

3. Spark framework for running the Spark SQL queries.

4. D3.js - For visualization.

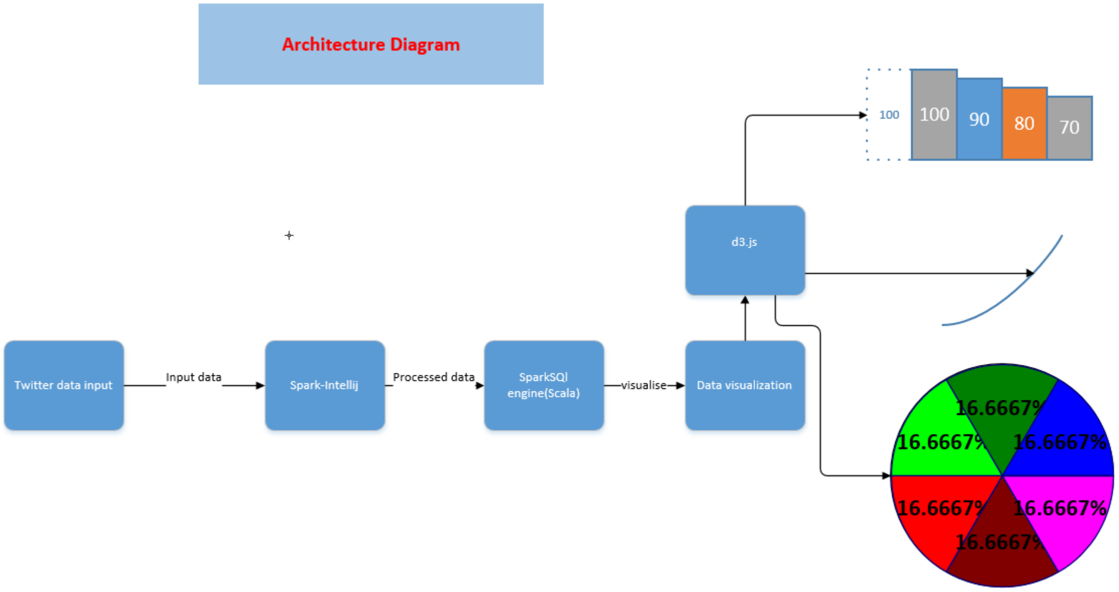
5.IBM Bluemix (Scala and Python Notebook)

D3.js:

D3.js is used to create data visualizations using a library in java script format which produces dynamic result. This D3 took help of HTML, CSS and SVG web standards. D3 provides a manageable control over visual result when compared to other similar technologies.

https://d3js.org

Architechtural Diagram :



**References:**

We have collected the tweets by using curl command from twitter in Linux.

```
curl --get 'https://stream.twitter.com/1.1/statuses/sample.json' --header 'Authorization: OAuth
oauth_consumer_key="ZXia2FH1Ngj2vkRoZX1dlrxzh", oauth_nonce="df6691667070788264b4d18f24d6cba6",
oauth_signature="Q83adxwgokgljUP6v16KGTZPAM8%3D", oauth_signature_method="HMAC-SHA1",
oauth_timestamp="1456725586", oauth_token="220954604-3kKoMiUTYGiHDtS21YqNTQV8rJnaMXnTuGe6guyj",
oauth_version="1.0"' --verbose
```

**Data Analytic service Apache Spark-1.4.1**

**SCALA NOTEBOOK:** This notebook is connected to the dashDB database and to the Scala service and ran all the queries on the table of VIRAT and Cricket and stored the results in a new file with "*.Parquet"* file format.

**IPYTHON NOTEBOOK:** After creating a new notebook in python, Using IPYTHON notebook we have accessed the "*.Parquet"* file of scala notebook and depicted the plots and the graphs for all the results from the queries.

Reference:

https://d3js.org

**http://spark.apache.org/sql/**

**dev.twitter.com/apps**

**GitHUB URL :**

https://github.com/vilasmamidyala/PBD_Project2016

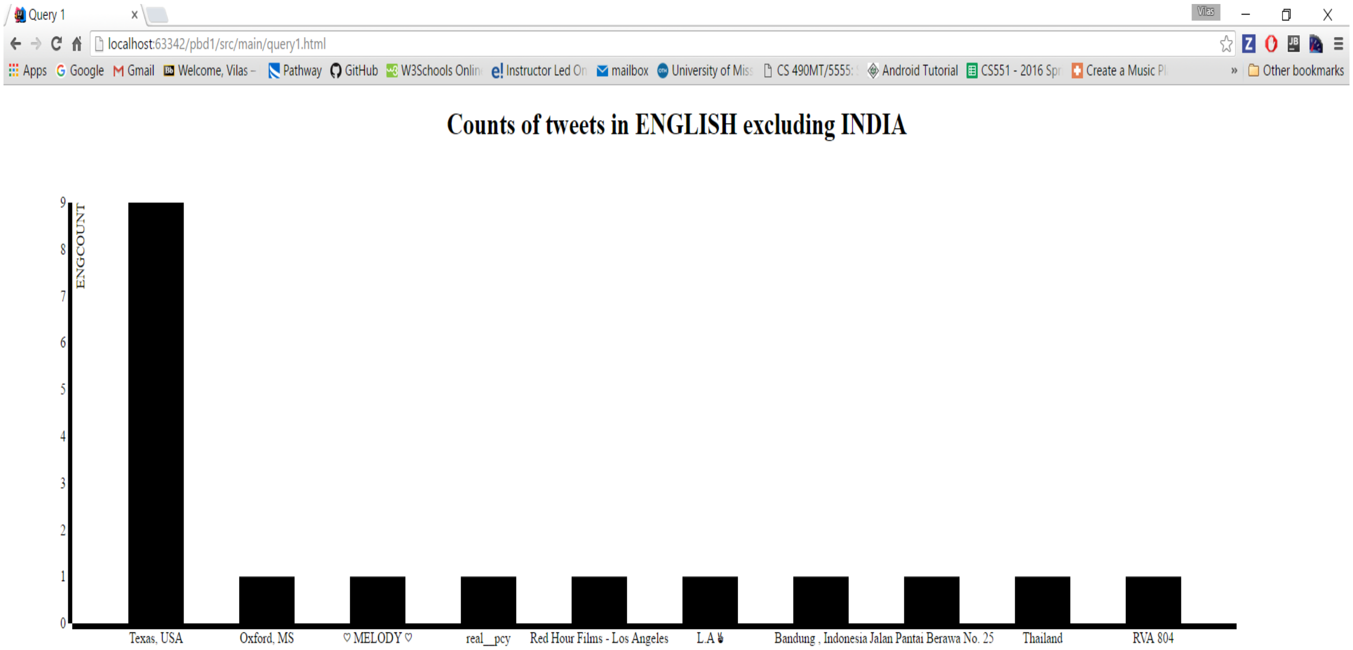
**Tweet File DropBox Link**

https://www.dropbox.com/s/36r8a8vjcowvt60/Tweets2.txt?dl=0

QUERIES ::

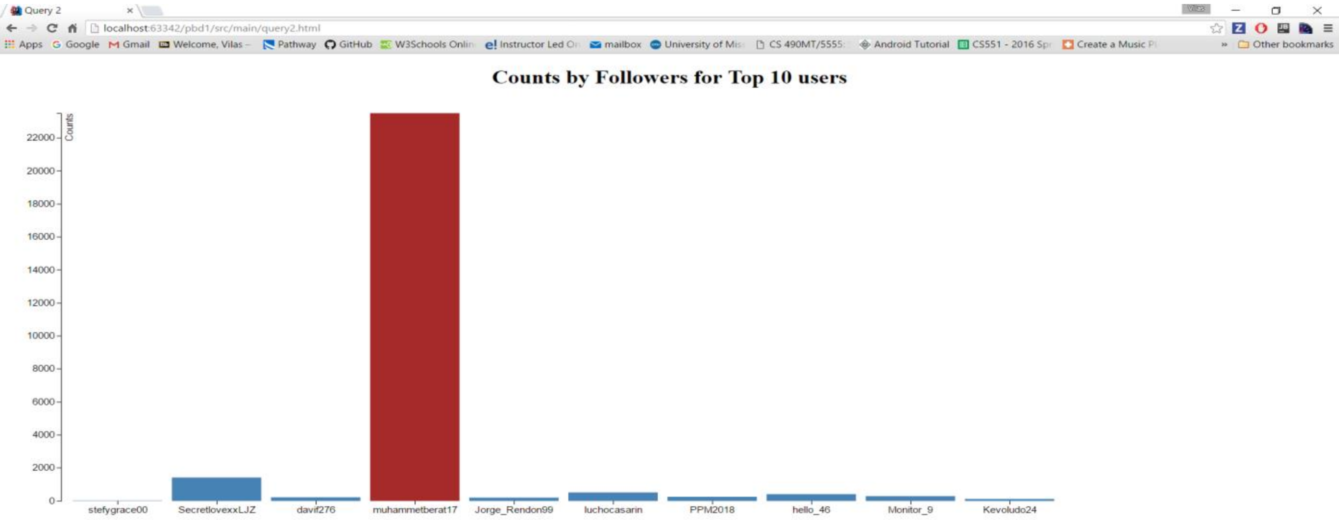
Query 1 : SELECT user.location as location,COUNT(*) AS ENGCOUNT FROM tweets WHERE lang='en' and user.location NOT IN ('INDIA','India','india','null') GROUP BY user.location order by 2 desc limit 10

This query collects the tweets that are in “English” other than India.



Query 2 : select user.screen_name as username,max(user.followers_count) As followers_count from tweets2 WHERE user.followers_count IS NOT NULL group by user.screen_name,user.followers_count limit 10

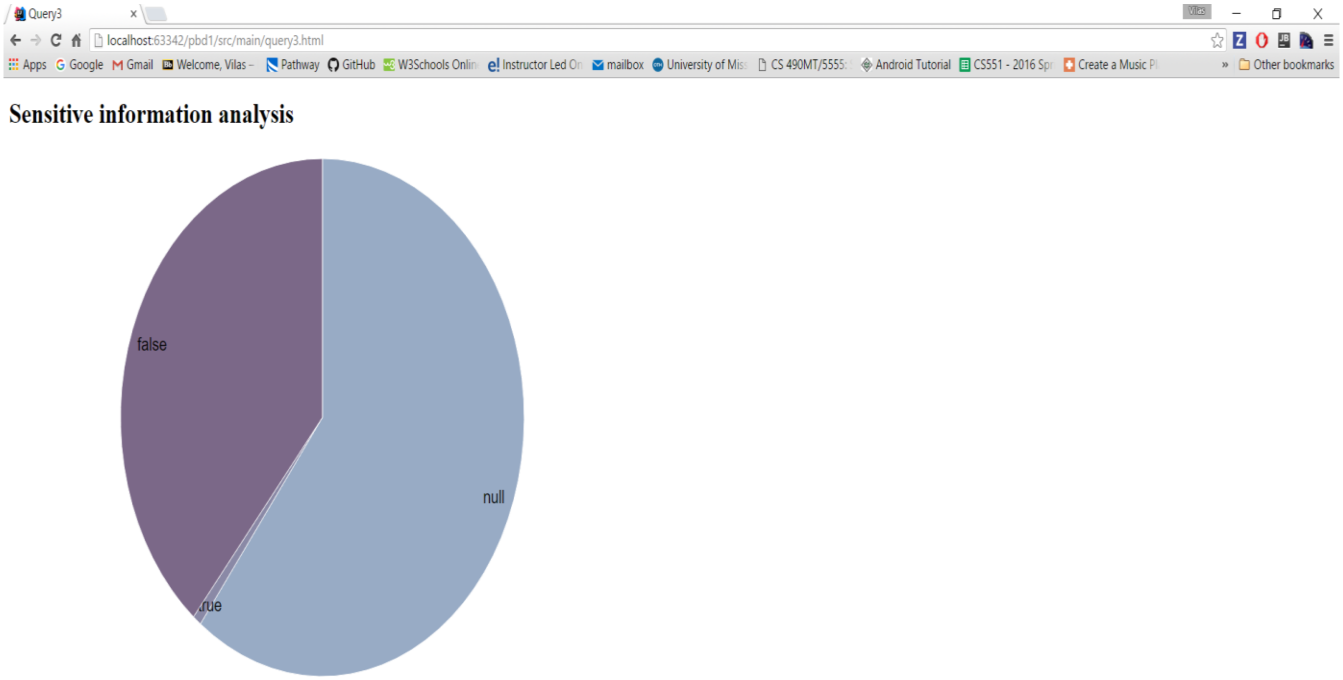
This query extracts the data which resulted in the count followers of top 10 users.



Query 3 : select distinct possibly_sensitive,count(*) as Counts from tweets2 group by possibly_sensitive.

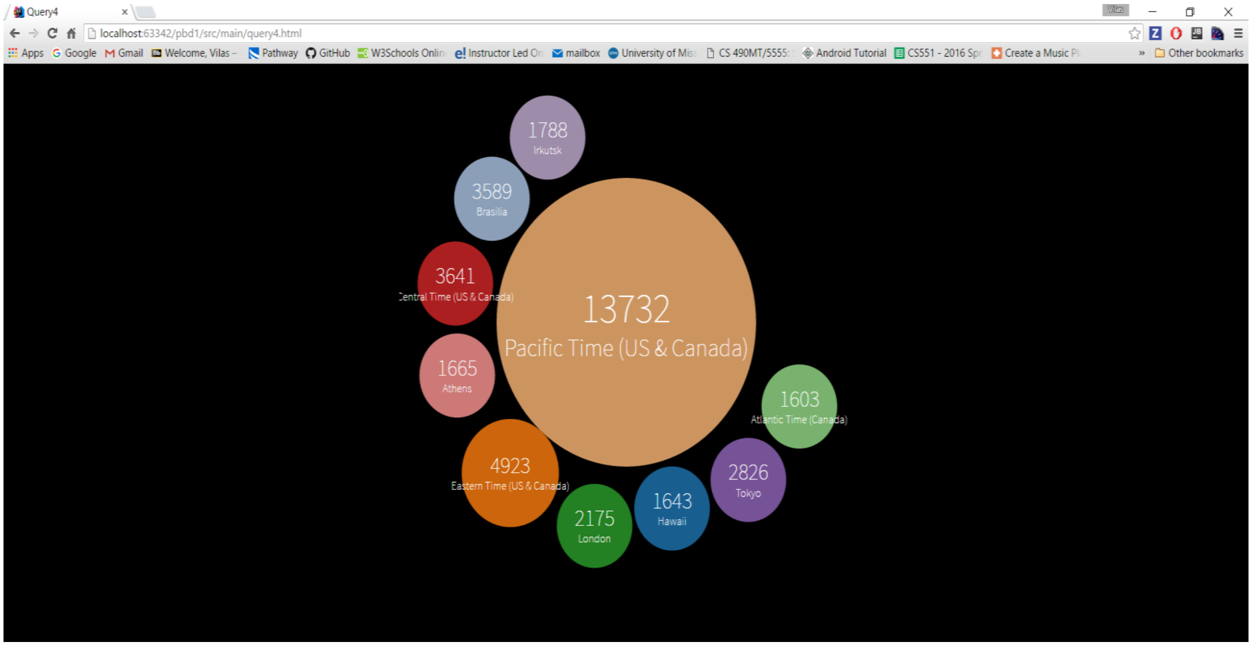
This query shows that how many users are tweeted through sensitive information.

Sensitive information in the sense a new field is added to API responses which only surfaces when a tweet contains a link.



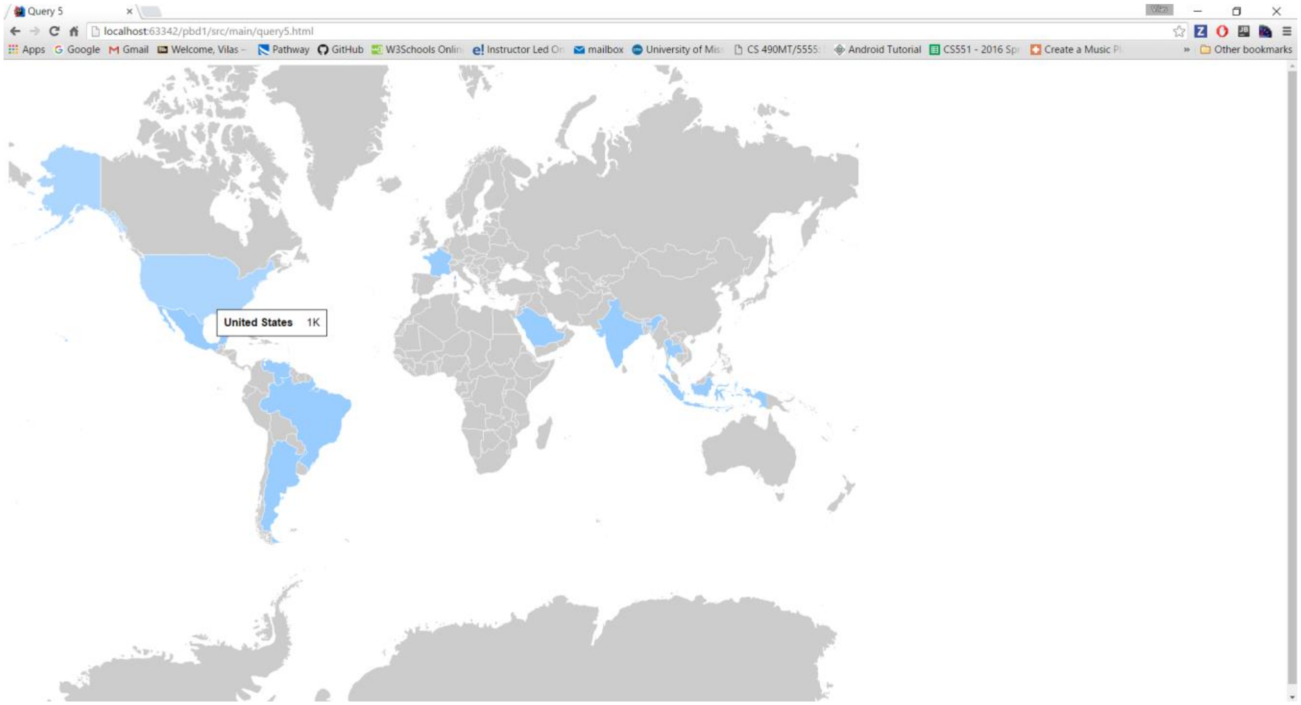
Query 4 : SELECT user.time_zone as timezone, SUBSTR(created_at, 0, 10) as postedtime, COUNT(*) AS total_count FROM tweets2 WHERE user.time_zone IS NOT NULL AND SUBSTR(created_at, 0, 10) in ('Sat Feb 27') GROUP BY user.time_zone,SUBSTR(created_at, 0, 10) ORDER BY total_count DESC LIMIT 10"

This query displays how many tweets are posted from top 10 time zones.



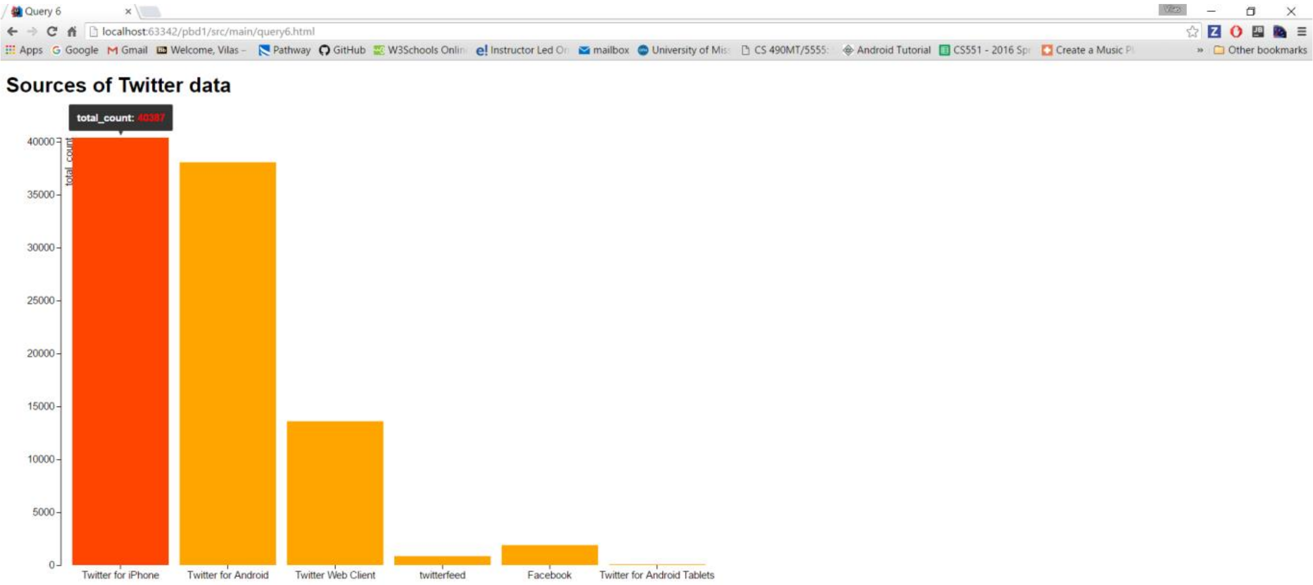
Query 5 : SELECT distinct user.location as location,count(*) AS total_count FROM tweets2 GROUP BY user.location ORDER BY total_count DESC LIMIT 20

This query shows number of tweets posted for selected country, here it is United States while taking screen shot.



Query 6 : select distinct source,count(*) AS total_count from tweets2 where source like '%Twitter for Android%' or source like '%Twitter for iPhone%' or source like '%Twitter Web Client%' or source like '%twitterfeed%' or source like '%Facebook%' group by source limit 10

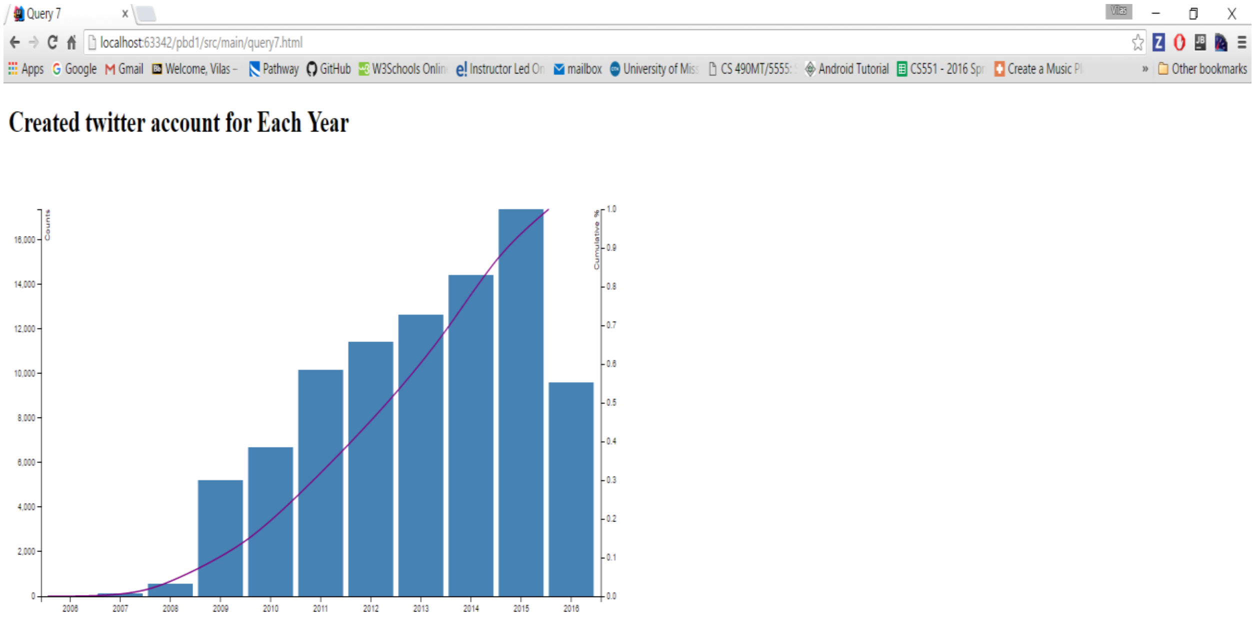
This query shows all the tweets that are posted from different sources like Android , iphone.



Sources of Twitter data

Query 7 : select SUBSTR(user.created_at, 26, 30) as Account_created,count(*) as Counts from tweets3 where SUBSTR(user.created_at, 26, 30) is not null group by SUBSTR(user.created_at, 26, 30) order by Account_created limit 15

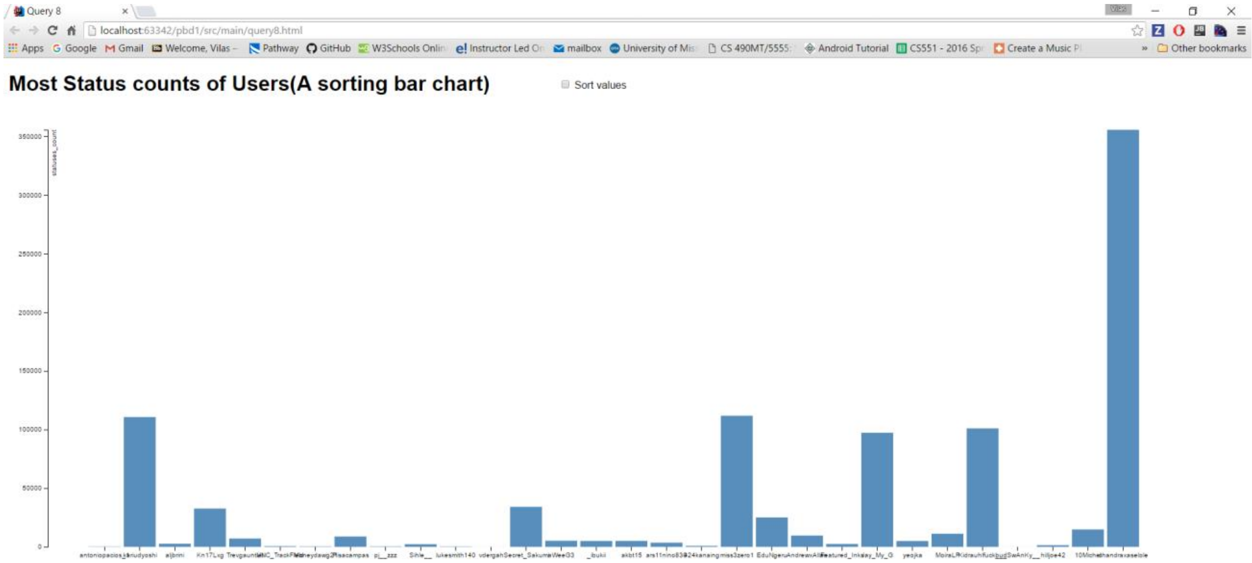
This query results that how many twitter accounts are created each year.



Created twitter account for Each Year

Query 8 : select distinct user.screen_name as username,max(user.statuses_count) As statuses_count from tweets2 WHERE user.statuses_count IS NOT NULL group by user.screen_name limit 30

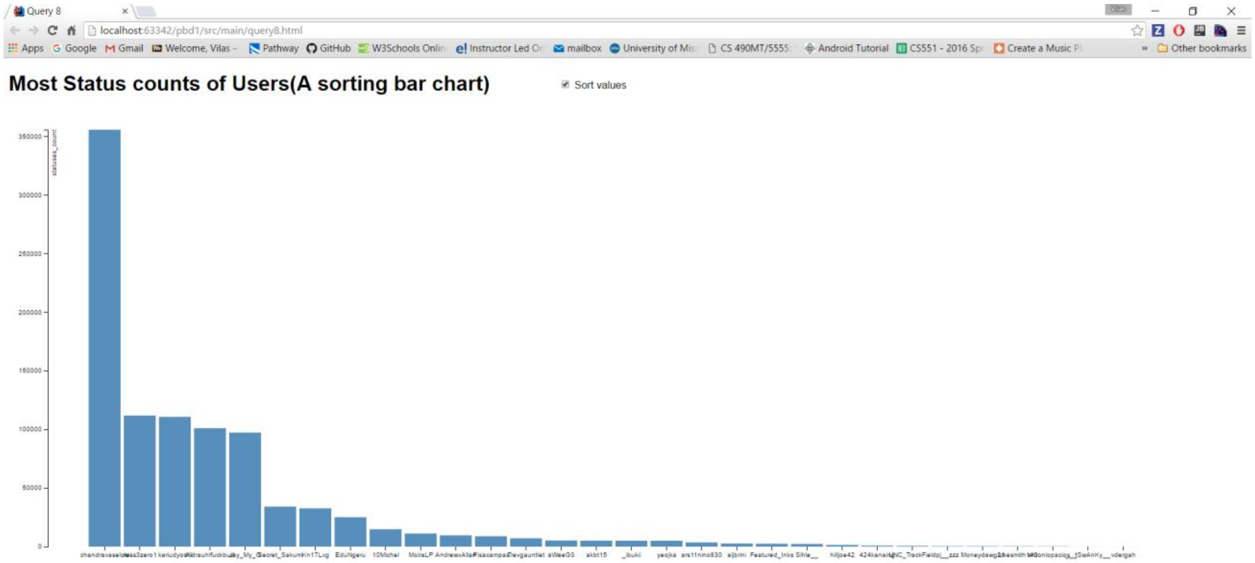
This query collects data showing that the user and the maximum no.of statuses that he has.



localhost:63342/pbd1/src/main/query8.html

Most Status counts of Users(A sorting bar chart)

☐ Sort values



localhost:63342/pbd1/src/main/query8.html

Most Status counts of Users(A sorting bar chart)

☑ Sort values

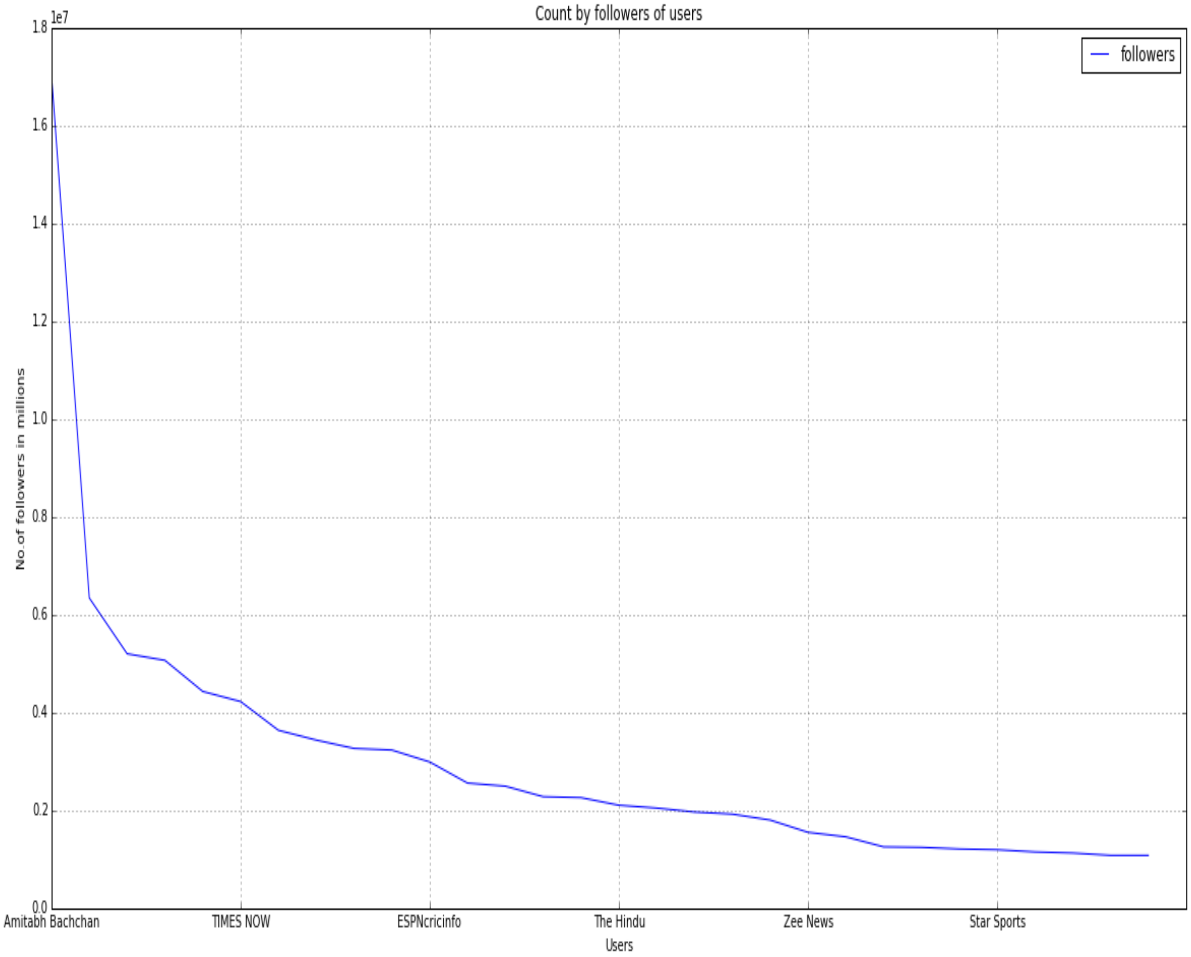
Queries in IBM Bluemix:

We have used two data bases . First data base contains all the tweets that are related to kohli and cricket. Second data base contains tweets related to dance.

Queries using first database : Kohli

Query 1 :

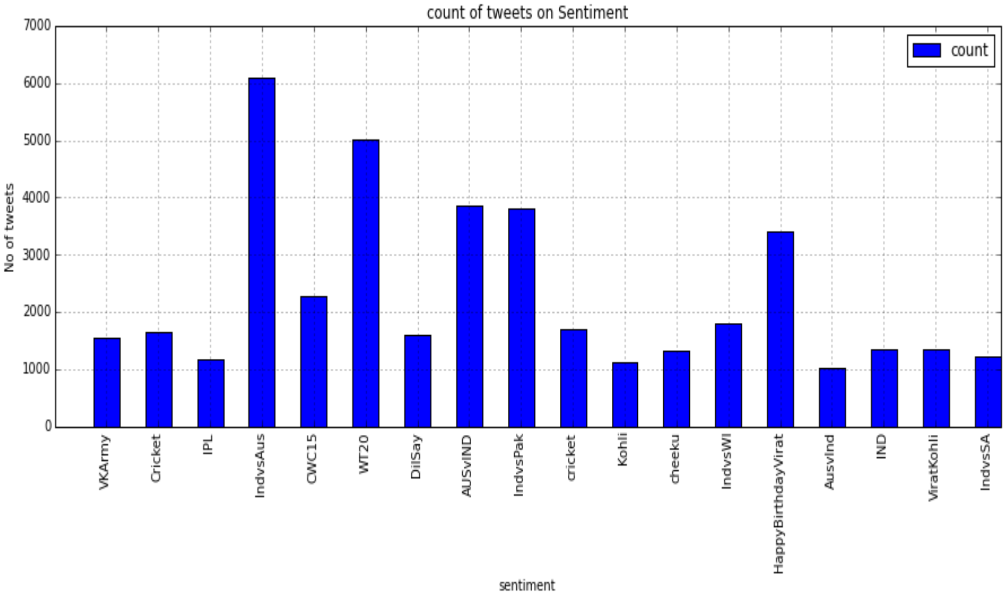
select USER_DISPLAY_NAME,max(USER_FOLLOWERS_COUNT) as followers from tweets GROUP by USER_DISPLAY_NAME order by followers desc limit 20.



Query 2 :

select DISTINCT (HASHTAG),COUNT (*) as count from tweets GROUP BY HASHTAG having count>100 order by count desc 10

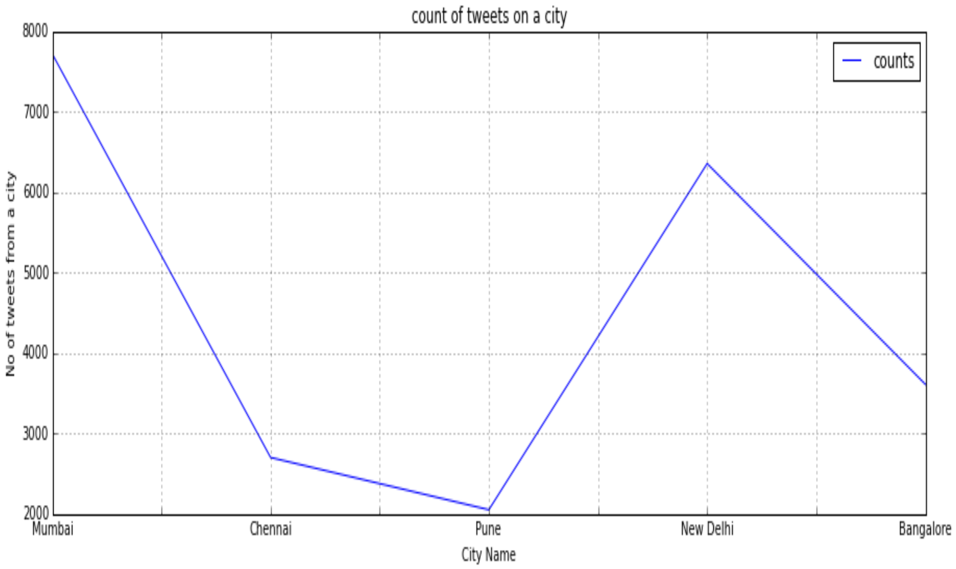
This query deals with the topic of hashtag on which more tweets are running.



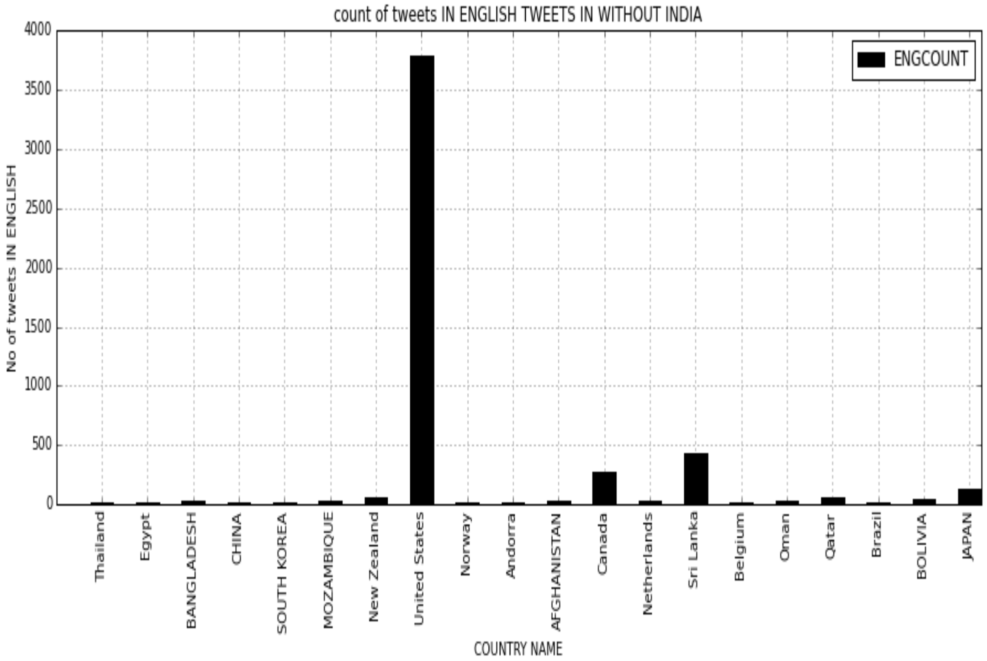
Query 3 :

SELECT USER_CITY, COUNT(*) AS counts FROM tweets GROUP BY USER_CITY HAVING USER_CITY in ('Mumbai','New Delhi','Bangalore','Chennai','Pune') order by 2 desc

This query reveals the data of top 5 cities specified from which more tweets are posted.

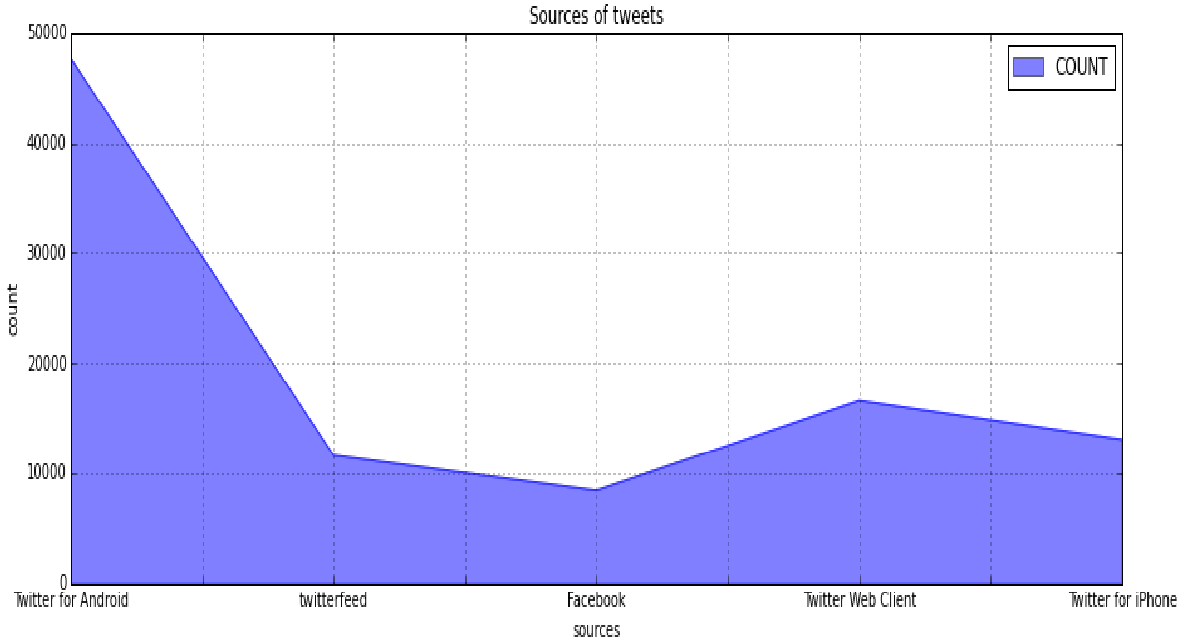


Query 4 : SELECT DISTINCT USER_COUNTRY,COUNT(*) AS ENGCOUNT FROM tweets WHERE MESSAGE_LANGUAGE='en' and USER_COUNTRY NOT IN ('INDIA','India','india','null') GROUP BY USER_COUNTRY order by 2 desc limit 50



Query 5 :

select DISTINCT MESSAGE_GENERATOR_DISPLAY_NAME, COUNT(*) AS COUNT from tweets where MESSAGE_GENERATOR_DISPLAY_NAME in ('Twitter for Android','Twitter Web Client','Twitter for iPhone','twitterfeed','Facebook') GROUP BY MESSAGE_GENERATOR_DISPLAY_NAME



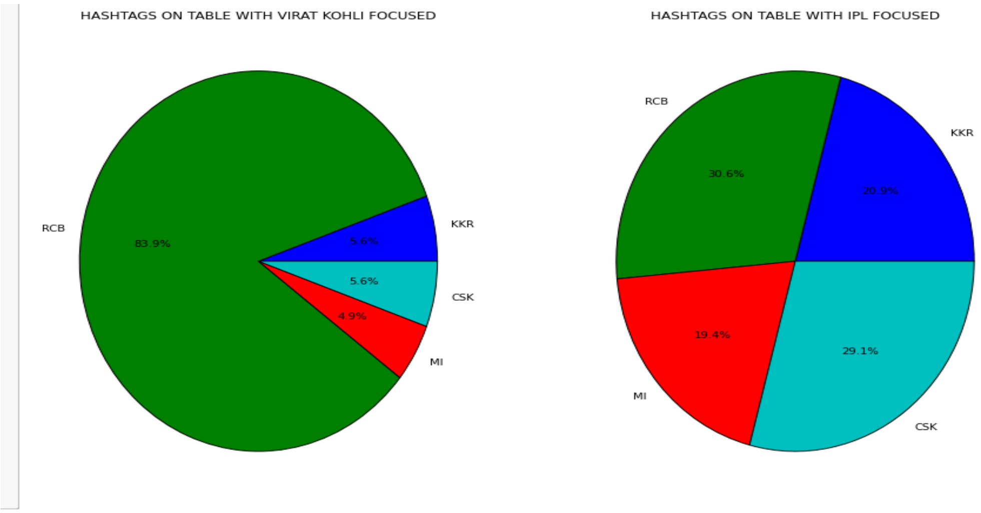
Query 6:

select HASHTAG,count(*) AS COUNTS from CRICKET1 WHERE HASHTAG IN ('RCB','CSK','KKR','MI') group by HASHTAG ORDER BY 2 DESC

This query collects data of all the tweets that are hashtagged with virat kohli focused.

select HASHTAG,count(*) AS COUNTS from CRICKET2 WHERE HASHTAG IN ('RCB','CSK','KKR','MI') group by HASHTAG ORDER BY 2 DESC")

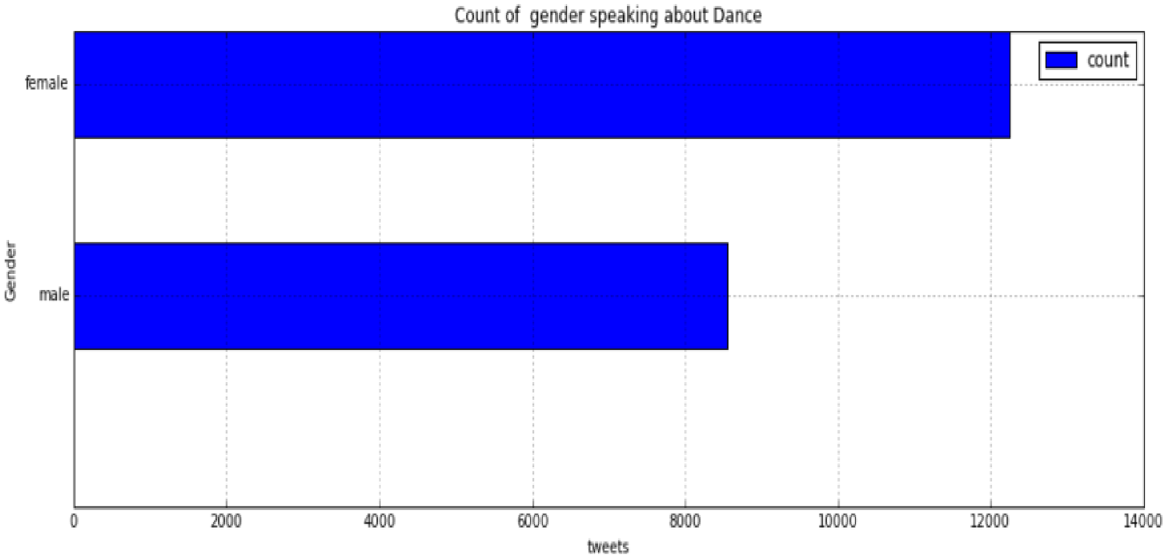
This query collects data of all the tweets that are hashtagged with IPL focused.



Queries on second data base : Dance

Query 7: select distinct USER_GENDER,count(*) as count from tweets group by USER_GENDER having USER_GENDER IN ('male','female')

This query results that how many users are female and how many are male who are who are tweeting about dance.



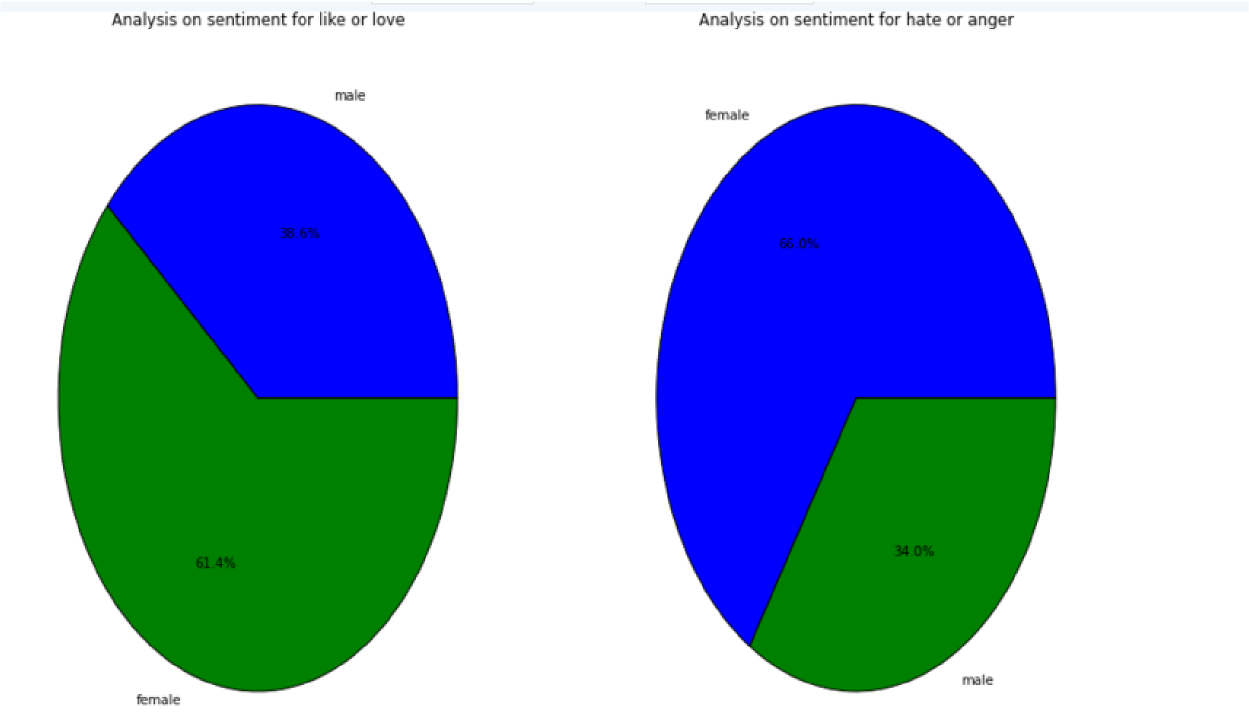
Query 8: This query deals with sentimental analysis.

select USER_GENDER,count(*) as counts from senti where MESSAGE_BODY like('%like%') or MESSAGE_BODY like('%love%') GROUP BY USER_GENDER HAVING USER_GENDER IN ('male','female')

This query collects all data from both the genders who are positive like love and like out of all the tweets about dance.

select USER_GENDER,count(*) as counts from senti where MESSAGE_BODY like '%hate%' or MESSAGE_BODY like '%angry%' GROUP BY USER_GENDER HAVING USER_GENDER IN ('male','female')

This query collects all data from both the genders who arenegative like hate and anger out of all the tweets about dance .


Analysis on sentiment for like or love
male
female
Analysis on sentiment for hate or anger
female
male