**CS5540**

**PRINCIPLES OF BIG DATA MANAGEMENT**

**(FALL 2018)**

**INTERIM PROJECT REPORT**

A picture containing object

Description generated with very high confidence

BY

**TEJASWI AYYADAPU (16278799)**

**RUPESH SAI RAM(16272957)**

**RAHUL REDDY YERVA (16231559)**

**OBJECTIVE:**

* To work on the tweets related to Search Engine and to figure out how to store them in Spark SQL.
* To write interesting analytical queries to explore and understand the data collected.
* To develop interesting visualizations of the above written queries.

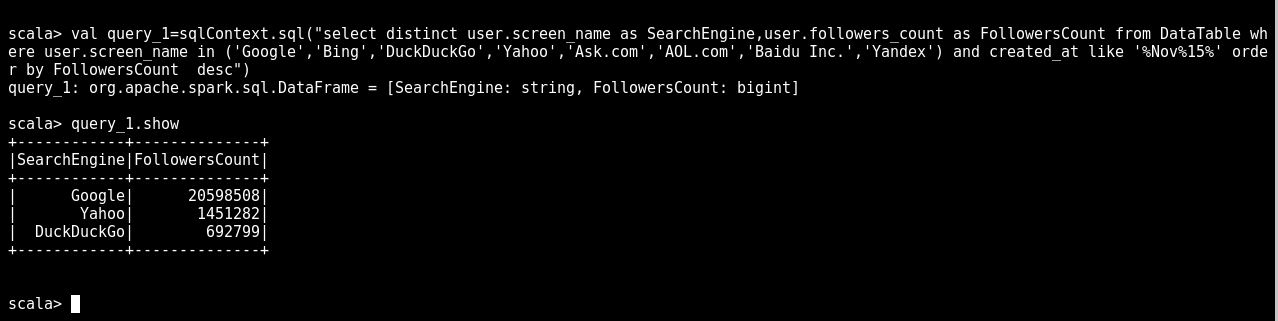
**VISUALIZATION :** To perform the visualization on the result sets and to view it in some pictorial representations like the pie charts, bar graphs etc.

**SOFTWARES PLANNING TO USE:**

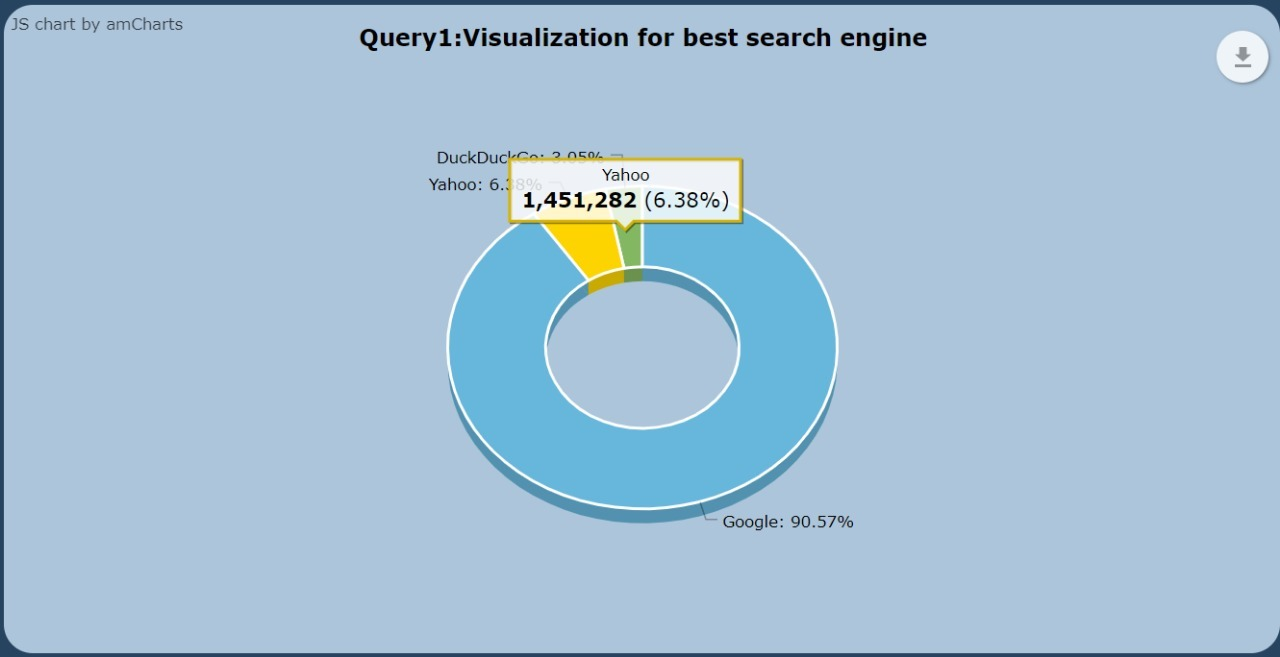
* Apache Spark
* Scala (Execution of Queries)
* High Charts for Visualizations, Tableau
* Twitter4J Library for Tweets Extraction

**1.Query to fetch best search engine according to followers count of official pages?**

>val query\_1=sqlContext.sql("select distinct user.screen\_name,user.followers\_count from DataTable where user.screen\_name in ('Google','Bing','DuckDuckGo','Yahoo') and created\_at like '%Nov%15%'")



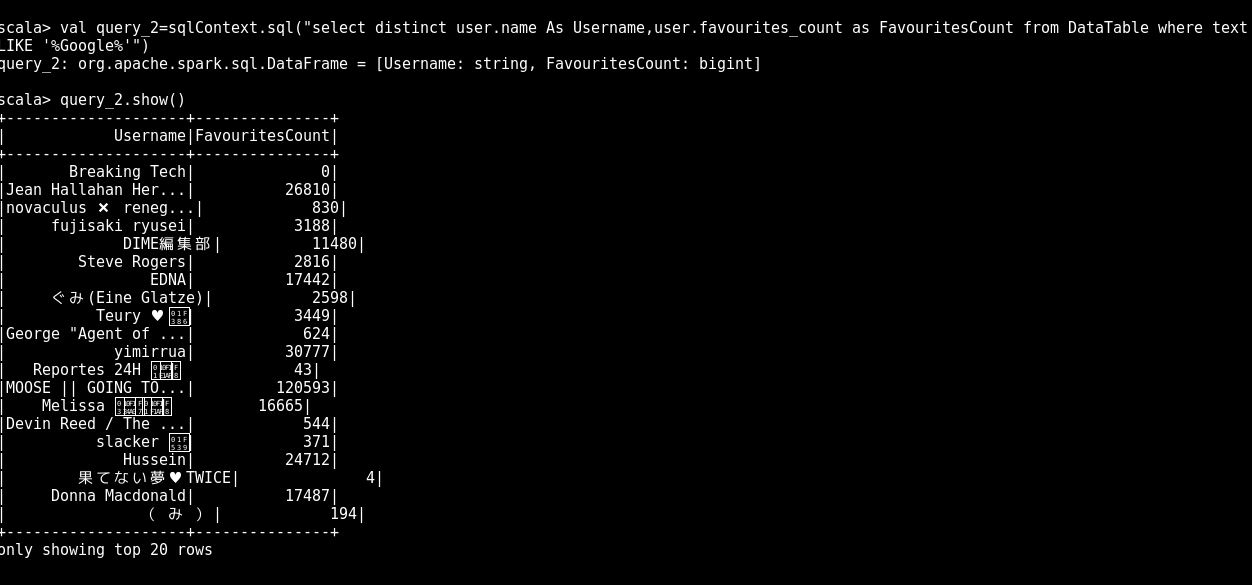
**Graph for Query 1:**



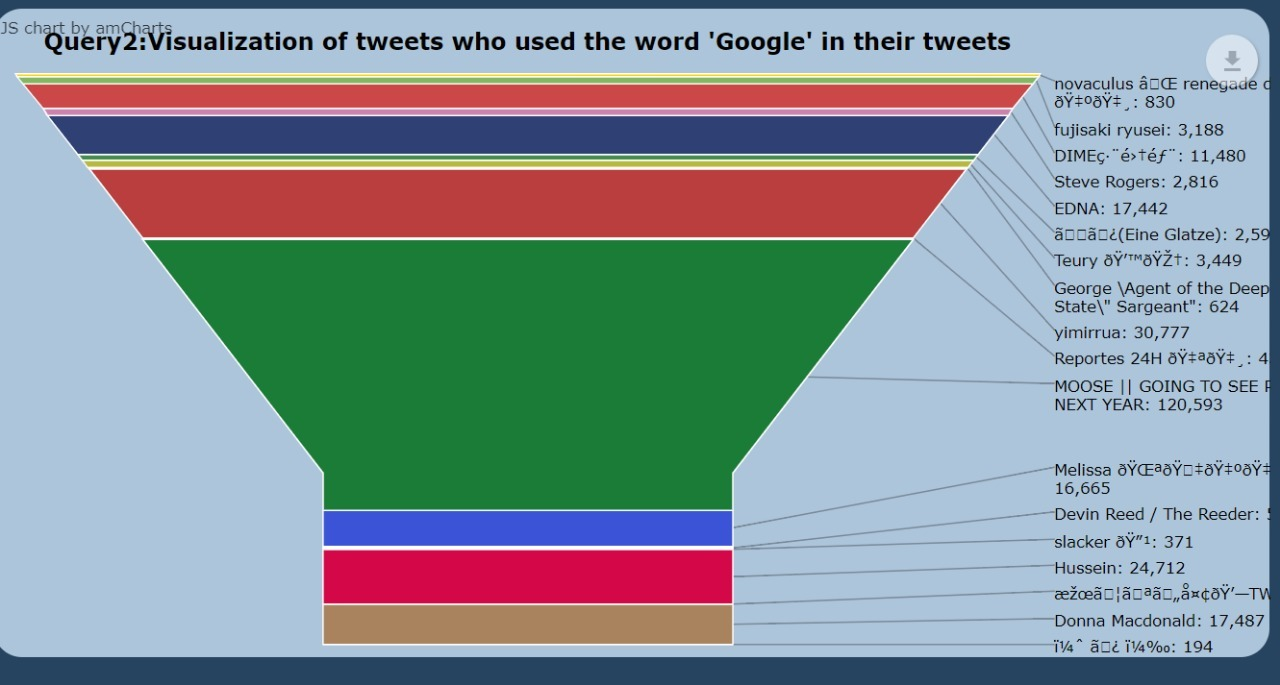
**2.Query to fetch all the users who have used the word google in thier tweets**

>

val query\_2=sqlContext.sql("select user.name As Username,user.favourites\_count from DataTable where text LIKE '%Google%'")



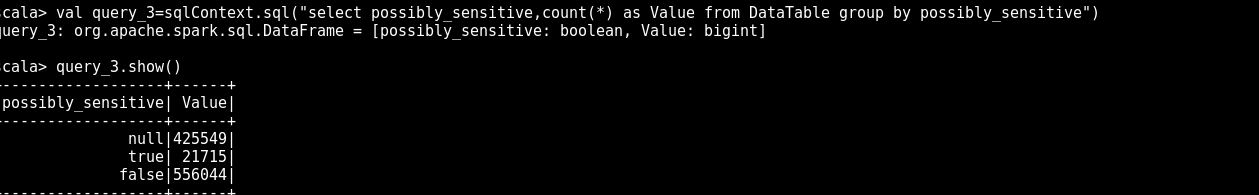
**Graph for Query 2:-**



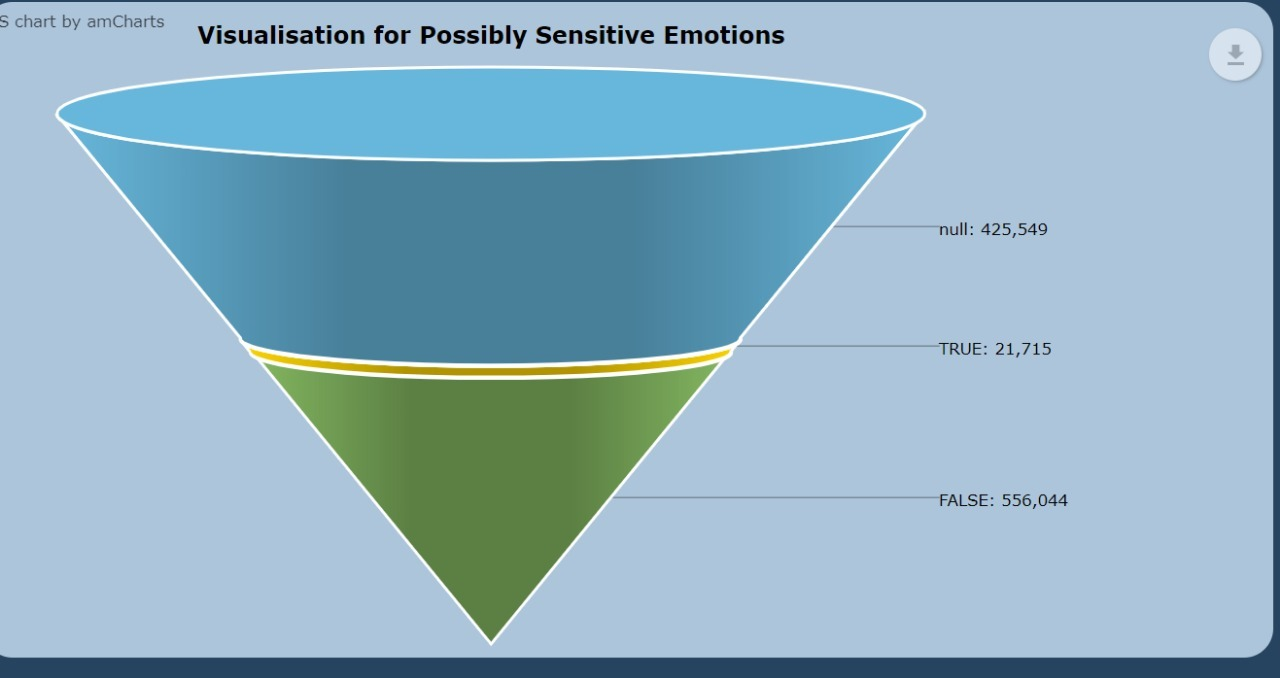
3.Query to count the possible sensitive emotions from the data

>

val query\_3=sqlContext.sql("select possibly\_sensitive,count(\*) as Value from DataTable group by possibly\_sensitive")



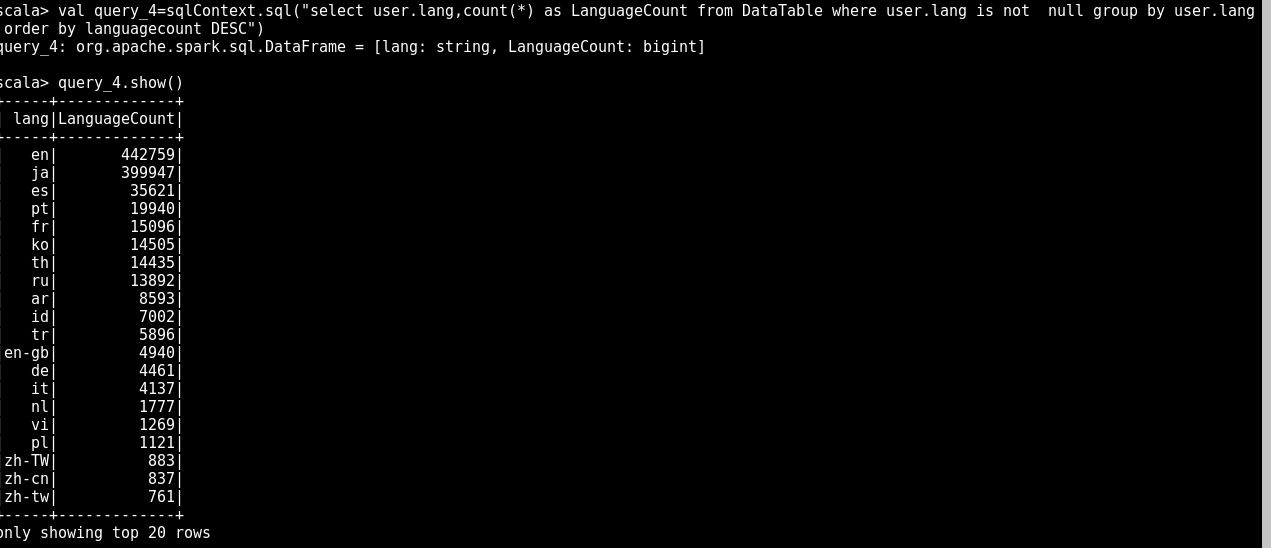
**Graph for Query 3:-**



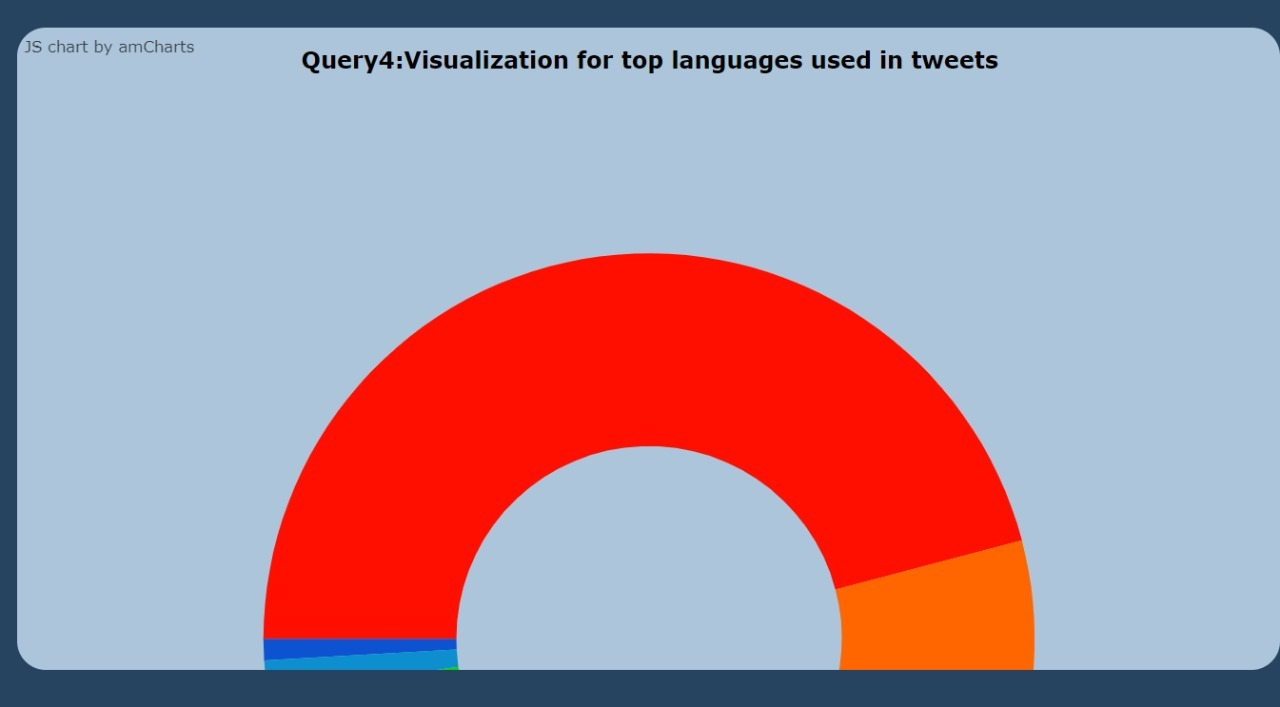
4.Query to fetch number of tweets in top languages

>

val query\_4=sqlContext.sql("select user.lang,count(\*) as LanguageCount from DataTable where user.lang is not null group by user.lang order by lang\_count DESC")

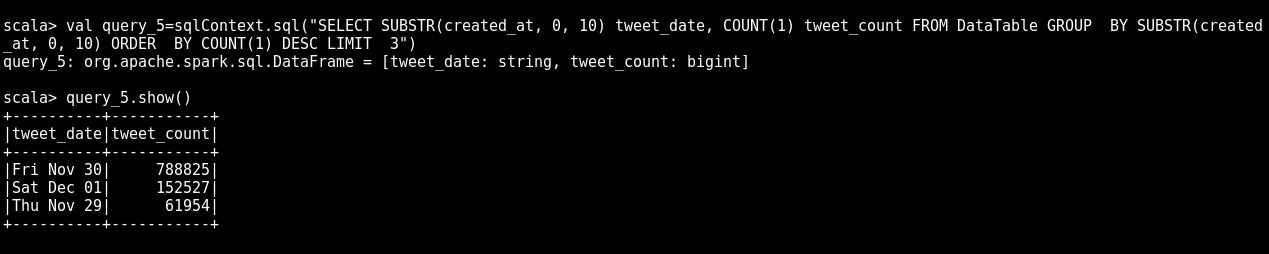


**Graph for Top Languages:**

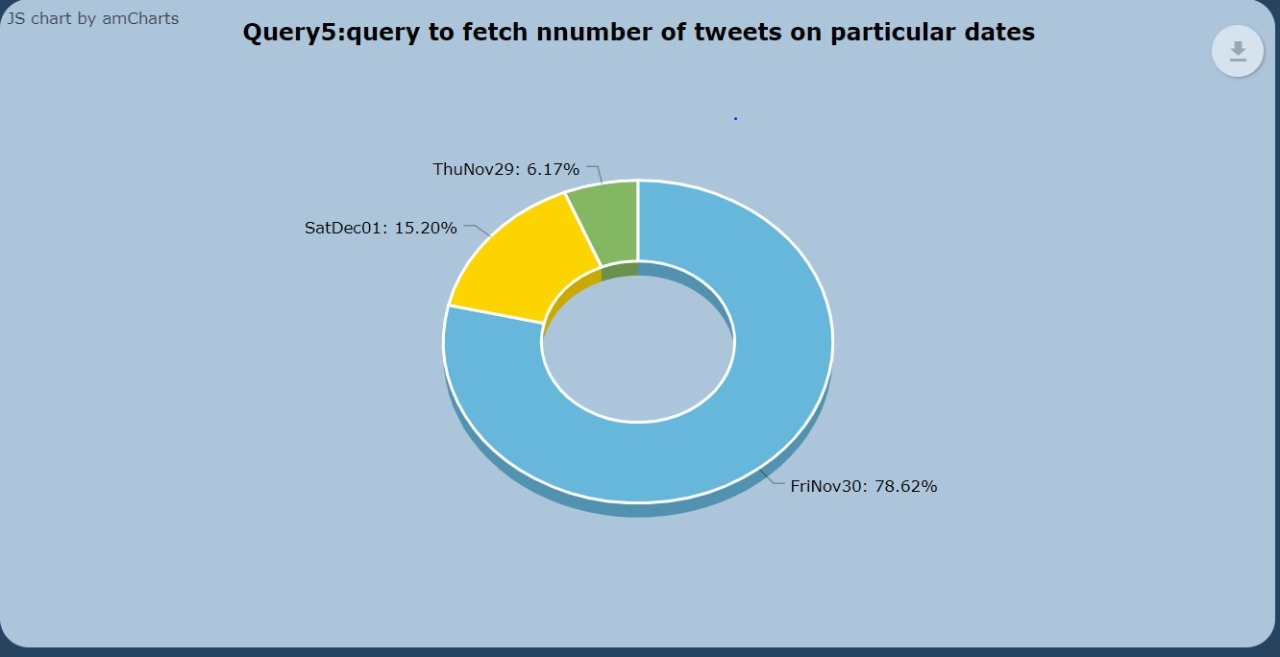


5.Query to fetch number of tweets on particular dates:-

>val query\_5=sqlContext.sql("SELECT SUBSTR(created\_at, 0, 10) tweet\_date, COUNT(1) tweet\_count FROM DataTable GROUP BY SUBSTR(created\_at, 0, 10) ORDER BY COUNT(1) DESC LIMIT 5")



**Graph for Query 5**



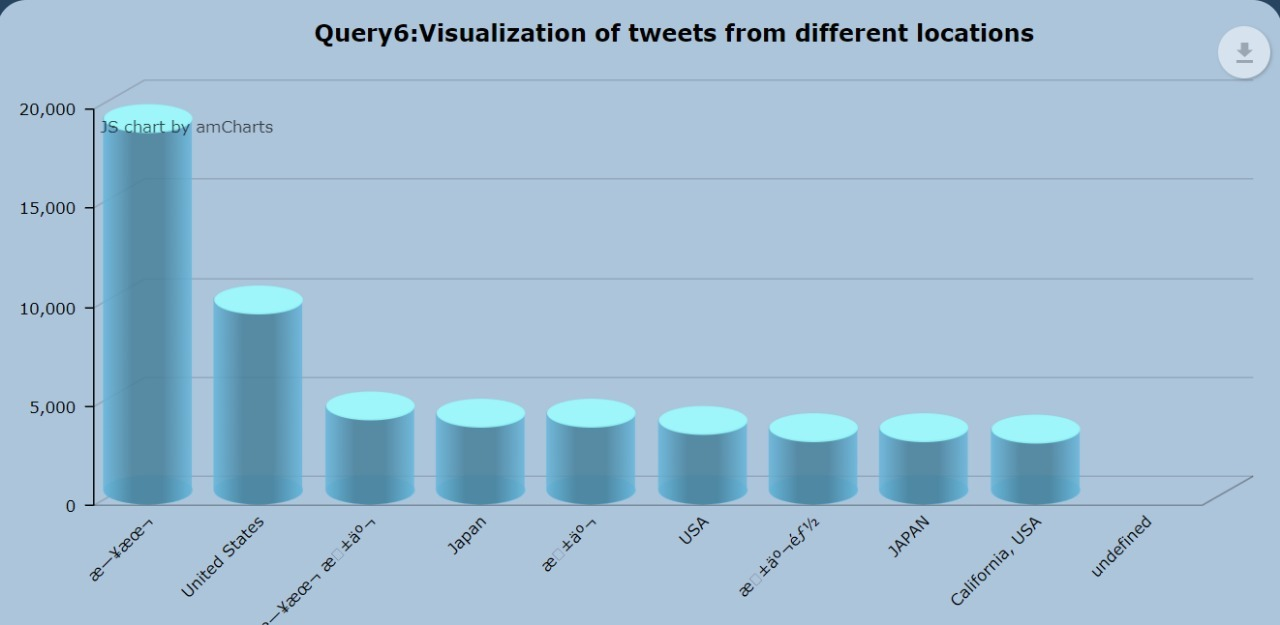
6.Query to fetch number of tweets from different locations:-

>

val query\_6=sqlContext.sql("select count(\*) as count,user.location from DataTable where user.location is not null group by user.location order by count desc")



Graph for Query 6:

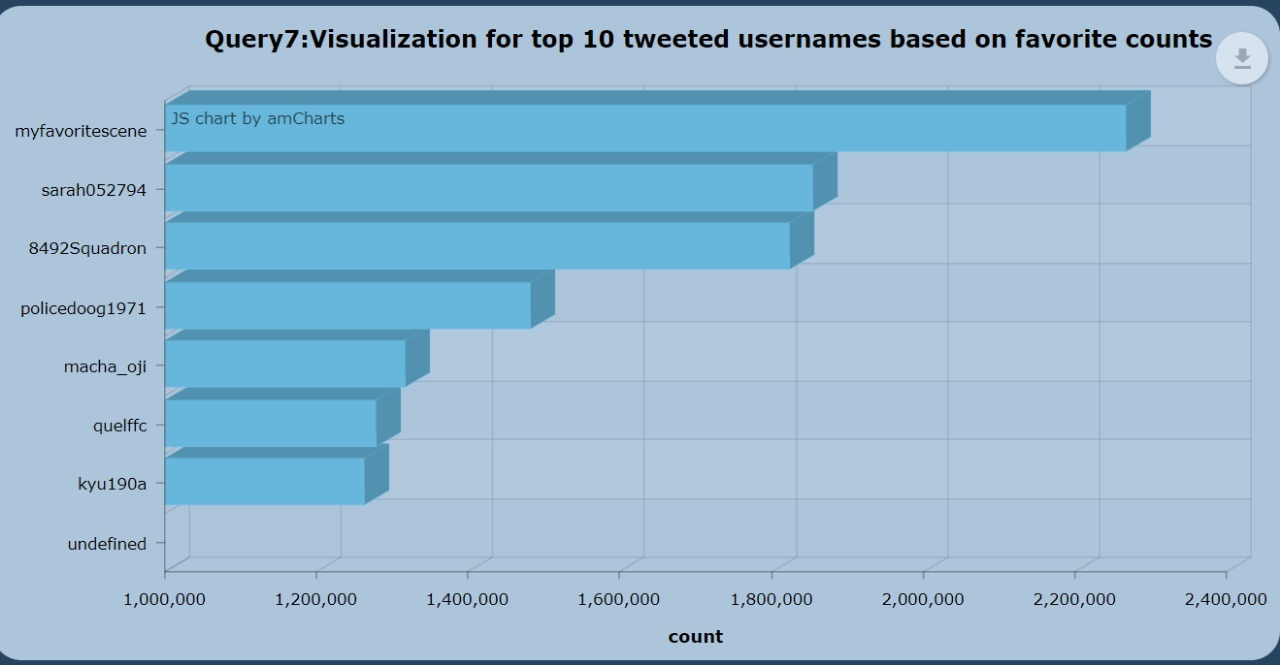


7.Query to fetch top tweeted user name based on favourite counts

val query\_7=sqlContext.sql("select user.screen\_name as ScreenName,user.favourites\_count as Favourites from DataTable order by Favourites desc limit 1")

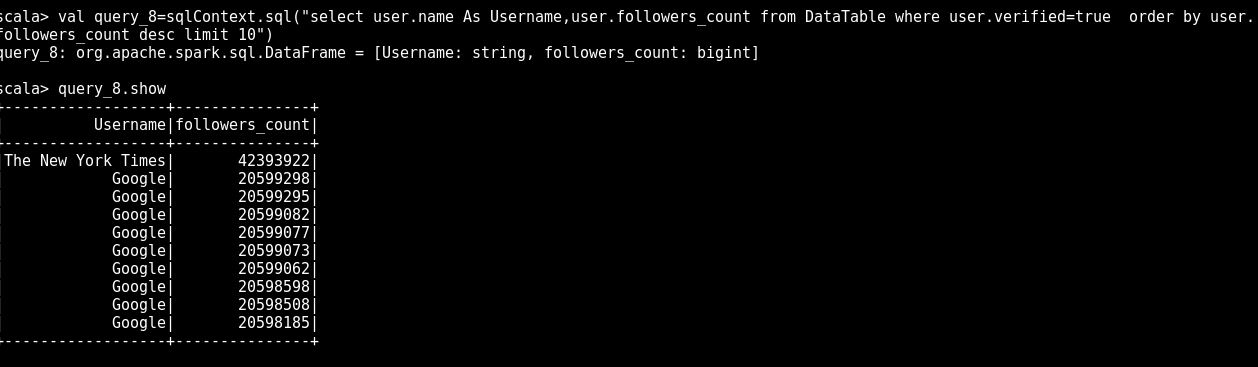


Graph for Query 7:-



8.Query to fetch top verified users with followers count

val query\_8=sqlContext.sql("select user.name As Username from DataTable where user.verified=true order by user.followers\_count desc limit 5")



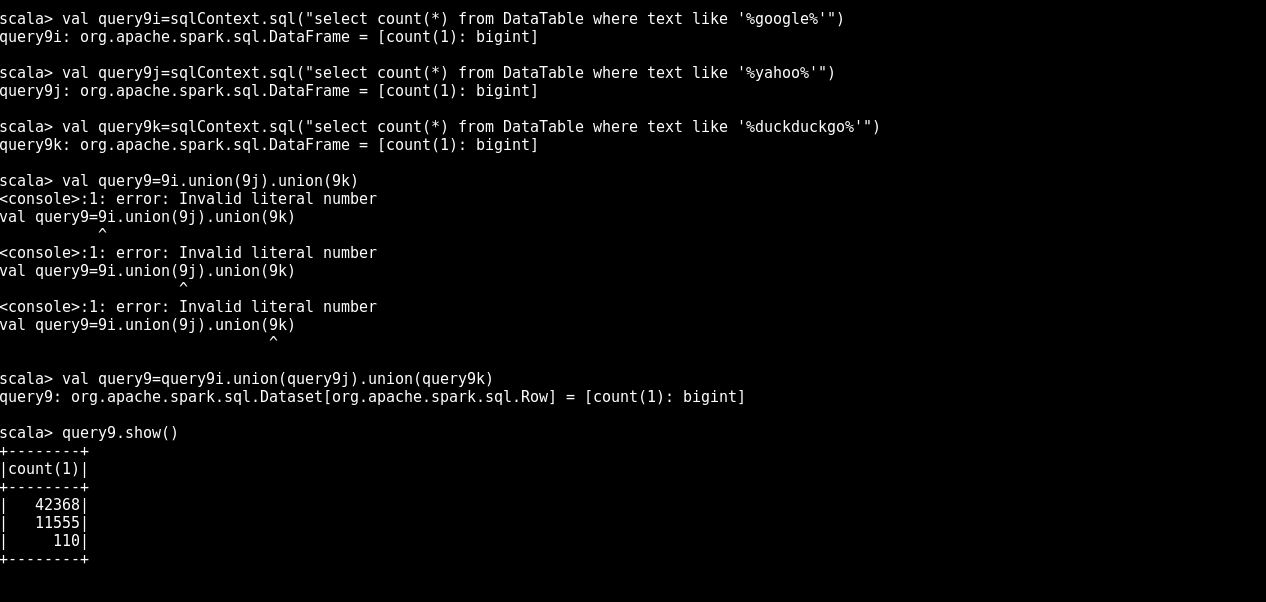
9. Query for tweets count for best search engine

val query9i= sqlContext.sql("select count(\*) from DataTable where text LIKE '%google%'")

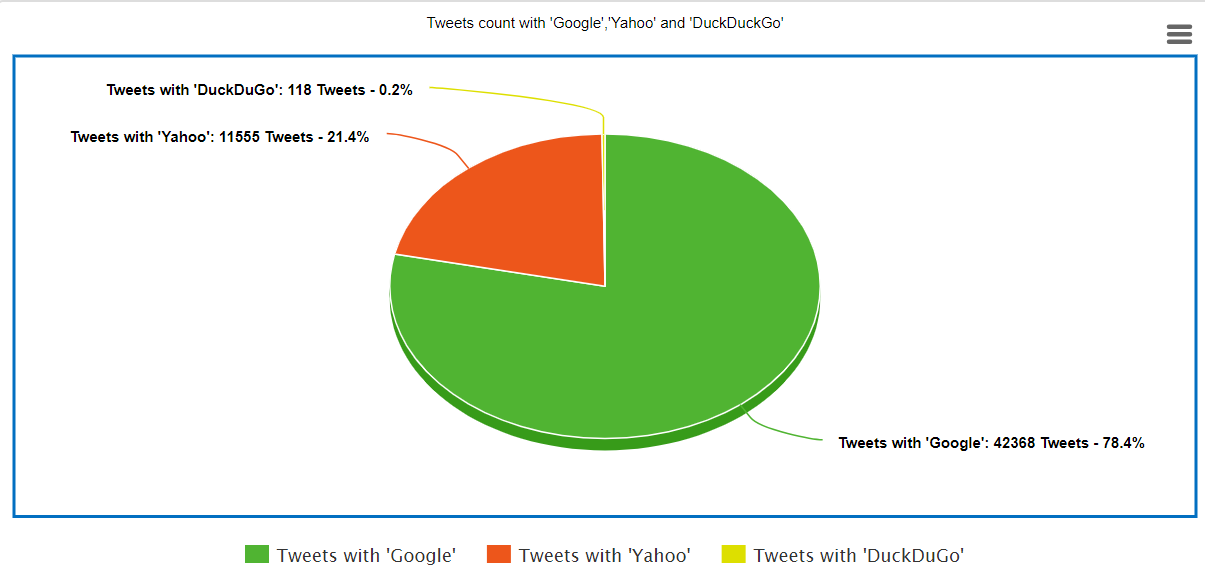
val query9j= sqlContext.sql("select count(\*) from DataTable where text LIKE '%yahoo%'")

val query9k= sqlContext.sql("select count(\*) from DataTable where text LIKE '%duckduckgo%'")

val query9= query9i.union(query9j).union(query9k)



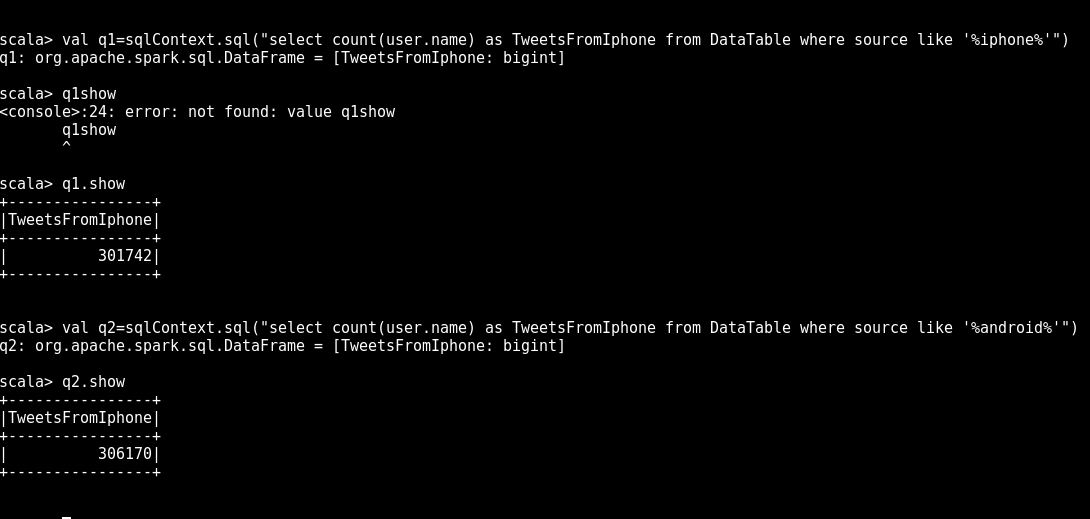
Graph for Query 9



10.Query to fetch number of users who tweeted from iphone and android

>val q1=sqlContext.sql("select count(user.name) as TweetsFromIphone from DataTable where source like '%iphone%'")

>val q2=sqlContext.sql("select count(user.name) as TweetsFromAndroid from DataTable where source like '%android%'")



**Graph for Query 10**

