**Twitter Data Analysis**

**Authored By**

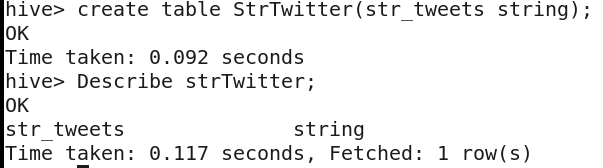
Divya Jayaprakash (dxj160830)

BUAN 6346 – Big Data Analytics

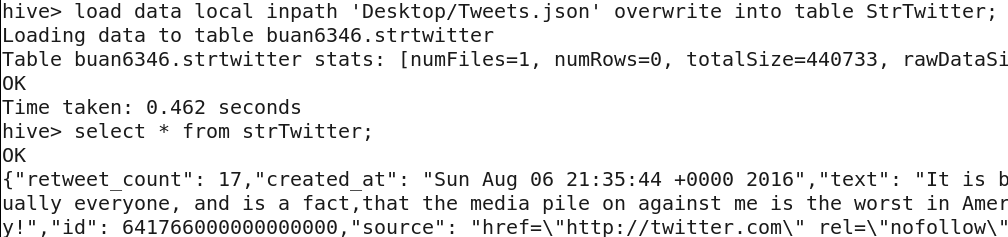
Prof. Kashif Saeed

**Twitter Data Loading Into Hive Table:**

**create table StrTwitter(str\_tweets string);**



**load data local inpath 'Desktop/Tweets.json' overwrite into table StrTwitter;**



**create table Twitter as select**

**get\_json\_object(str\_tweets,'$.retweet\_count') as retweet\_count,**

**from\_unixtime(unix\_timestamp(get\_json\_object(str\_tweets,'$.created\_at'),"EEE MMM d HH:mm:ss Z yyyy"), 'YYYY-MM-dd') as date,**

**get\_json\_object(str\_tweets,'$.text') as text,**

**get\_json\_object(str\_tweets,'$.id') as id,**

**get\_json\_object(str\_tweets,'$.source') as source,**

**get\_json\_object(str\_tweets,'$.user.location') as location,**

**get\_json\_object(str\_tweets,'$.user.id') as id1,**

**get\_json\_object(str\_tweets,'$.user.id\_str') as id\_str,**

**get\_json\_object(str\_tweets,'$.user.name') as name,**

**get\_json\_object(str\_tweets,'$.user.screen\_name') as screen\_name,**

**get\_json\_object(str\_tweets,'$.user.geo\_enabled') as geo\_enabled,**

**get\_json\_object(str\_tweets,'$.user.lang') as lang,**

**get\_json\_object(str\_tweets,'$.user.protected') as protected,**

**get\_json\_object(str\_tweets,'$.user.verified') as verified,**

**get\_json\_object(str\_tweets,'$.user.followers\_count') as followers\_count,**

**get\_json\_object(str\_tweets,'$.user.friends\_count') as friends\_count,**

**get\_json\_object(str\_tweets,'$.user.listed\_count') as listed\_count,**

**get\_json\_object(str\_tweets,'$.user.favourites\_count') as favourites\_count,**

**get\_json\_object(str\_tweets,'$.user.statuses\_count') as statuses\_count,**

**get\_json\_object(str\_tweets,'$.user.profile\_background\_color') as profile\_background\_color,**

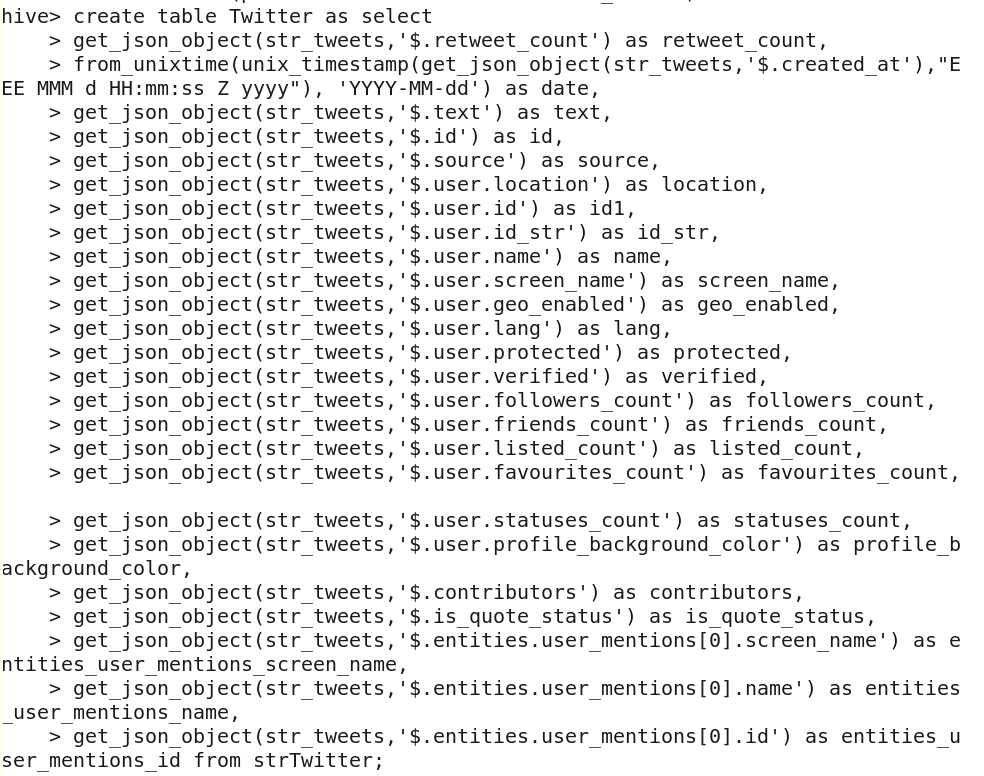
**get\_json\_object(str\_tweets,'$.contributors') as contributors,**

**get\_json\_object(str\_tweets,'$.is\_quote\_status') as is\_quote\_status,**

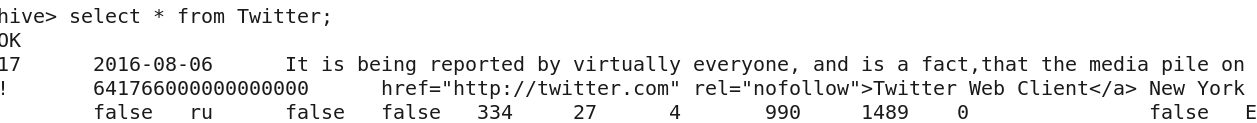
**get\_json\_object(str\_tweets,'$.entities.user\_mentions[0].screen\_name') as entities\_user\_mentions\_screen\_name,**

**get\_json\_object(str\_tweets,'$.entities.user\_mentions[0].name') as entities\_user\_mentions\_name,**

**get\_json\_object(str\_tweets,'$.entities.user\_mentions[0].id') as entities\_user\_mentions\_id from strTwitter;**



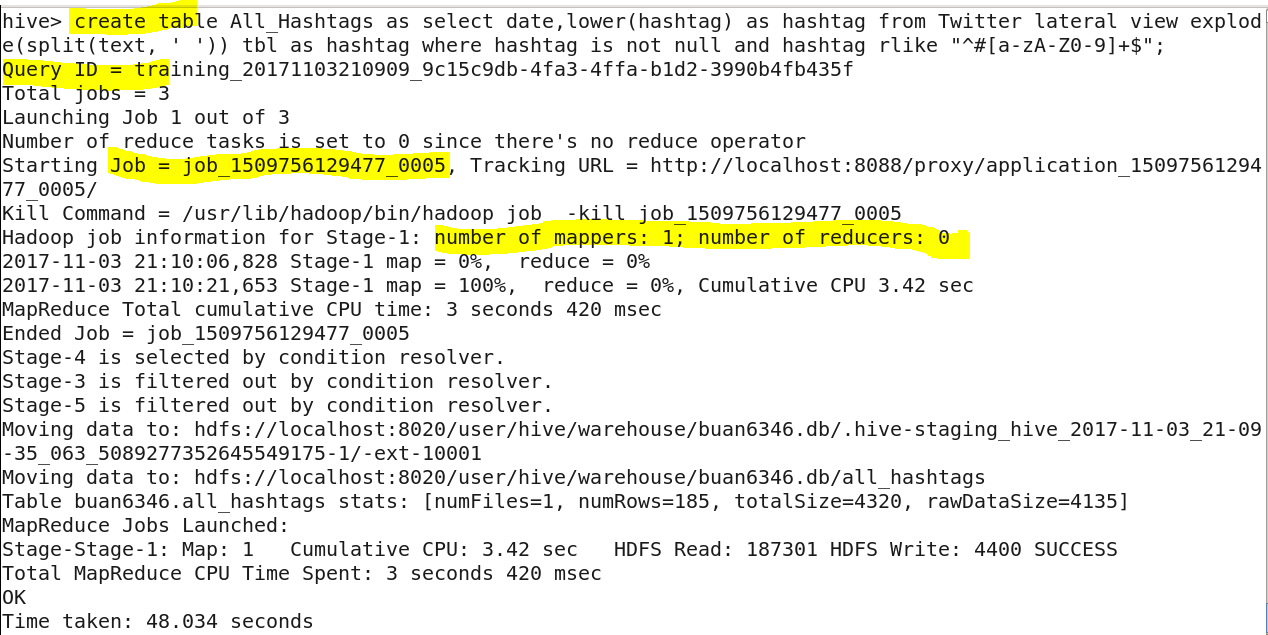
**Select \* form Twitter;**

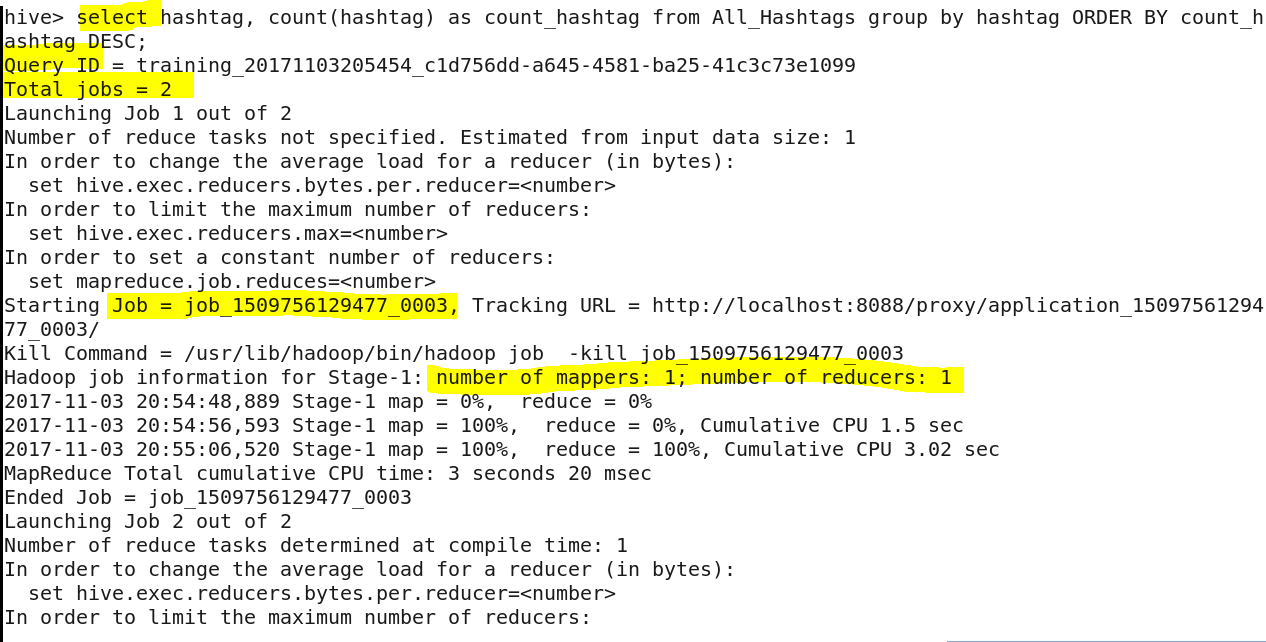


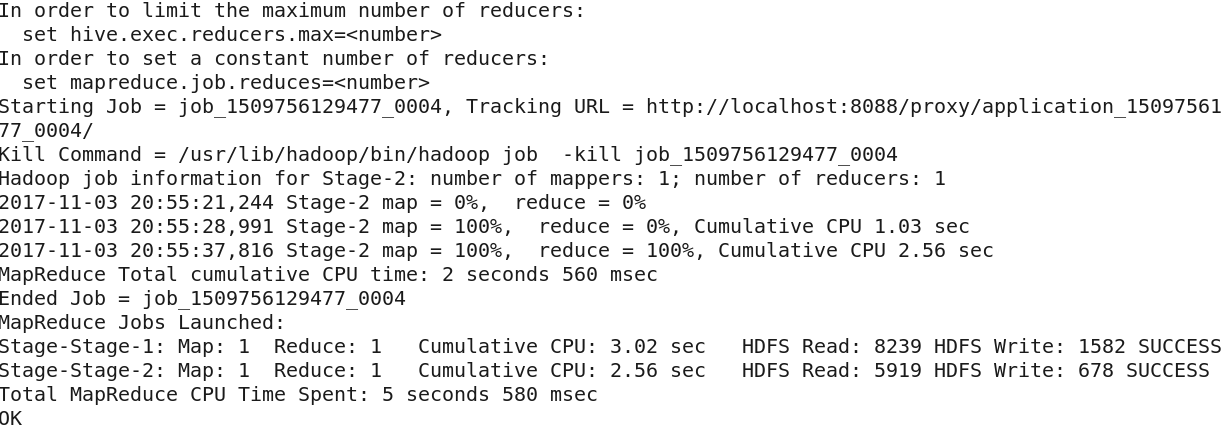
**1) a) What are the hashtags tweeted and how many times are they used?**

**create table All\_Hashtags as select date,lower(hashtag) as hashtag from Twitter lateral view explode(split(text, ' ')) tbl as hashtag where hashtag is not null and hashtag rlike "^#[a-zA-Z0-9]+$";**

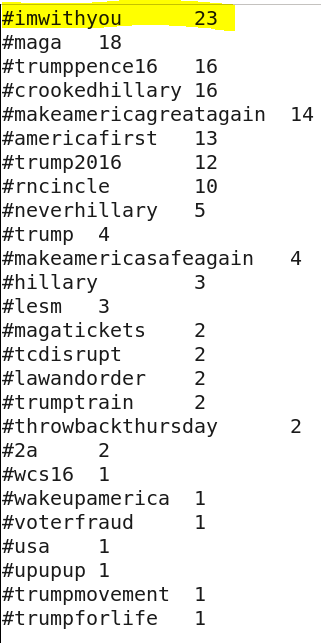
**select hashtag, count(hashtag) as count\_hashtag from All\_Hashtags group by hashtag ORDER BY count\_hashtag DESC;**

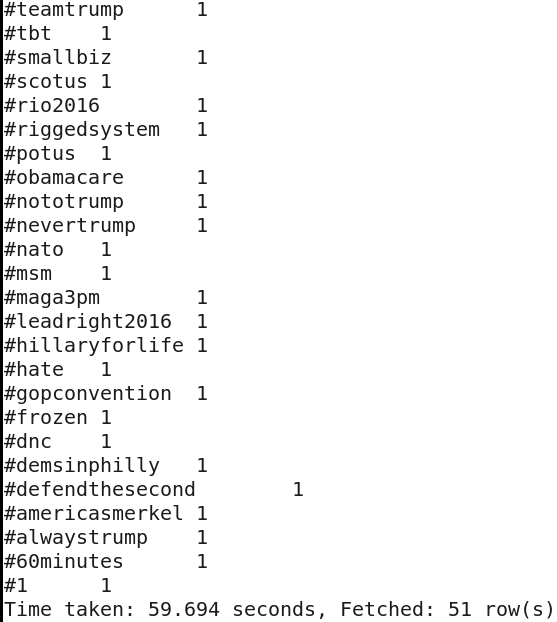






**Answer:**





**1) b) What is the most trending hashtag in a day and how many times are they tweeted? [Note: day should be in the format ‘yyyy-mm-dd’]**

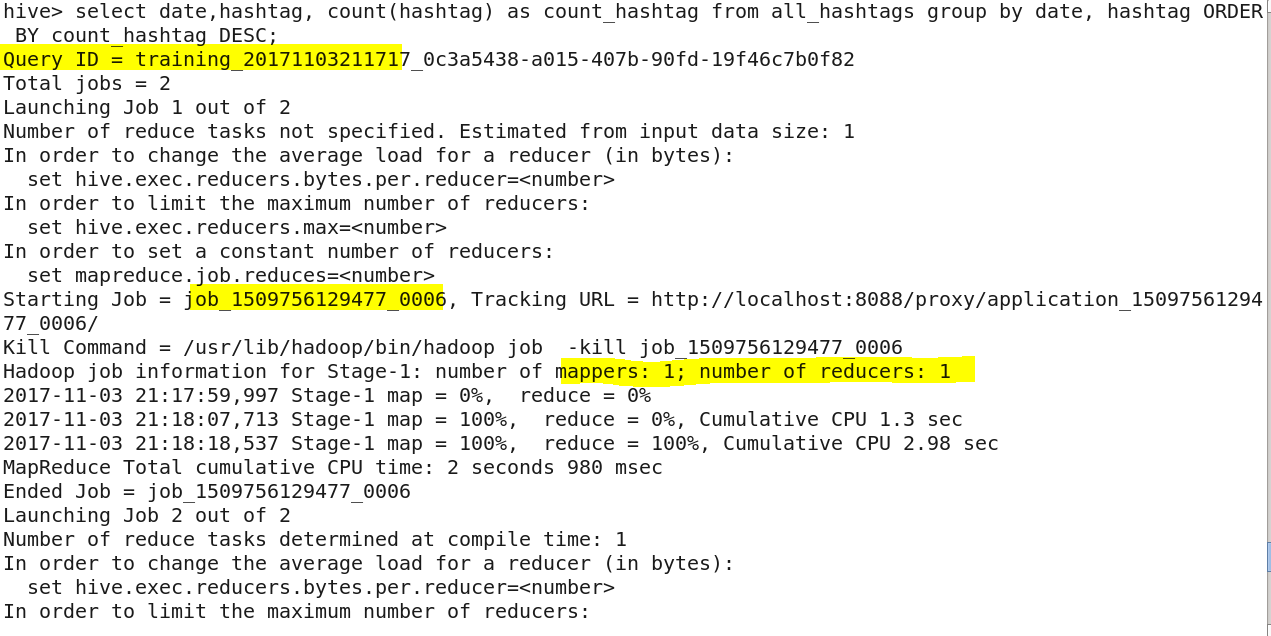
**Select date,hashtag, count(hashtag) as count\_hashtag from all\_hashtags group by date, hashtag ORDER BY count\_hashtag DESC;**

**Answer:**

**For each day, the most trending hashtags for that day has been listed below.**

**On 20 th September 2011, the #imwithyou has been tweeted twice.**

**This is the max tweet a twitter handle has got in a day.**







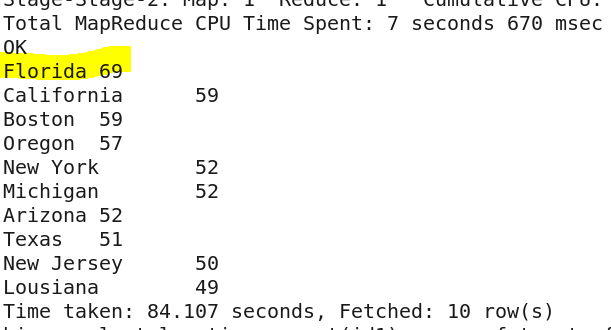
**1) c) Which state users are most active, and how many tweets are posted by them?**

**select location, count(id) as no\_of\_tweets from twitter group by location order by no\_of\_tweets DESC;**

**Answer:**

**Florida state has the most active users. The number of tweets posted by them is 69.**





**1) d) What is the total number of favorites received for each hashtag?**

**create table Favourites as select favourites\_count,lower(hashtag) as hashtag from Twitter lateral view explode(split(text, ' ')) tbl as hashtag where hashtag is not null and hashtag rlike "^#[a-zA-Z0-9]+$";**

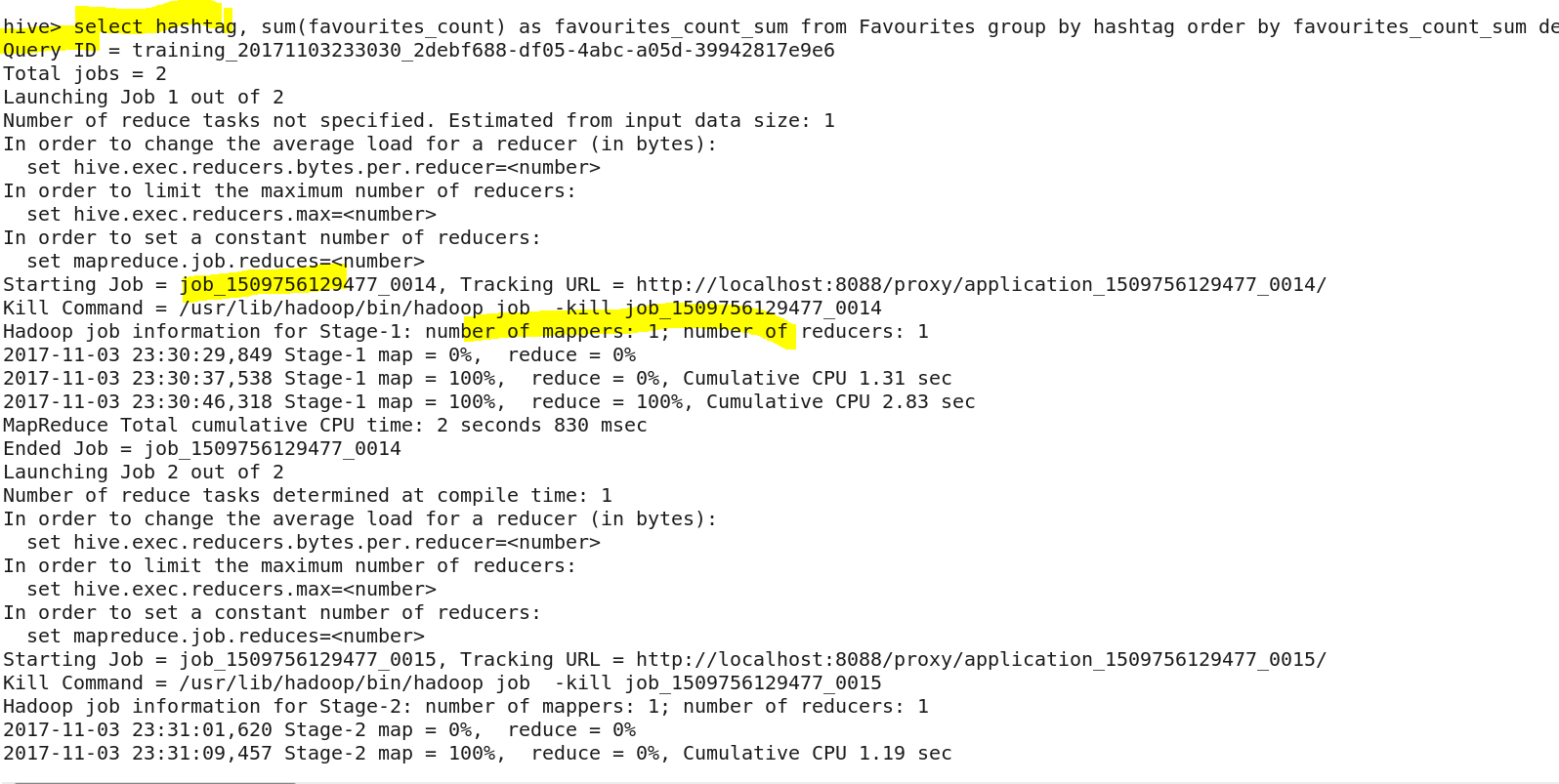
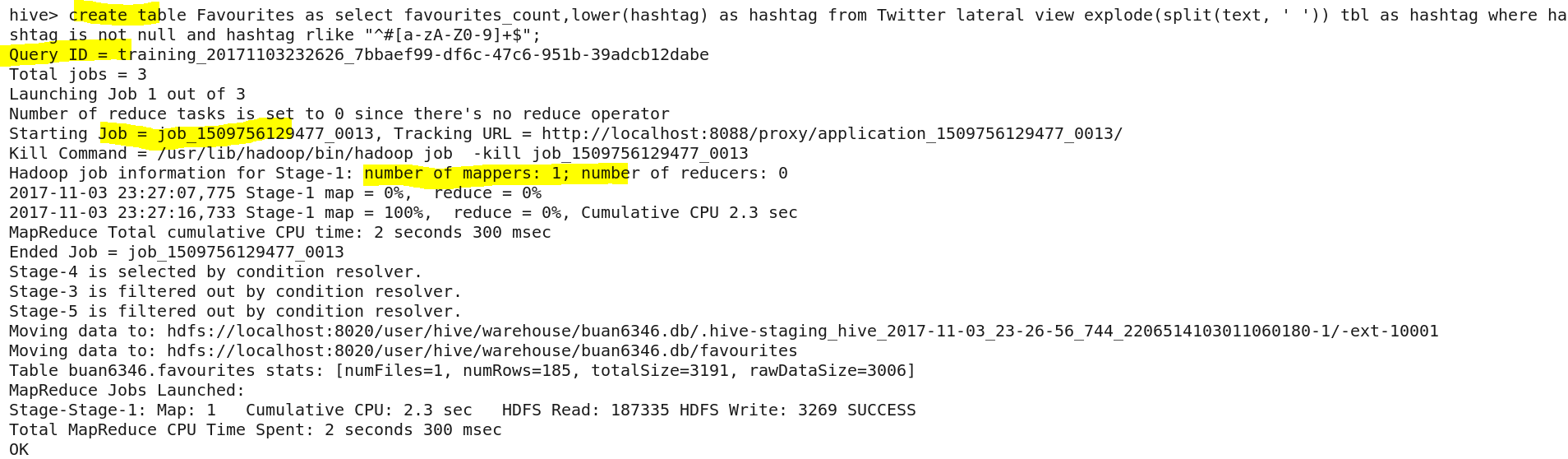
**select hashtag, sum(favourites\_count) as favourites\_count\_sum from Favourites group by hashtag order by favourites\_count\_sum desc;**

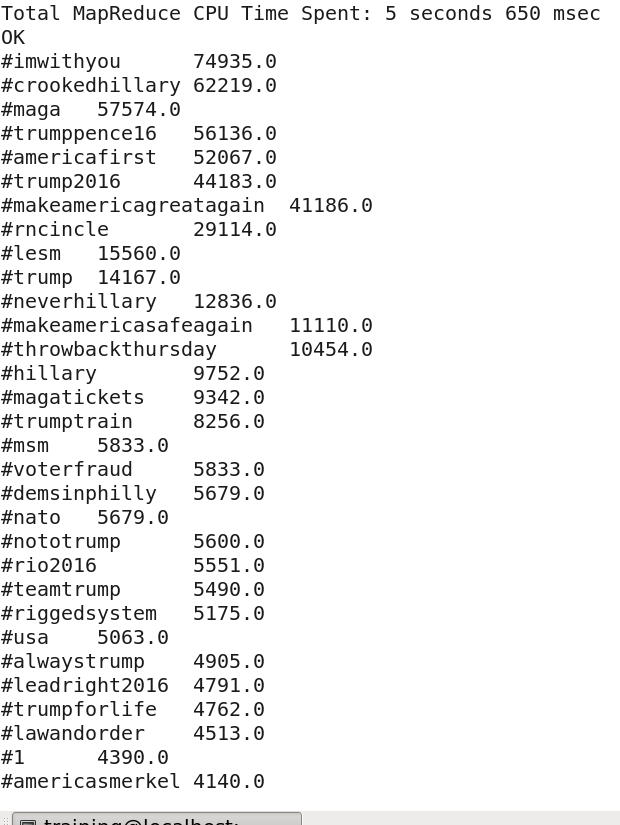
**Answer:**

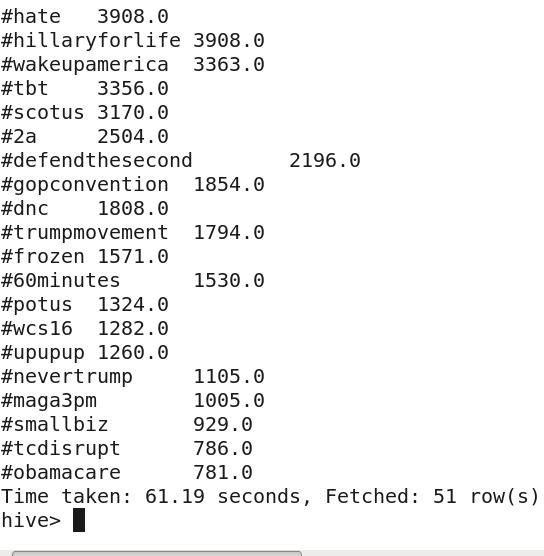
**The total number of favorites received for each hashtag is listed below.**

**#imwithyou hashtag tweets has 74935 favourites.**

**#Obamacare has 781 favourites.**







**1) e) Does each tweet have a positive or negative sentiment?**

**Note: Include the date in the format ’yyyy-mm-dd’, with tweet id, user name and the score.**

**create table dictionary(word string,score int)**

**row format delimited**

**fields terminated by ' \t';**

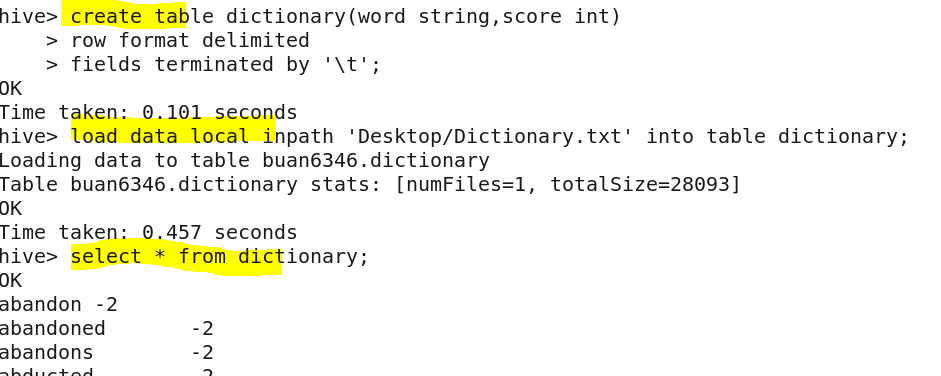
**create table Tweet\_Words as select name, date, id, lower(tweet\_word) as tweet\_word from Twitter lateral view explode(split(text, ' ')) tbl as tweet\_word;**

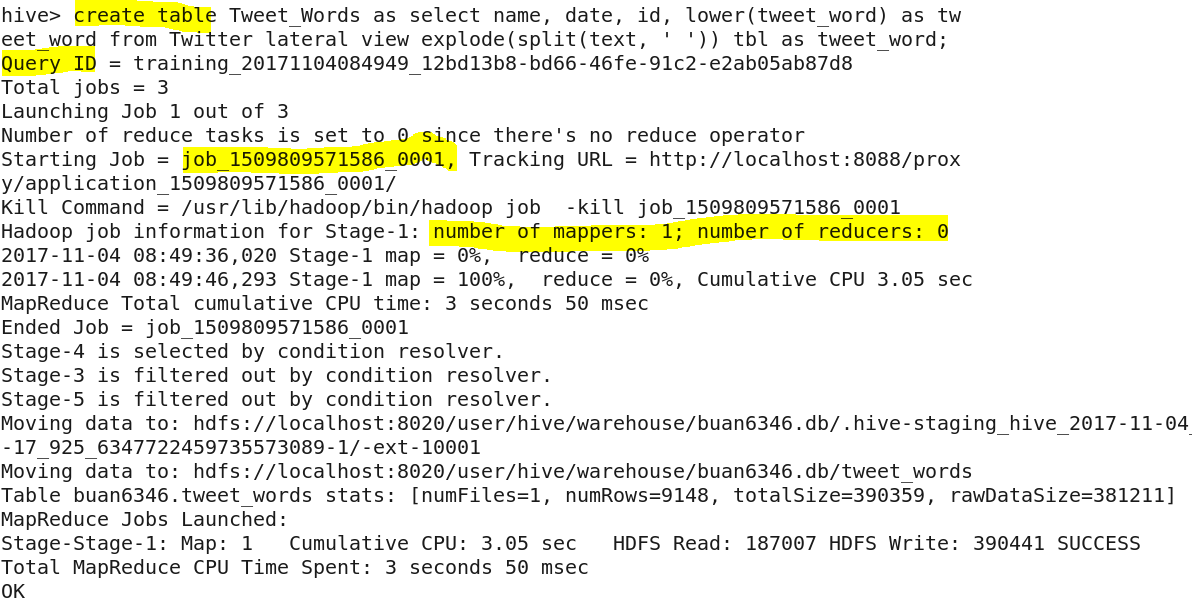
**create table Tweet\_Word\_Score as select tw.date, tw.id, tw.name, tw.tweet\_word, d.score from tweet\_words tw left outer join dictionary d on(tw.tweet\_word=d.word);**

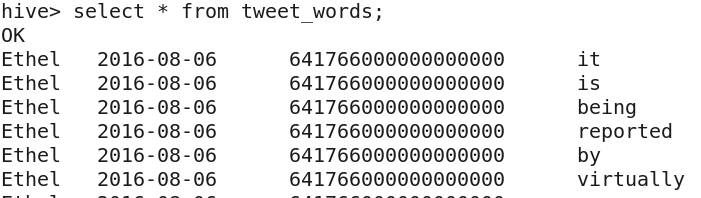
**select date, name, id, avg(score) AS tweetscore from Tweet\_Word\_Score group by date, name, id order by tweetscore DESC;**

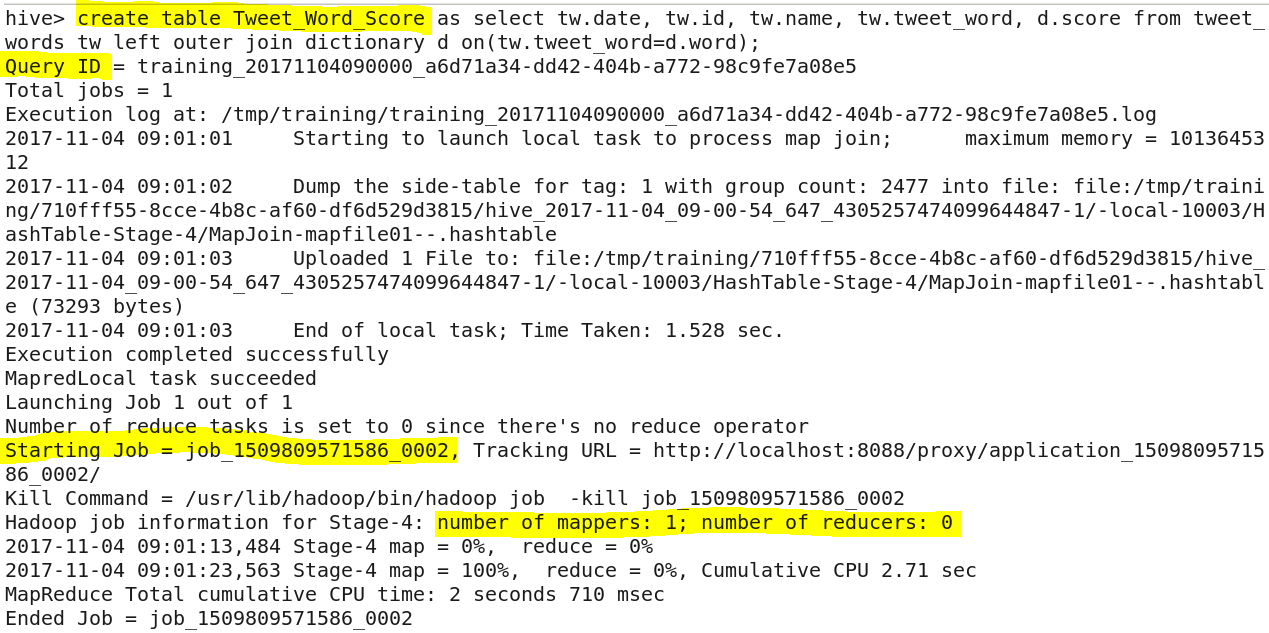
**Answer:**

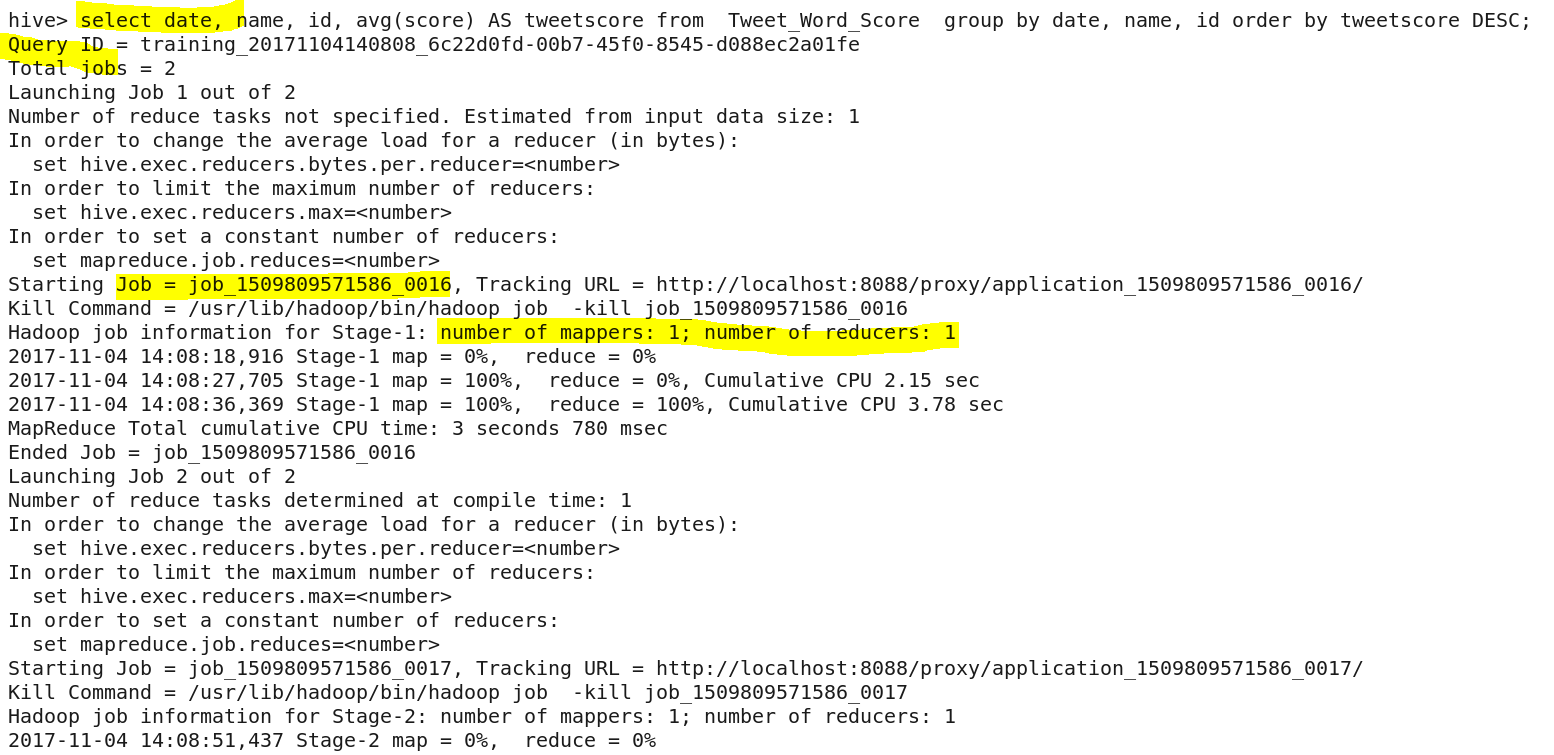
**No! Most of the tweets have positive or negative sentiments. The average of the score of the words in every tweet shows the positive or negative sentiment. Yet, for some tweets, the words donot have score. The sentiment of those tweets could not be computed.**

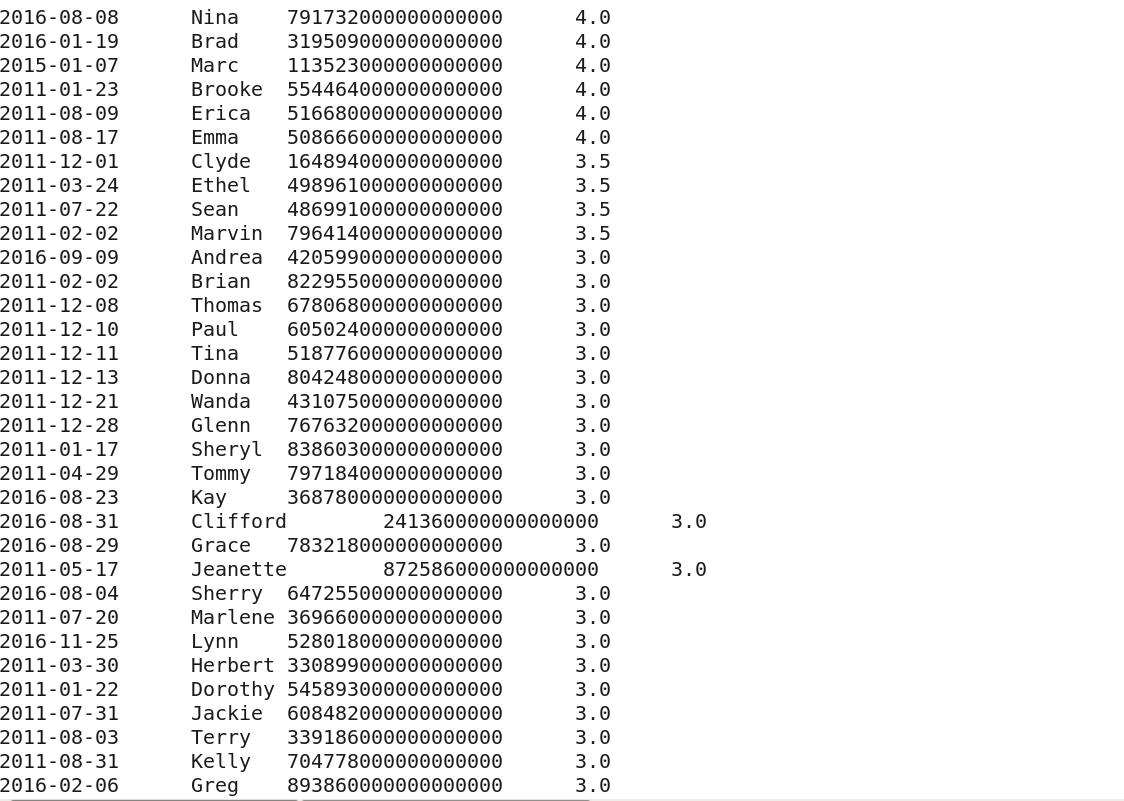


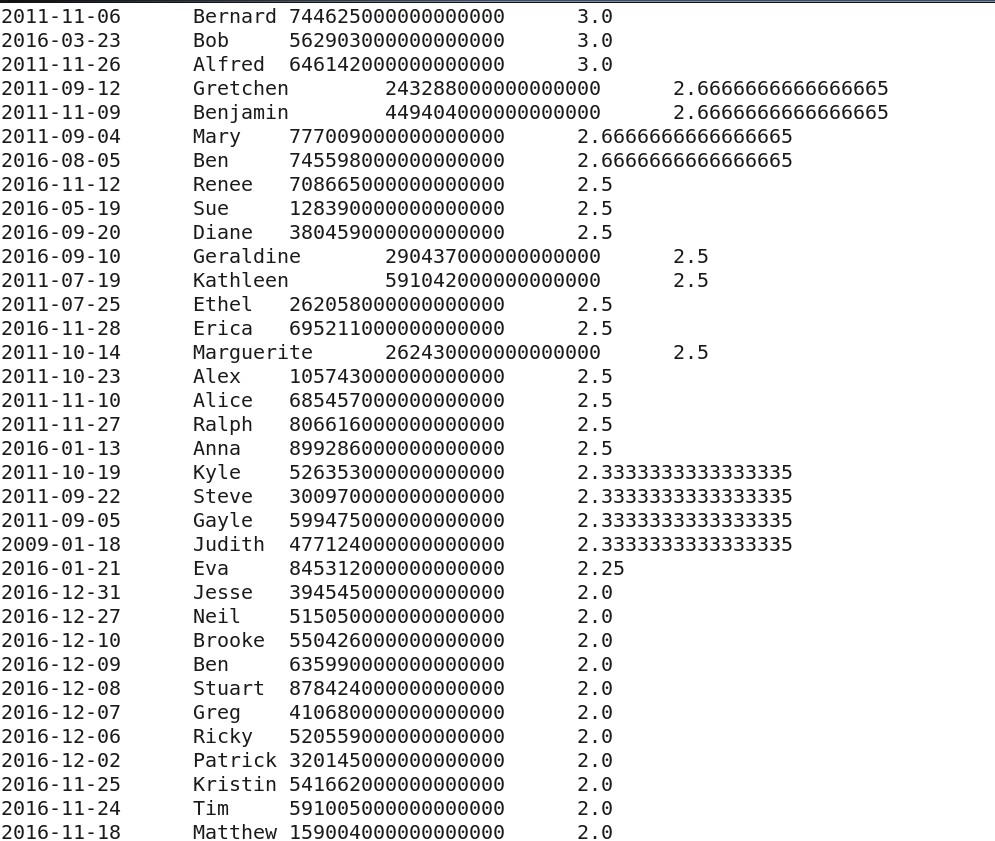


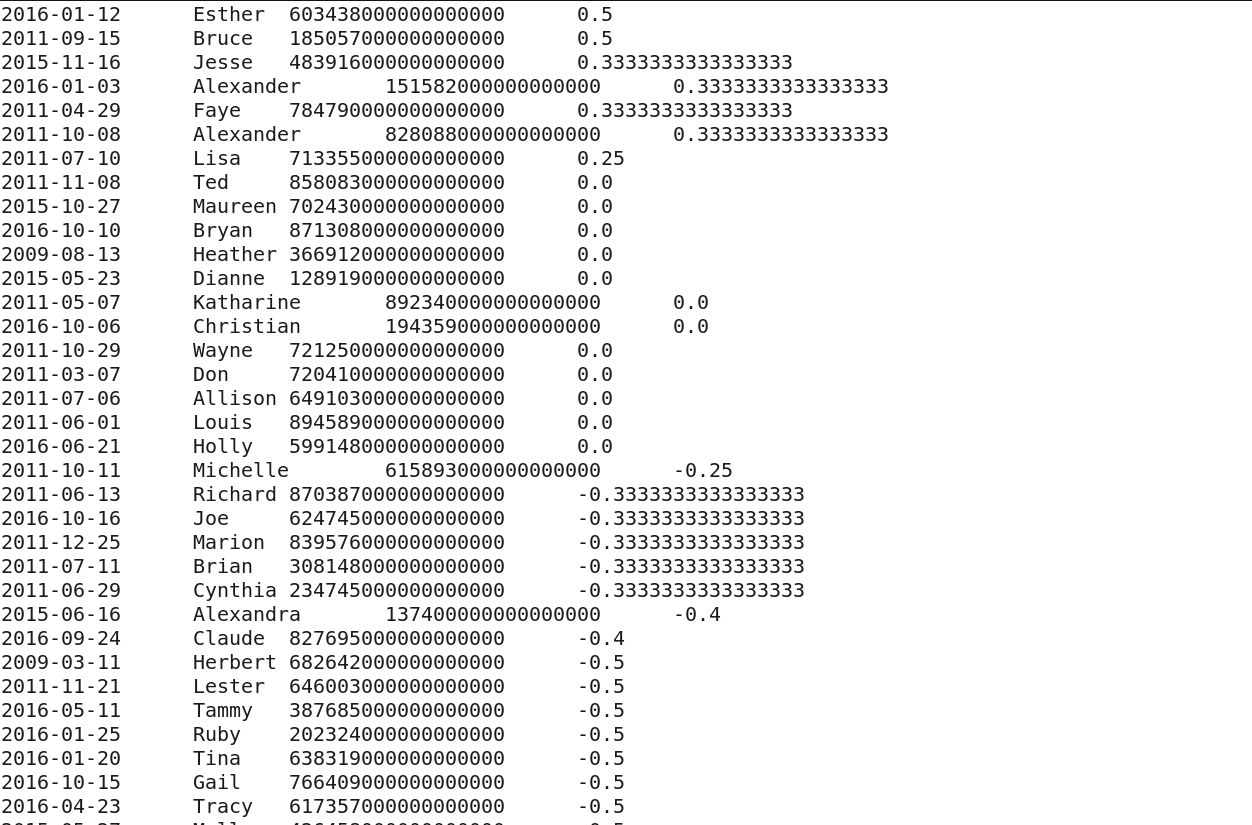


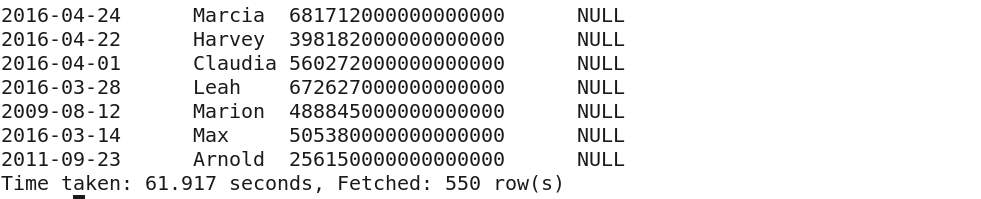












**1) f) Do you believe that the process outlined above has issues? If so, briefly explain how the process can be improved.**

**1) The dictionary scoring cannot be 100% accurate**. **In English context , a tweet may be positive but algorithm may predict it be negative.**

**Example 1:**

E.g. The word breathtaking has a score of 5. The word ‘not’ is not scored in the dictionary.

Suppose a tweet is like ‘…not breathtaking..’,

As of the current scoring system, the tweet is classified to have positive sentiment. But it is actually negative.

**Solution: Every word in a tweet should be present in the dictionary**

**Example 2:**

Say there is a tweet,

#Trump2016 Having trump government is not bad.

This is positive sense.

But the words ‘not’ and ‘bad’ may be scored too negative.

**Solution: May be we should write special complex queries to identify ‘not’ before negative words which actually makes it positive.**

**2) User sarcasms can be identified only by high definition algorithms and not by our approach**

**3) A tweet can be positive to one person but can be negative to another person. Probably we should track user’s tweeting history to figure out his/her sentiments.**

**4) Many words in the tweets were not listed in the dictionary at all. So the scores for those words were null.**

**Solution:**

Either the dictionary should be better to cover more words

Or

The inner join should be used between Twitter word table and the dictionary table to obtain the tweet score based on only that words available in the dictionary

**5) Certain words have symbols and numbers preceding or succeeding them.**

After split on space, such words makes no sense as per the dictionary. They cant be scored! Proper substring should be taken from the tweets and joined with dictionary word scores

e.g) enjoy!

stay.

fools,

**Solution:**

Data pre-processing is very essential.

We should not have done sentiment analysis on the raw data.

**6) In the data, the favourites count is given under user details.**

The ‘Like’ count should have been given in the data for every tweet and not under the user struct

**2) Bonus Question:**

**Twitter data comprises data types that are hard to handle at times. Using the same twitter data, display all the rows with the following columns: entity Id, entity screen\_name, Text, Date, user location, date. {Note: Include the date in the format ’yyyy-mm-dd’}**

**select entities\_user\_mentions\_id as Entity\_Id, entities\_user\_mentions\_screen\_name as Entities\_Screen\_Name,text,date,location from Twitter;**

