COVID-19 TWITTER ANALYSIS

COURSE: BIG DATA PROGRAMMING

TEAM 4

FINAL REPORT

Vidyullatha Lakshmi Kaza- 8

Aparna Manda- 11

Lohitha Yenugu- 19

Introduction:

Data analysis on tweets pertaining to COVID19. The entire world is shutdown due to the virus and we wanted to know the public opinion on this situation. So, we collected their opinion through tweets. Collected real-time tweets talking about the corona virus with keywords- COVID19, Corona and performed analysis using big data technologies- Map Reduce, Hive, Cassandra and Map Reduce sentimental analysis.

Background:

Analyzed twitter data- json structure to extract useful attributes

<https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json>

Twitter data preprocessing to remove characters like spaces, new lines and commas that might cause issues during csv encoding.

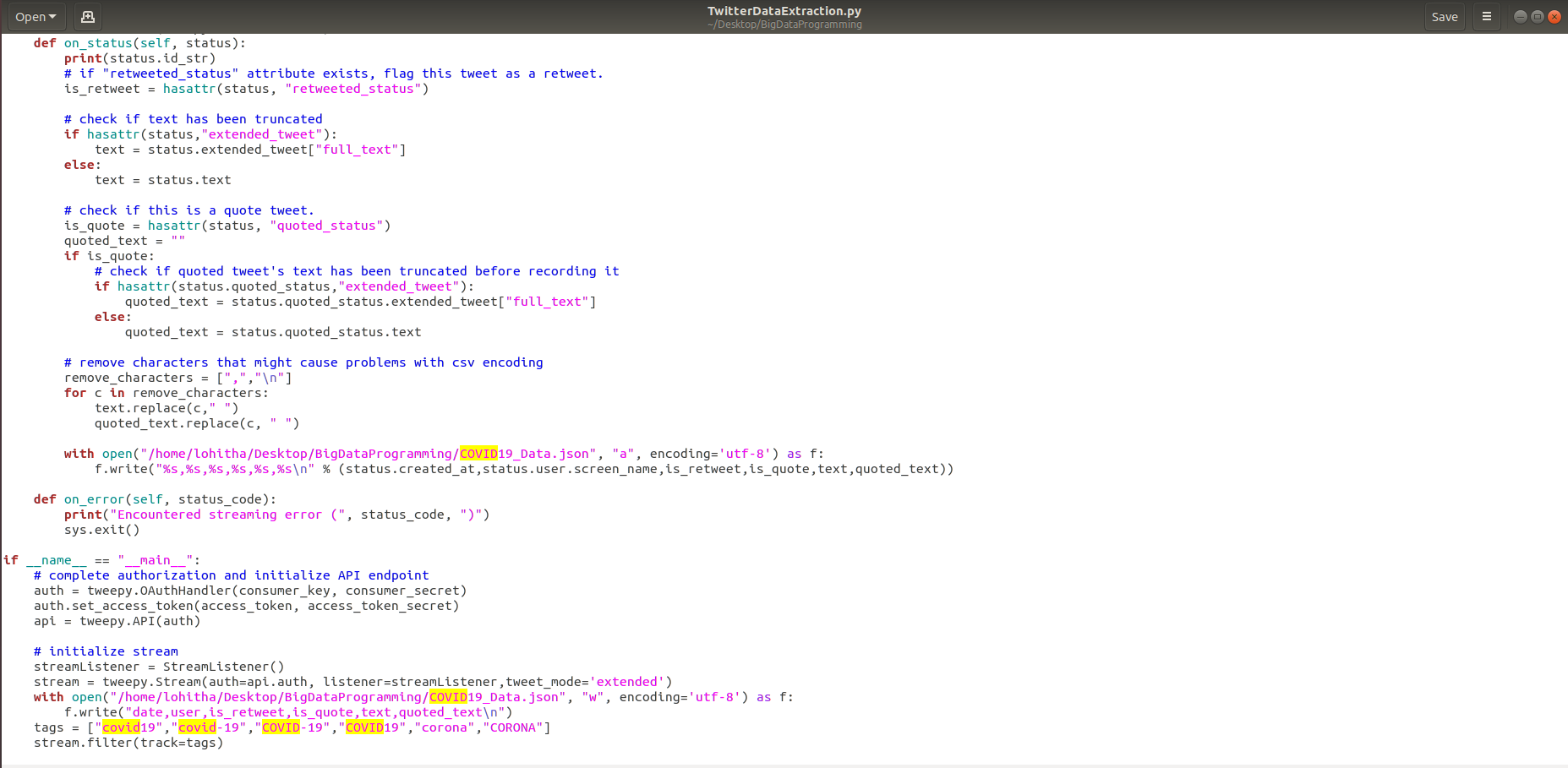
Features developed:

1. Twitter data collection on COVID-19
2. Map Reduce to count tweets by each user
3. Data Analysis of tweets using Hive
4. Sentiment Analysis of tweets using Map Reduce
5. Data Analysis of tweets using Cassandra
6. Twitter Data Analysis using Spark SQL

Data Collection:

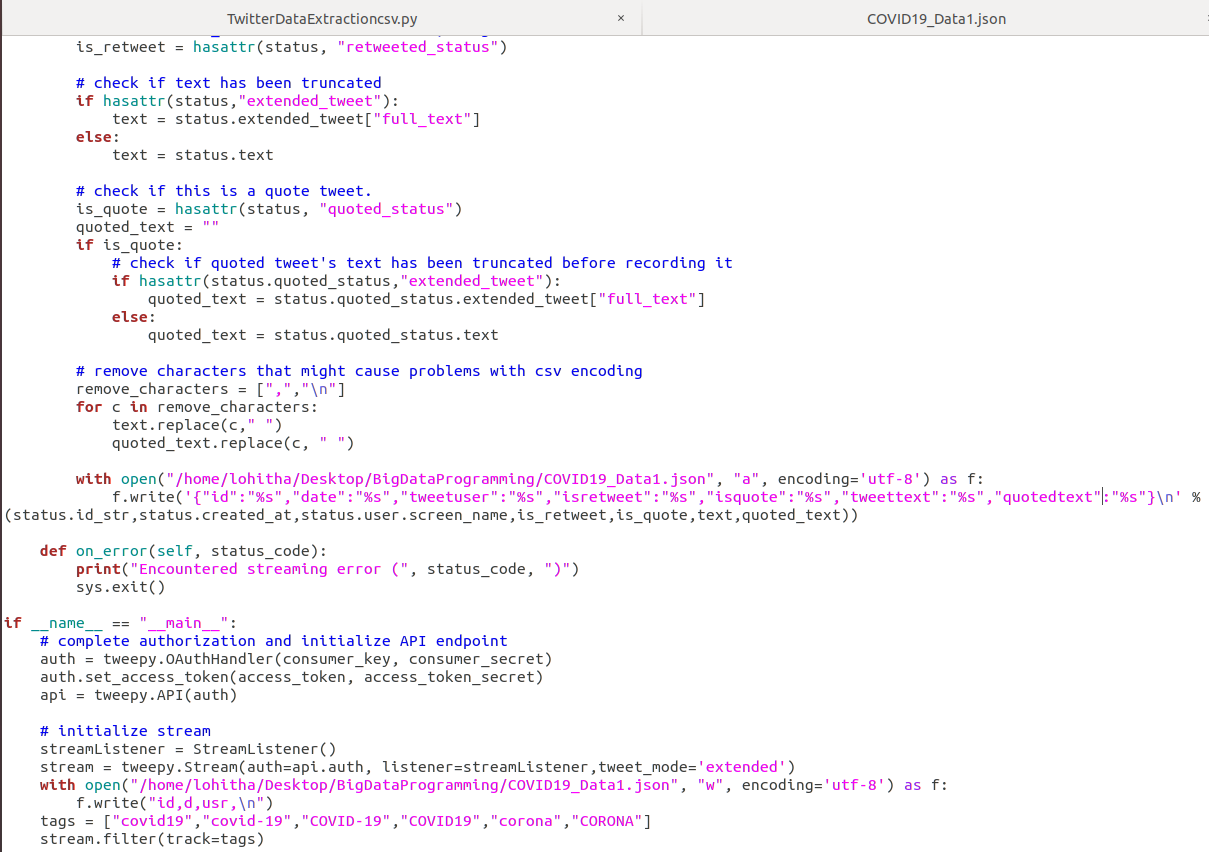
Python Code to extract live stream twitter data as CSV:

We used ‘Tweepy’ to connect to the Twitter API. And used authentication tokens from the Twitter developer account for authentication and connecting to Twitter API. To collect data on Covid, we are using tags- [“covid19”,”covid-19”,”corona”,”covid19”] to filter data. When data is received, we are checking if data has necessary attributes of interest. We are also preprocessing the data by removing characters like ‘,’ and spaces. Removing ‘,’ helps in loading the data to appropriate columns from csv file to the tables. Finally, opening the file in append mode to write the data being collected through StreamListener.



Python Code to extract live stream twitter data as JSON:

To collect only the tweet attributes of our interest, we are manually extracting the necessary tweet attributes and writing attributes in JSON format manually.



Collected Tweets:

Below are the sample tweets collected from the above code. Here, each tweet has attributes- date, user, is\_retweet, is\_quote, text, quoted\_text.



Dataset:

Collected real-time tweets using twitter streaming api- tweepy. Extracted features - tweet\_date, tweet\_user, tweet\_text, is\_retweet, is\_quote, quoted\_text.

tweet\_date – timestamp

tweet\_user – text

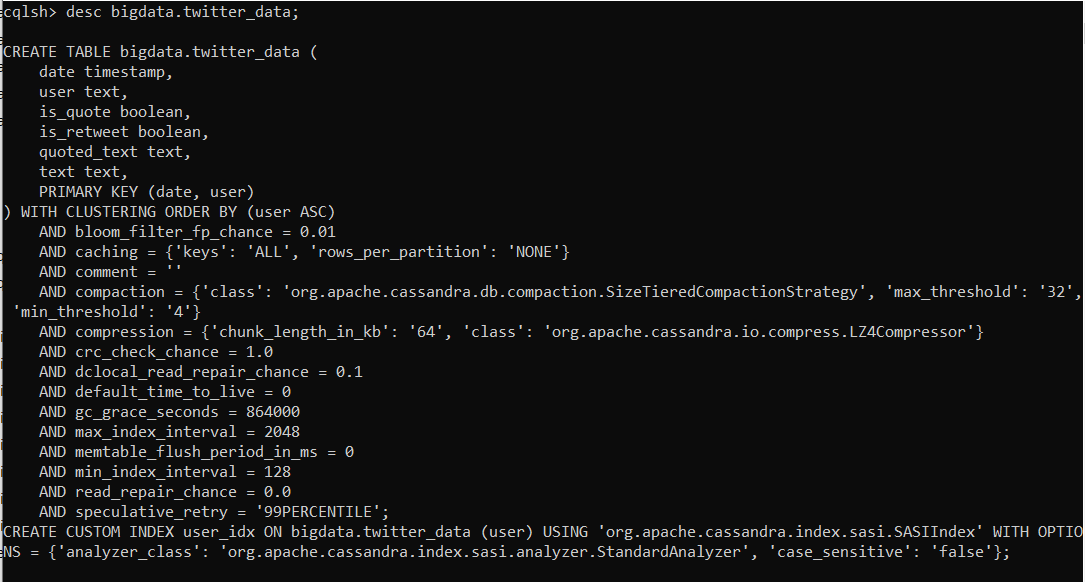
tweet\_text – text

is\_retweet – boolean

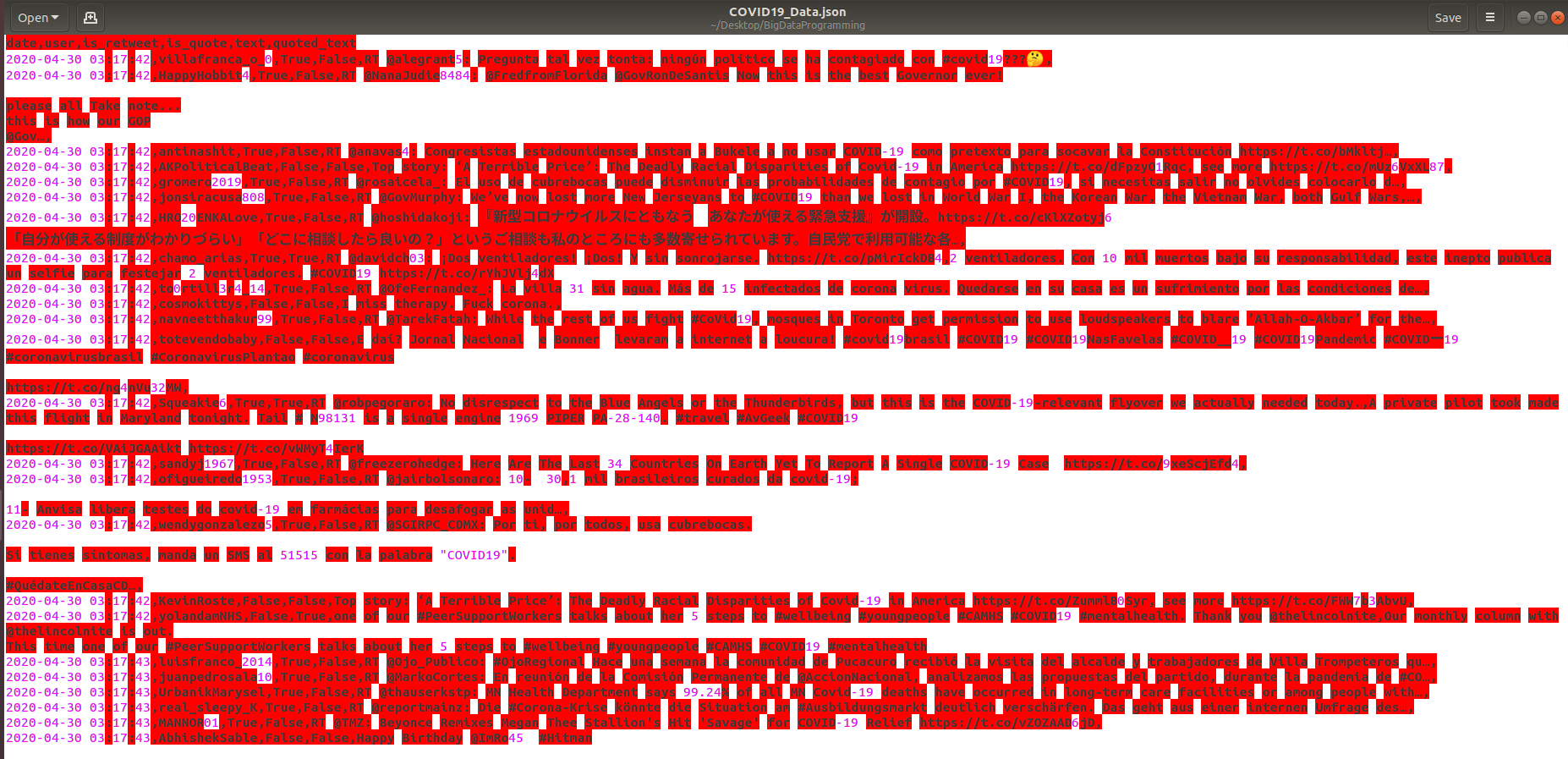
is\_quote – boolean

quoted\_text – text

Here, we are displaying the table metadata- columns, datatypes, compression used, indexes created on table- twitter\_data.



Below is the snapshot of data loaded into the Cassandra table. The size of the dataset collected during increment is 50MB and later collected 200MB.



Analysis of data/Implementation/Results:

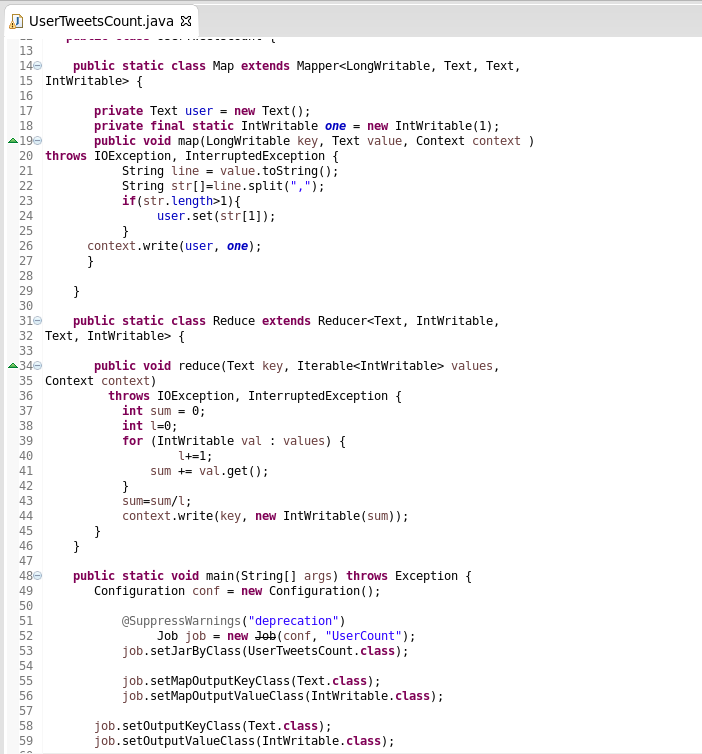
**Use Case 1:** Map Reduce to count tweets by each user

This is the mapreduce program to analyze the tweet frequency from each user i.e., we are counting the number of tweets from each user account and saving the results to HDFS.

In the main method, we are creating job “UserCount” and setting required job parameters. setMapOutputKeyClass- tells the mapreduce job about the expected type of the final output key which is here username. So, we are setting this class to ‘Text’. setMapOutputValueClass- tells the mapreduce job about the expected type of the final output value which is here count. So, we are setting this class to ‘IntWritable’.

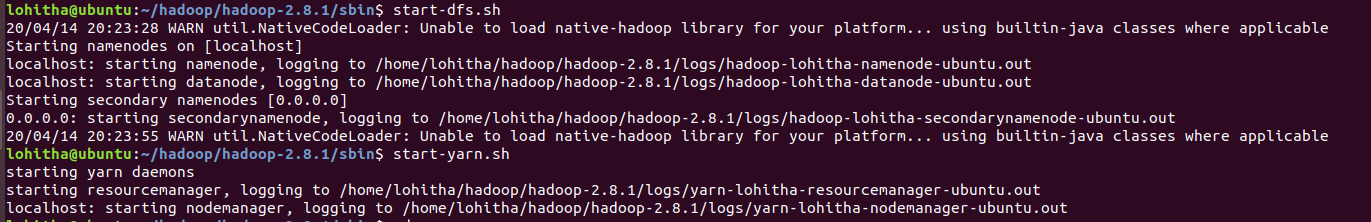
Map class extends the generic class Mapper with types <LongWritable, Text, Text, IntWritable> where <Longwritable,Text> are input types to the map method representing input file offset and file data. <Text,IntWritable> are map method output types representing username and count. Since the file contents are comma separated, we are splitting line by ‘,’ and checking if the tweet length is at least 2 and writing <username,1> to the context.

Reduce class extends the generic class Reducer with types <Text, IntWritable, Text, IntWritable> where <Text, IntWritable> are input types to reduce method i.e., output of map method. Here key is username and we are adding the value of same username to give <username,value> as output.



Starting HDFS namenode and datanodes:

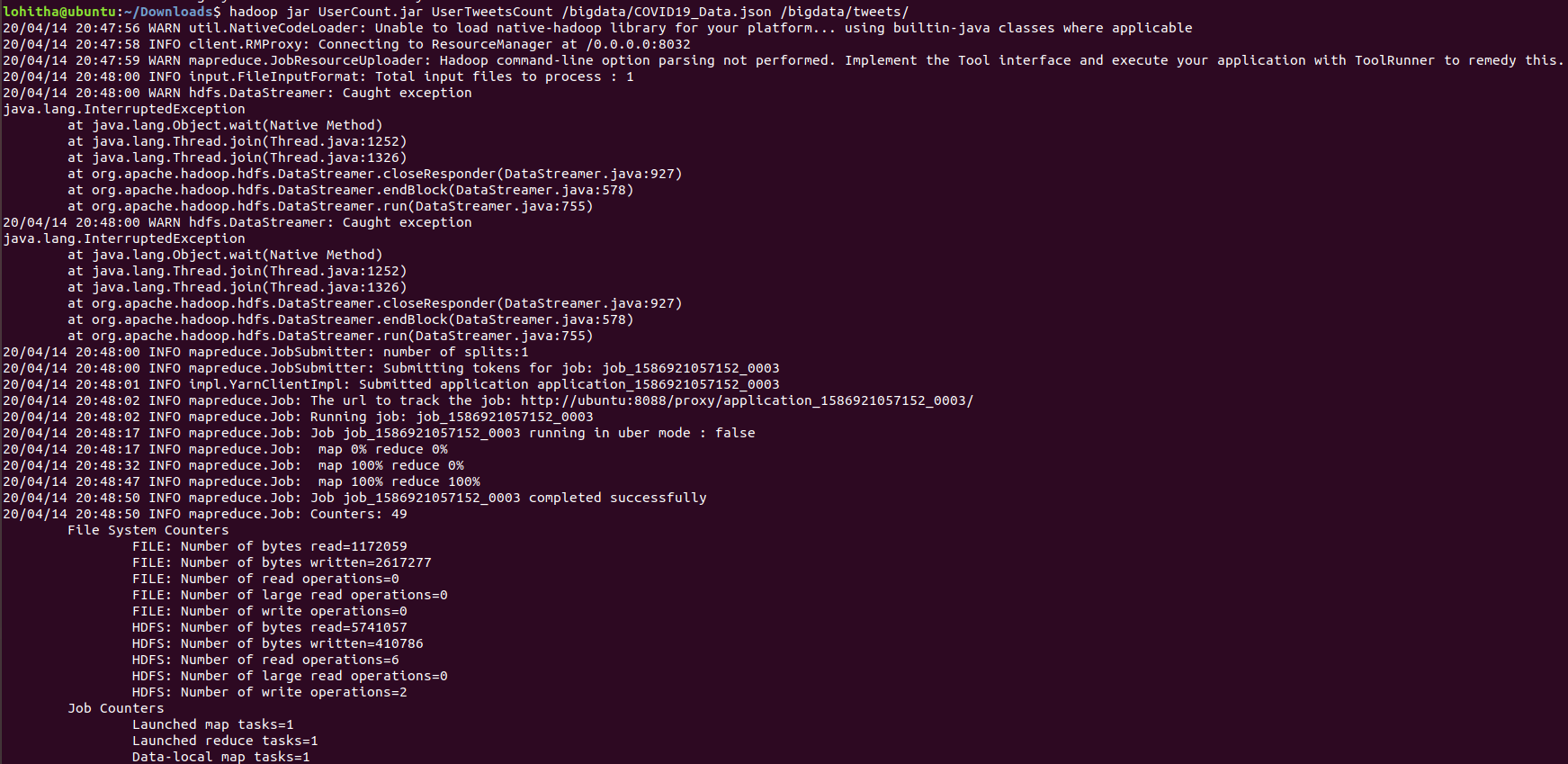
We are starting the HDFS daemons- namenode and datanode to perform operations on HDFS. Dataset is loaded to the HDFS directory. The above mapreduce program jar file is then executed with arguments input file location and output file location HDFS locations.

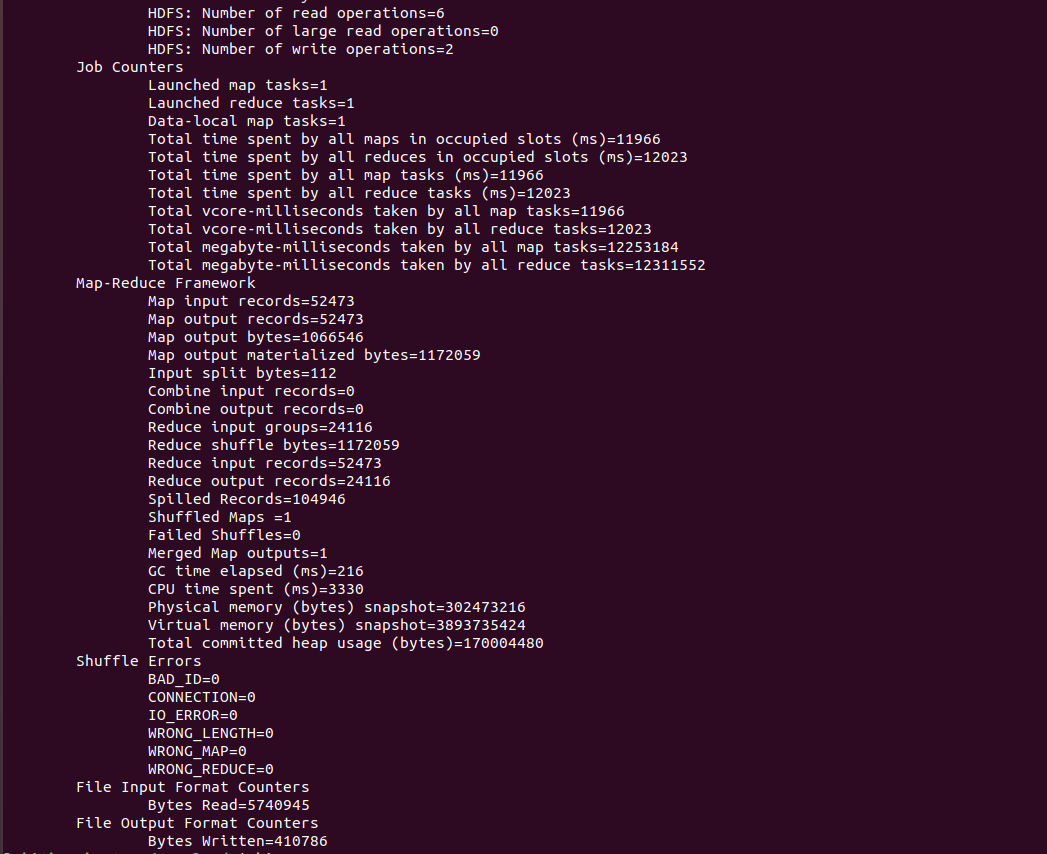


Loading input data from local to HDFS:



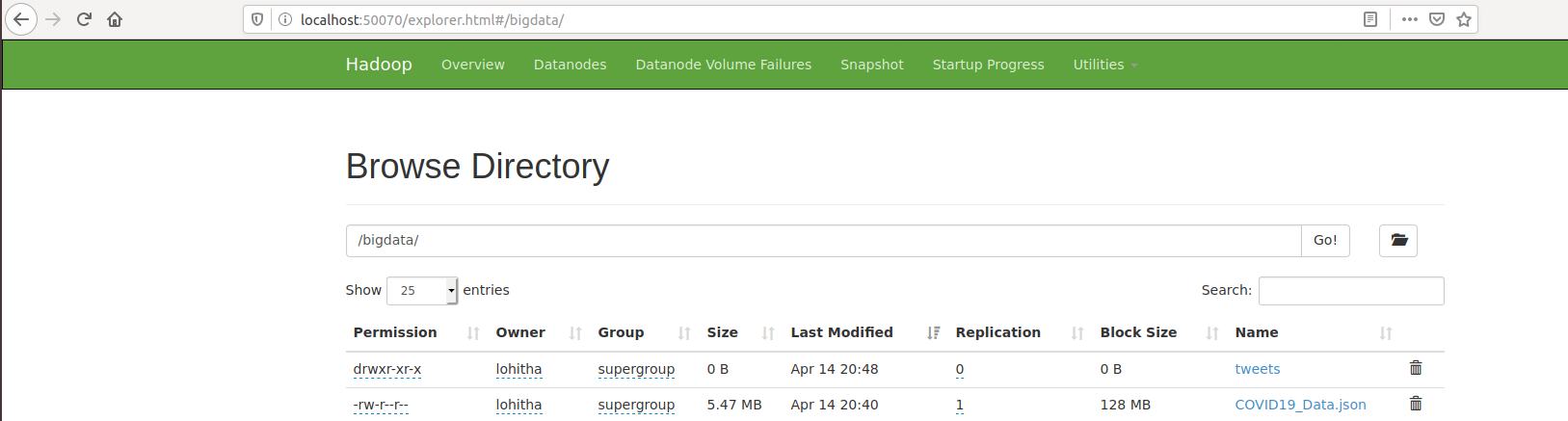
Running MapReduce:

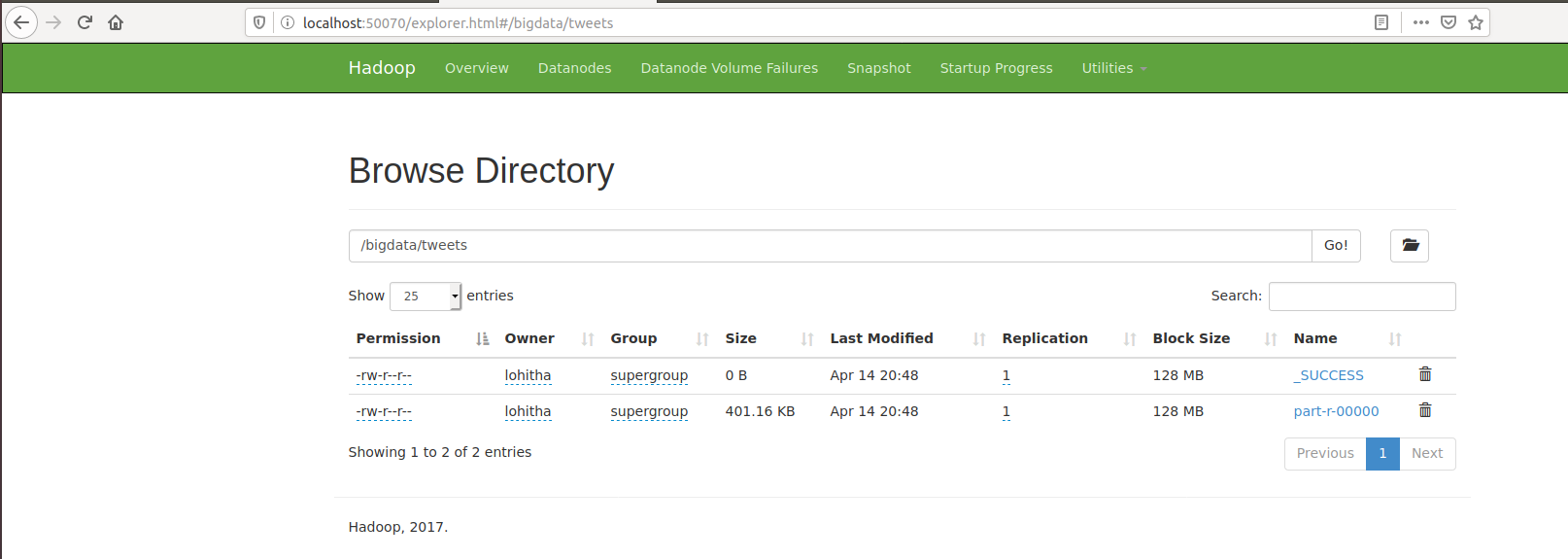




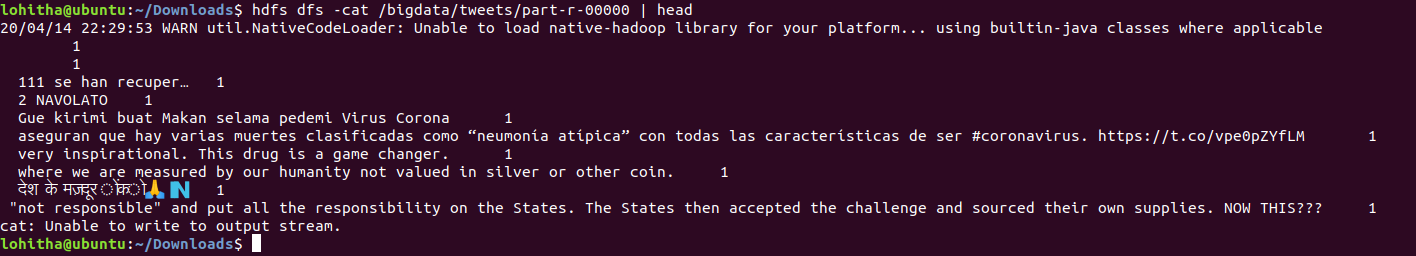
From the above map reduce job, all the file records are processed without any errors and are written to HDFS directory /bigdata/tweets.

HDFS File Structure:



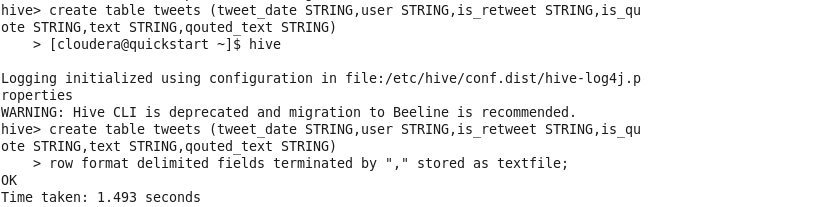


Output File:

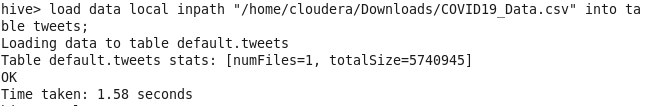


**Use Case 2:** Data analysis of tweets using Hive

Create Tweets Table:



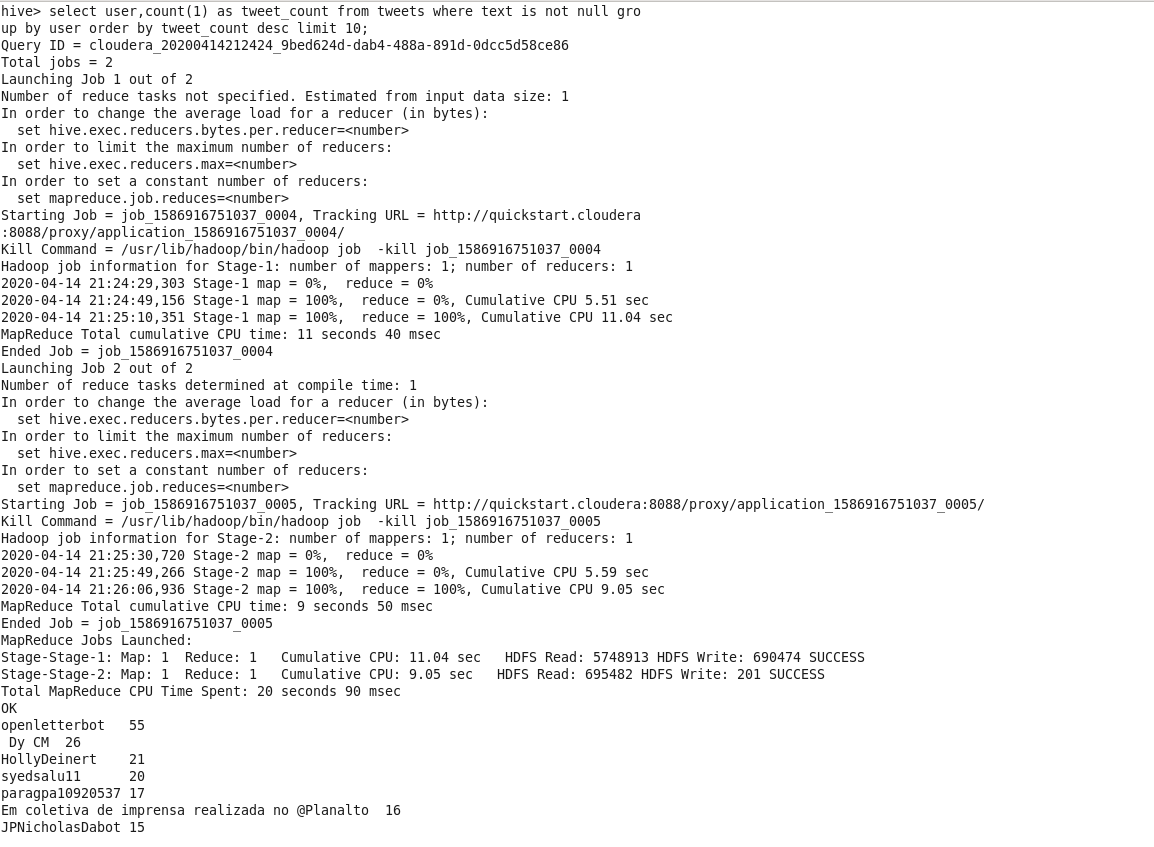
Load Twitter Data into Tweets table:

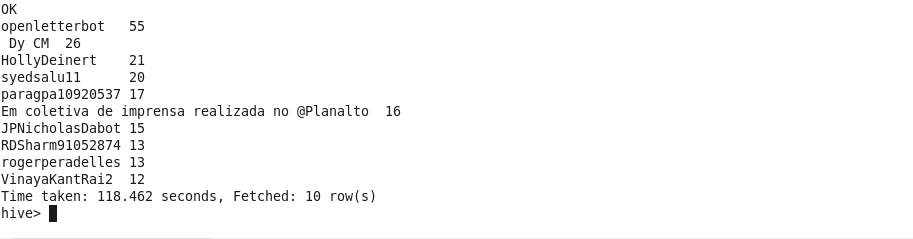


Query 1: Fetch top users with more number of tweets

SELECT user, COUNT(1) AS tweet\_count FROM tweets WHERE text IS NOT NULL GROUP BY user ORDER BY tweet\_count DESC LIMIT 10;

The above query is run to see the users tweeting more number of tweets on COVID. Highest tweets are tweeted/retweeted/quoted by openletterbot summing to 55 tweets.

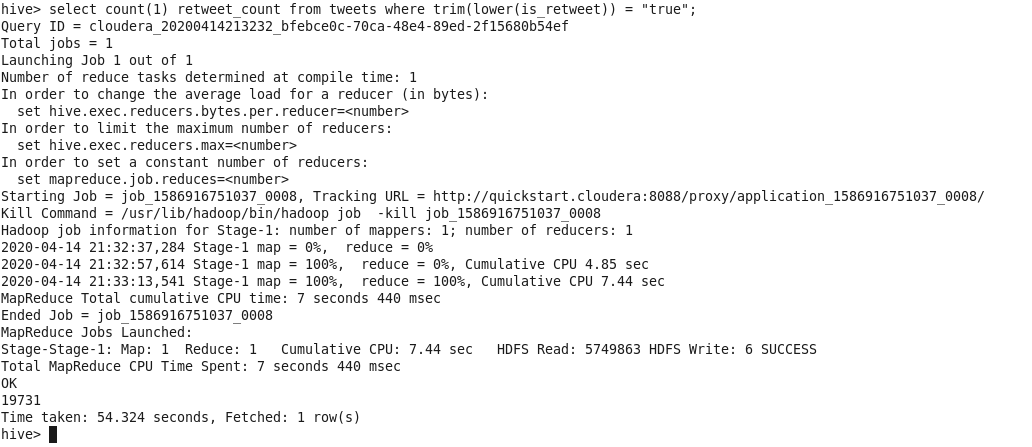




Query 2: Retweet count

SELECT COUNT(1) retweet\_count FROM tweets WHERE trim(lower(is\_retweet)) = “true”.

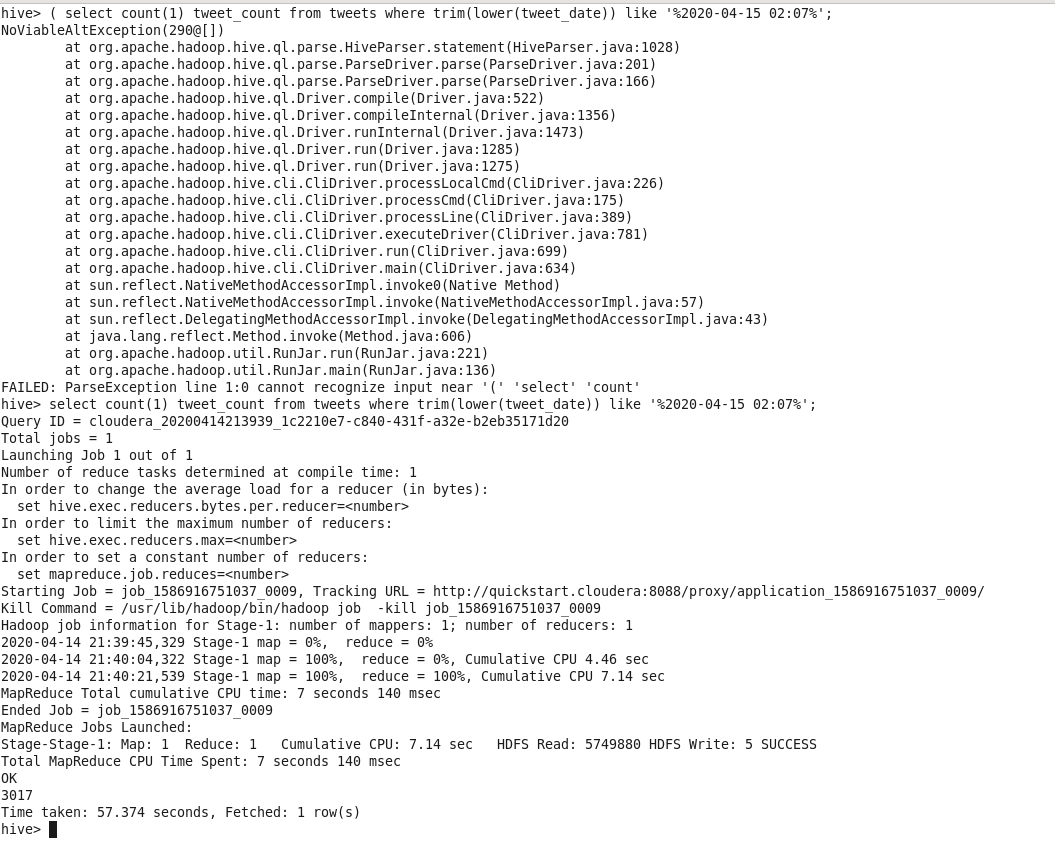
The above query is run to see the total number of tweets related to COVID that are being retweeted. We see that a total of 19731 tweets are just retweeted by users.



Query 3: Tweets per minute

SELECT COUNT(1) tweet\_count FROM tweets WHERE trim(lower(tweet\_date)) LIKE ‘%2020-04-15 02:07%’.

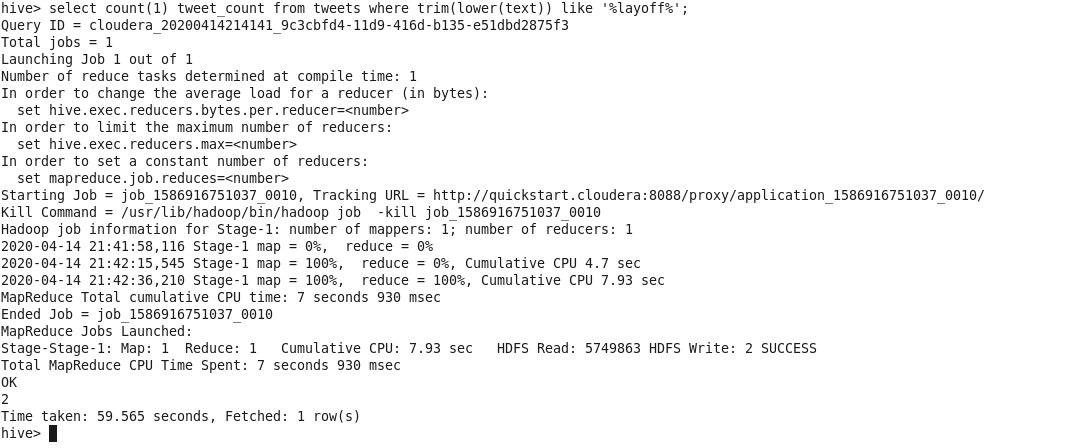
I wanted to know the number of tweets flowing in about the COVID/CORONA every minute on average. So, the above is executed on tweets and we see that on an average 3k tweets are tweeted on COVID per minute which is a big number.



Query 4: Tweets on Layoffs

SELECT COUNT(1) tweet\_count FROM tweets WHERE trim(lower(text)) LIKE ‘%layoff%’.

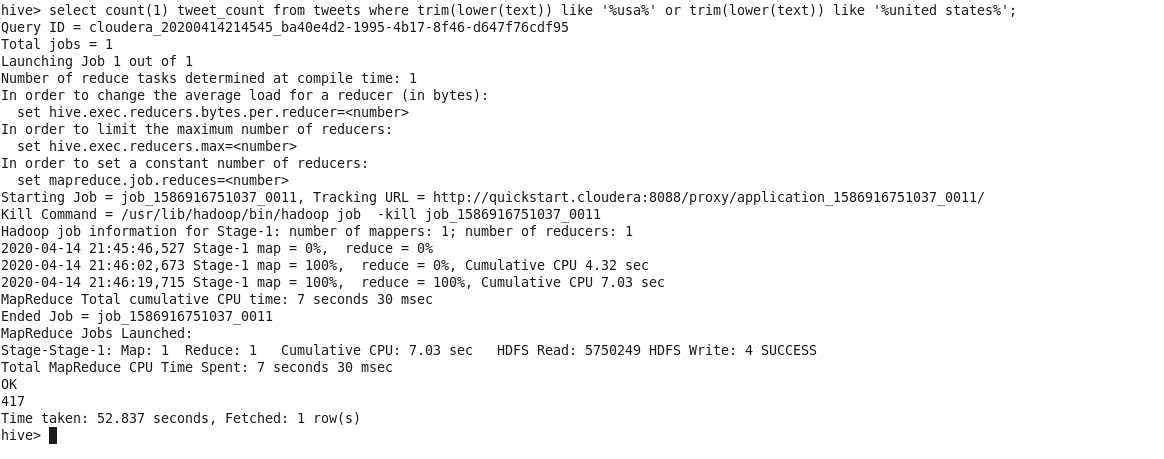
This is an important analysis. We all know that due to economic crisis employees are being laid off from the jobs. So, I wanted to know the number of people talking about the layoffs. So, we filtered the tweet text with the keyword- layoff and counted the total tweets in this context. We see just 2 tweets on layoffs which is quite shocking.



Query 5: Tweets on USA

SELECT COUNT(1) tweet\_count FROM tweets WHERE trim(lower(text)) LIKE ‘%usa%’ or trim(lower(text)) like ‘%united states%’;

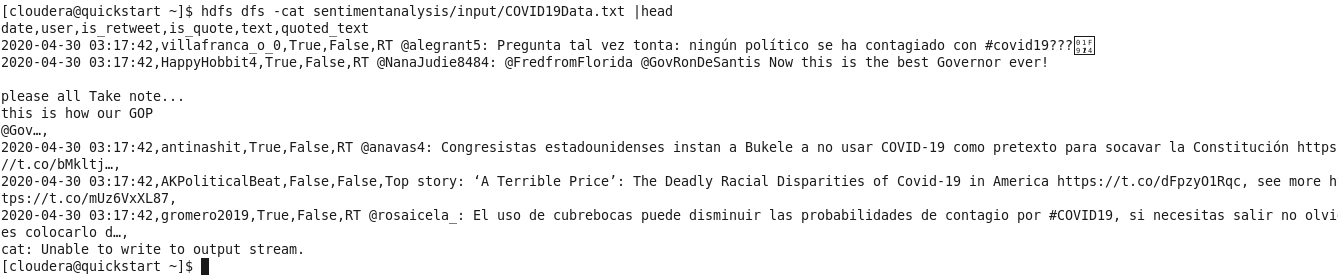
Since we are in United States and the virus is growing rapidly in United States, we wanted to analyze the count of tweets from around the world that are talking about COVID in United States. From the results, we see 417 tweets are about COVID in the United States which is not quite a large number as we thought.



**Use Case 3:** Twitter data sentimental analysis using Map Reduce

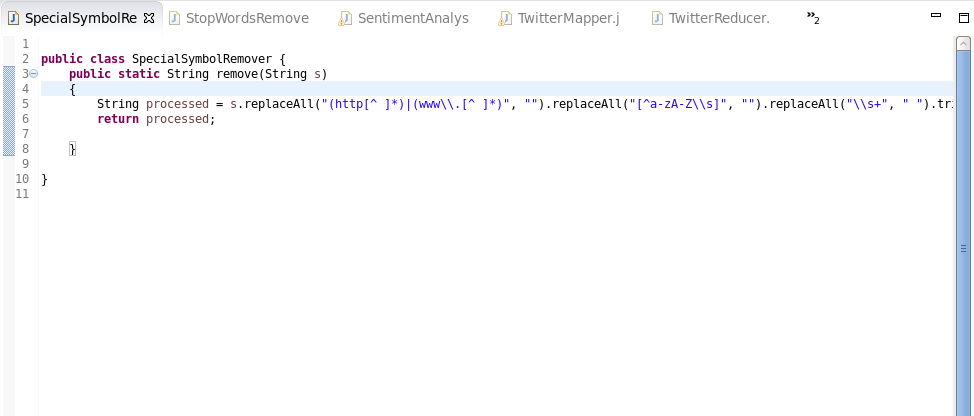
Analyzed the sensitivity of tweets. Divided tweets into 3 categories- positive, negative and neutral and displayed count for each category.

Data: Loaded input file to HDFS and displayed

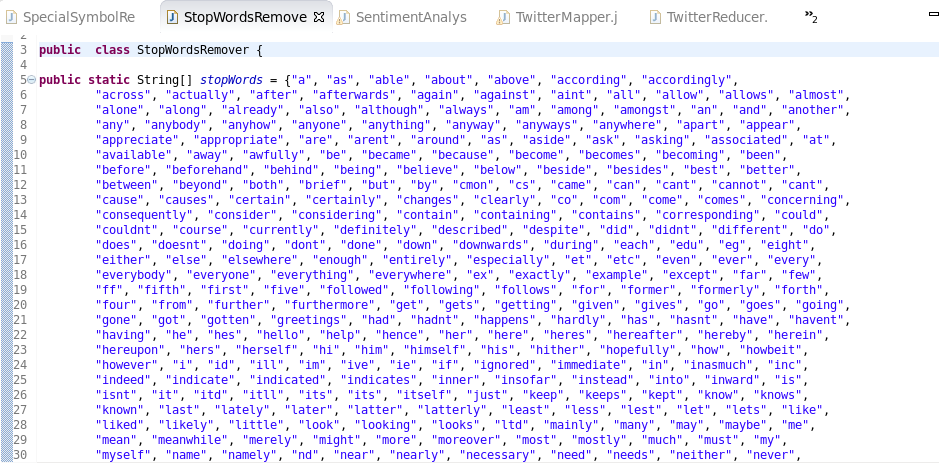


Code:

Created class- SpecialSymbolRemover for data preprocessing to remove urls in the tweets as they do not contribute to sensitivity of tweets.



Created class- StopWordsRemover for data preprocessing to remove stop words from data. Stop words like- a, to, pronouns are removed as they do not define tweet sensitivity.

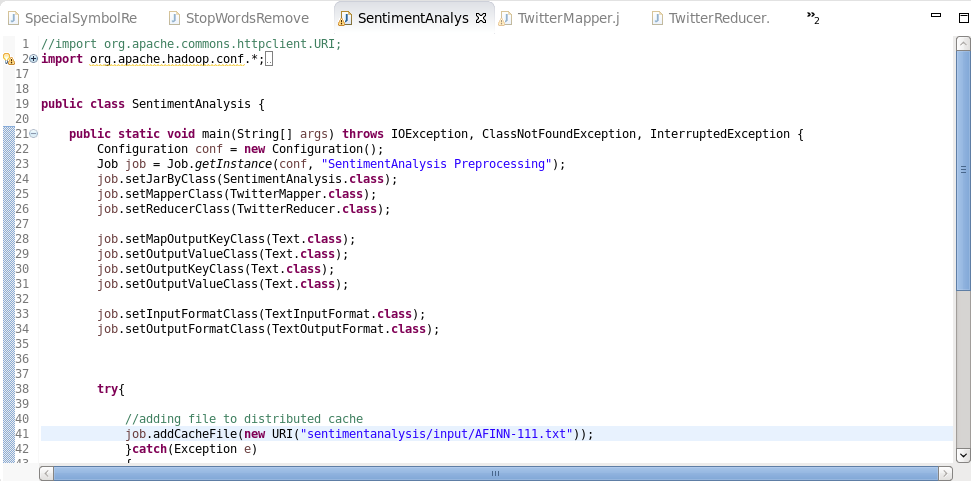






Main class- SentimentAnalysis

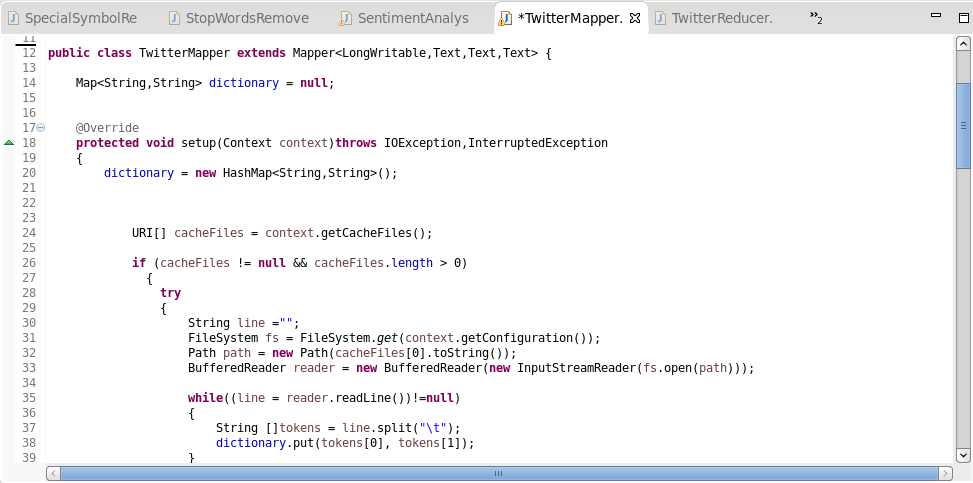
Created map, reduce jobs and added afinn-111 file to cache. Afinn-111 consists of sentiment values for various words describing emotions. Words describing emotions like anger, hatred have negative sentiment values. Words describing emotions like happiness, excitement have positive sentiment values. Words that describe neutral emotions hold sentiment value as ZERO.

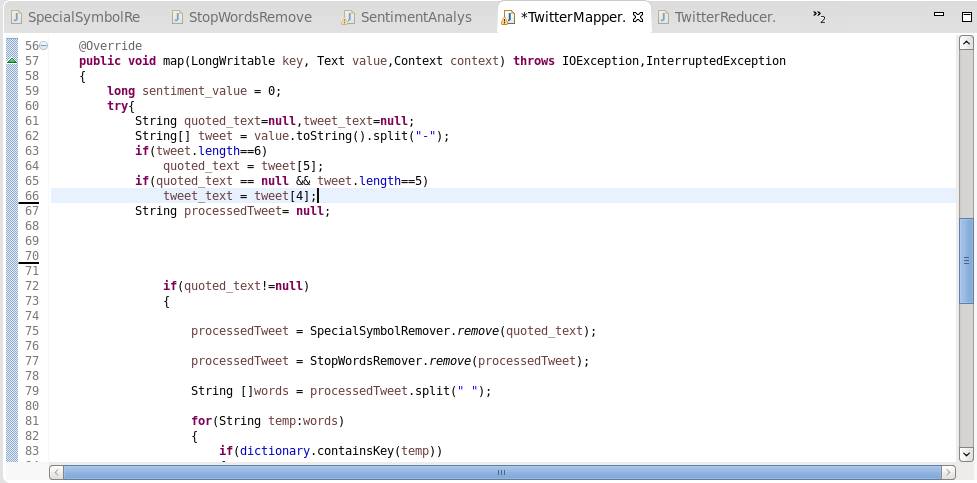




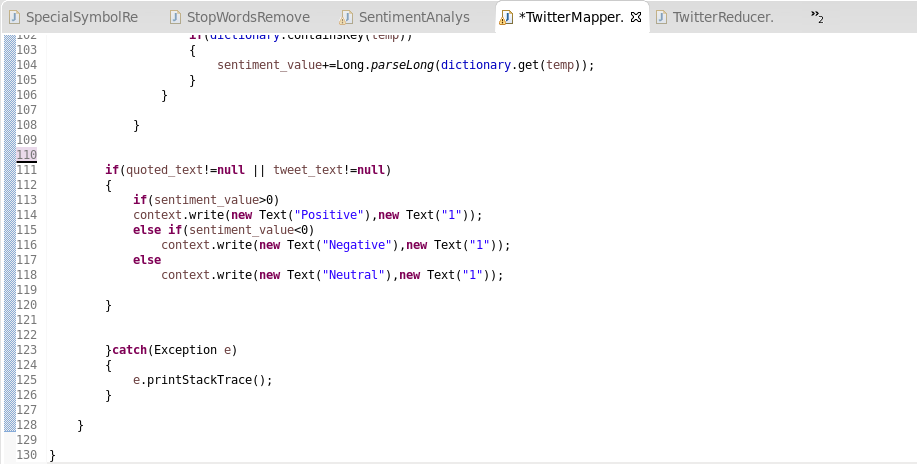
Mapper class- TwitterMapper

In the mapper class, loaded file from the cache and created map out of it. In the map function, for each tweet sentiment is calculated by summing each word sentiment. The output of map function is (sentiment\_category, 1). Example: (positive, 1)



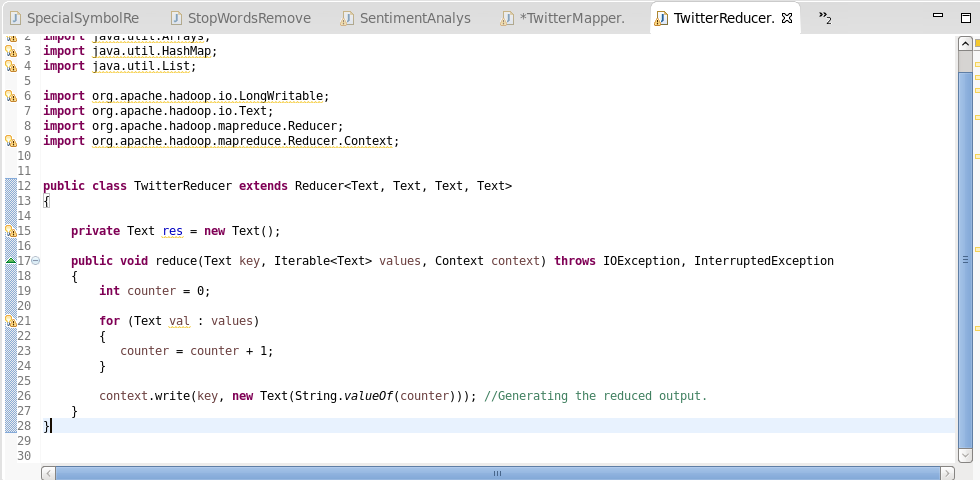


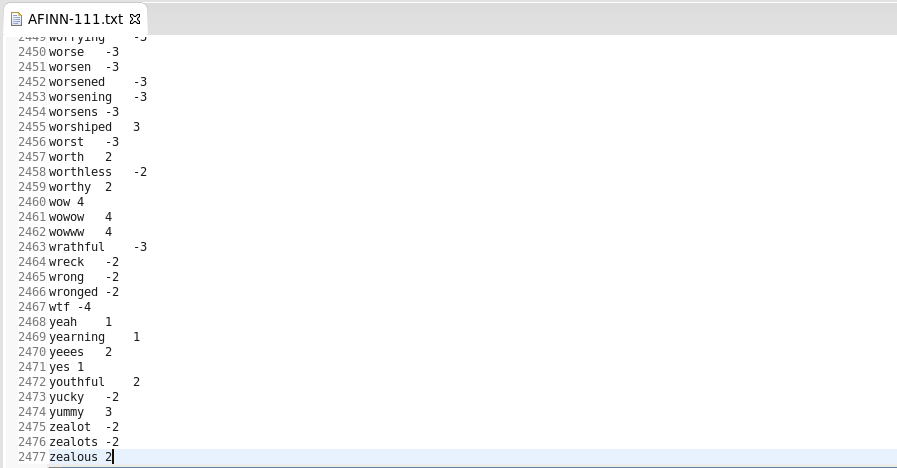




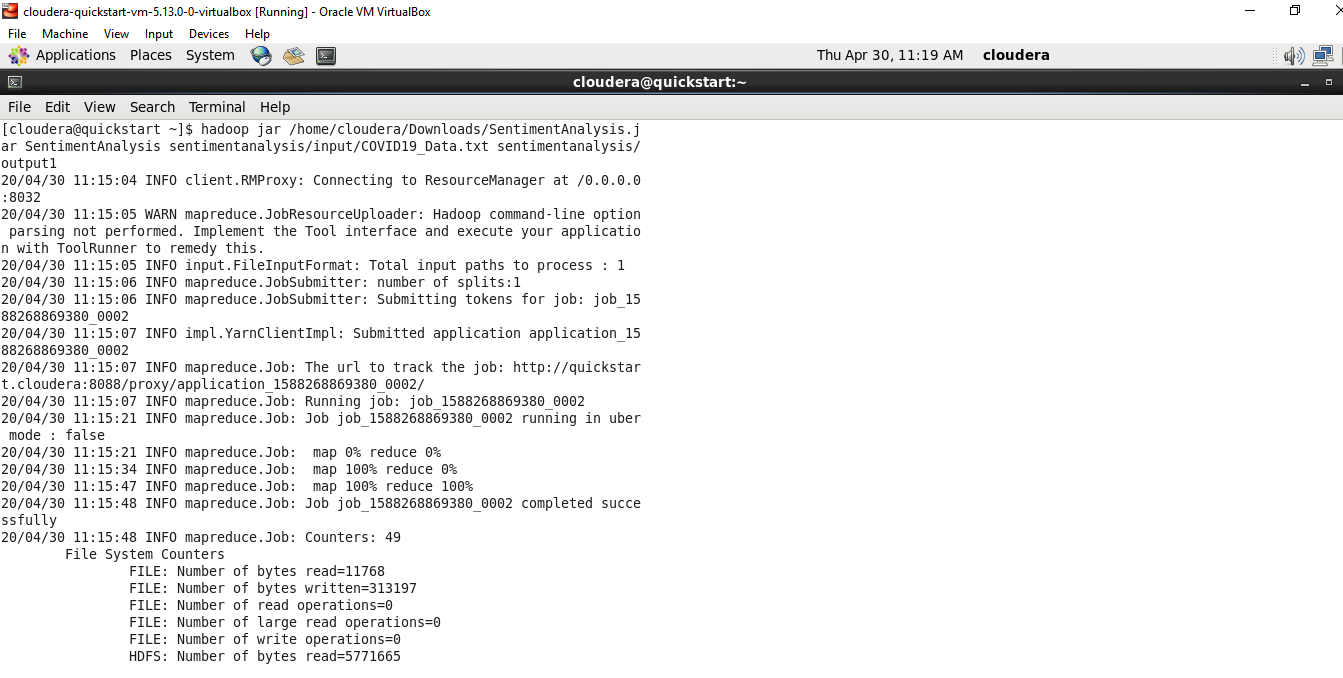
Reducer class- TwitterReducer

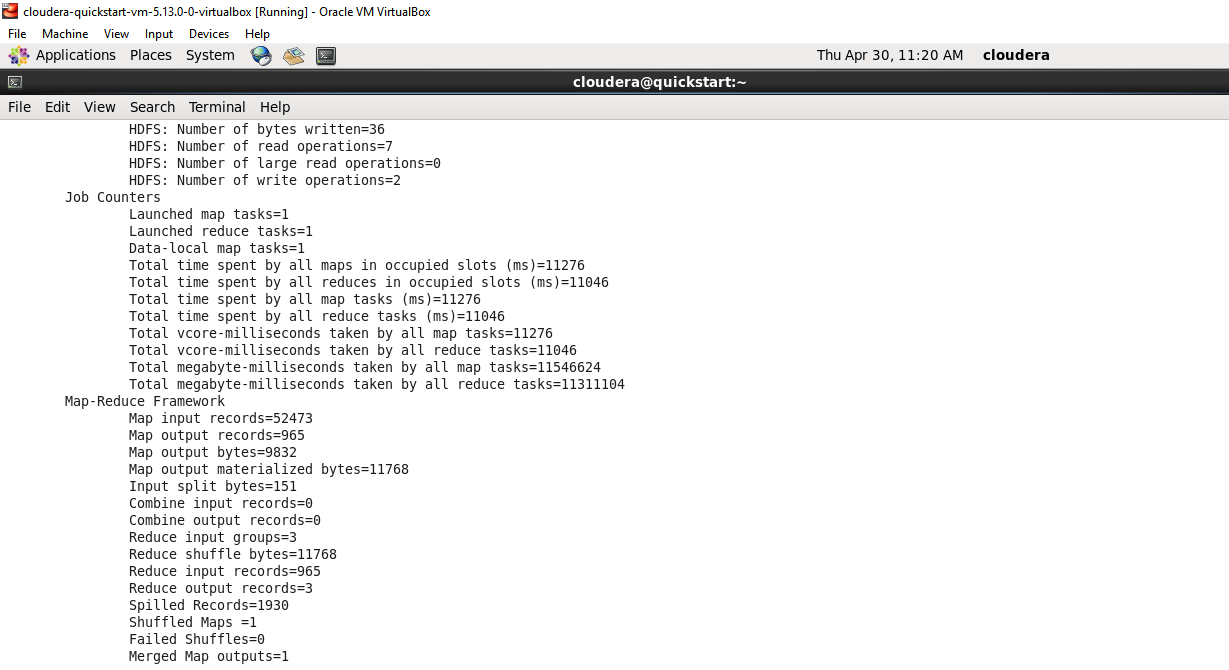
In the reducer class, each tweet category is summed for the final tweet count for each category.

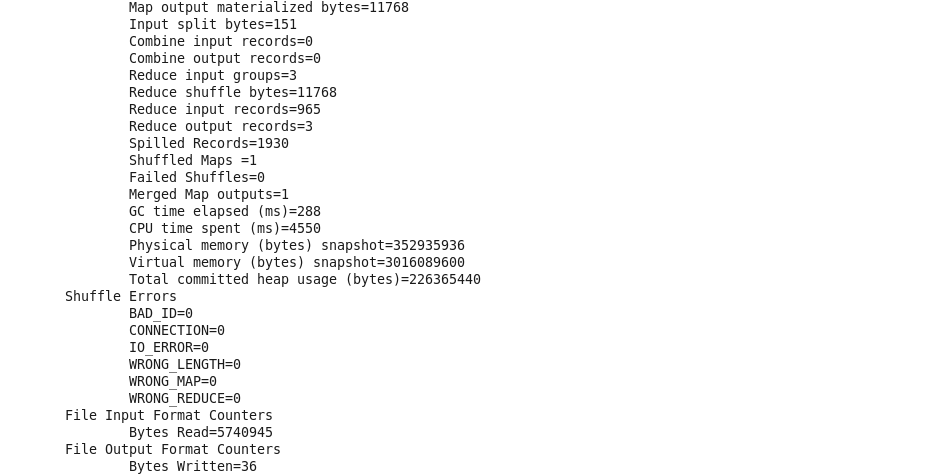


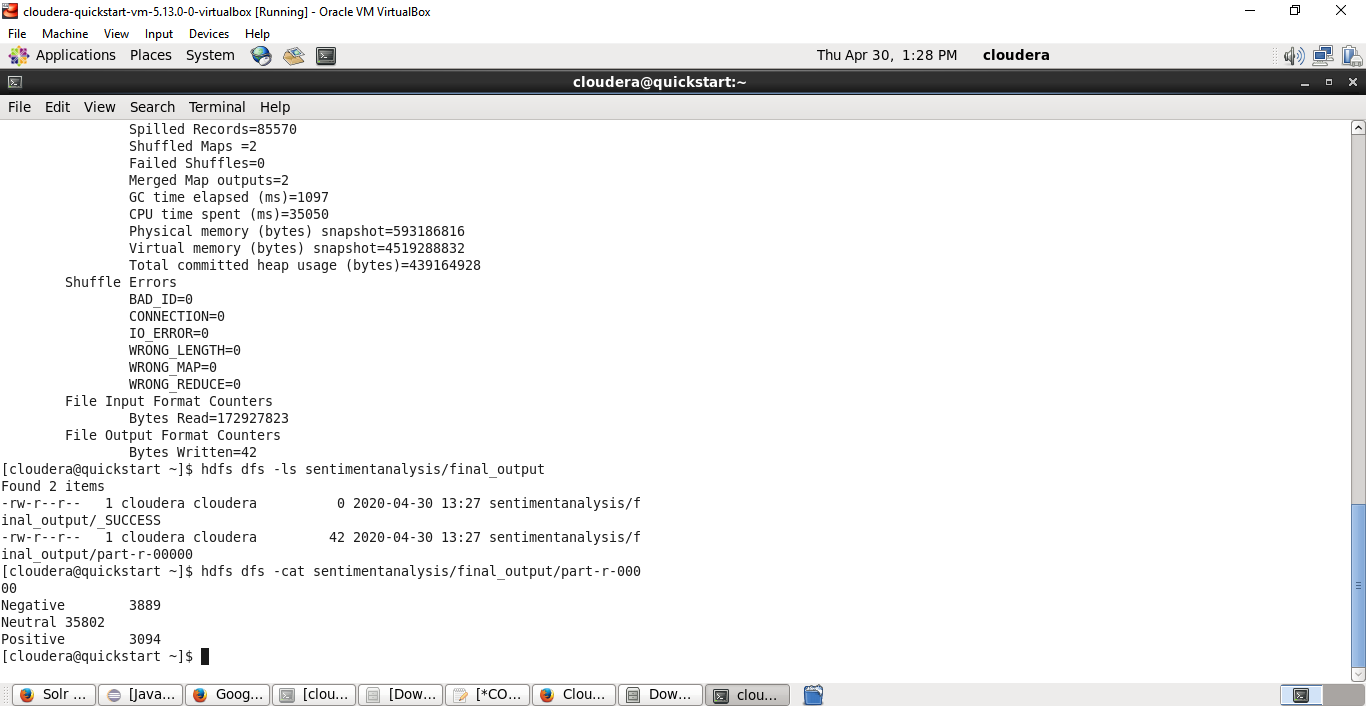


Output:









**Use Case 4:** Twitter data analysis using Cassandra

Useful analysis on tweets are performed using complex queries and user defined functions(UDFs).

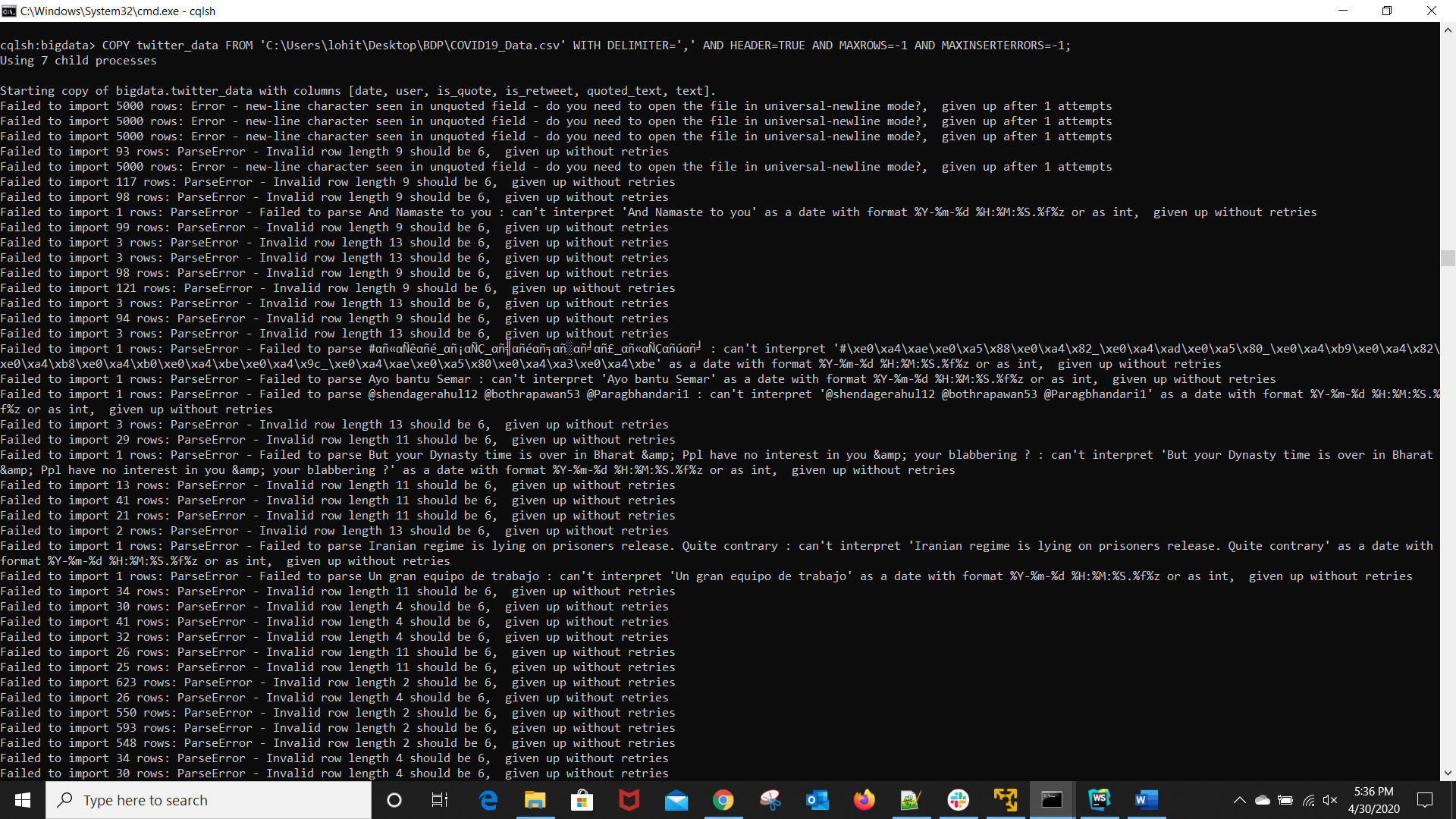
Create Table:

Twitter\_data table is created with 6 columns- date, user, is\_retweet, is\_quote, text, quoted\_text. Primary key is a composite key with columns (date,user).



Load Twitter data:

COVID19 csv file is loaded into twitter\_data table with more than 2 lakh records.



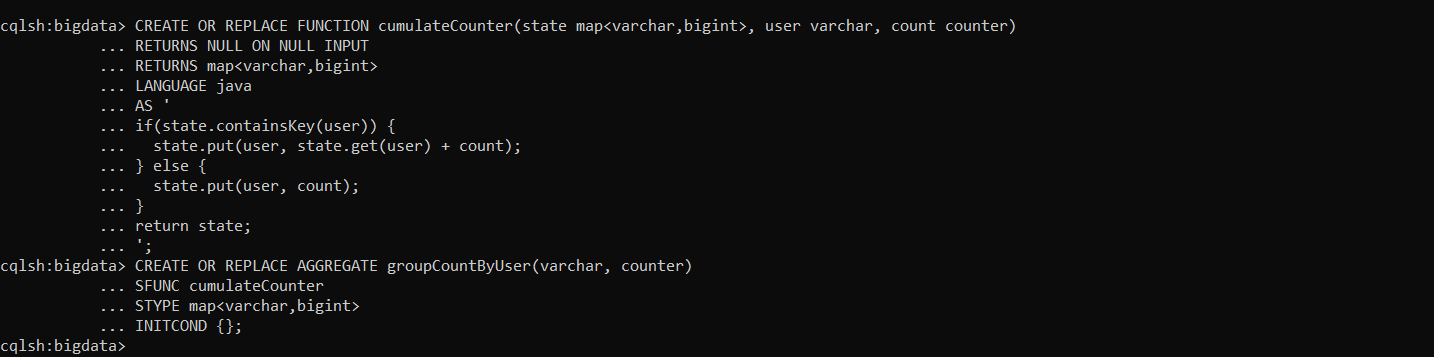


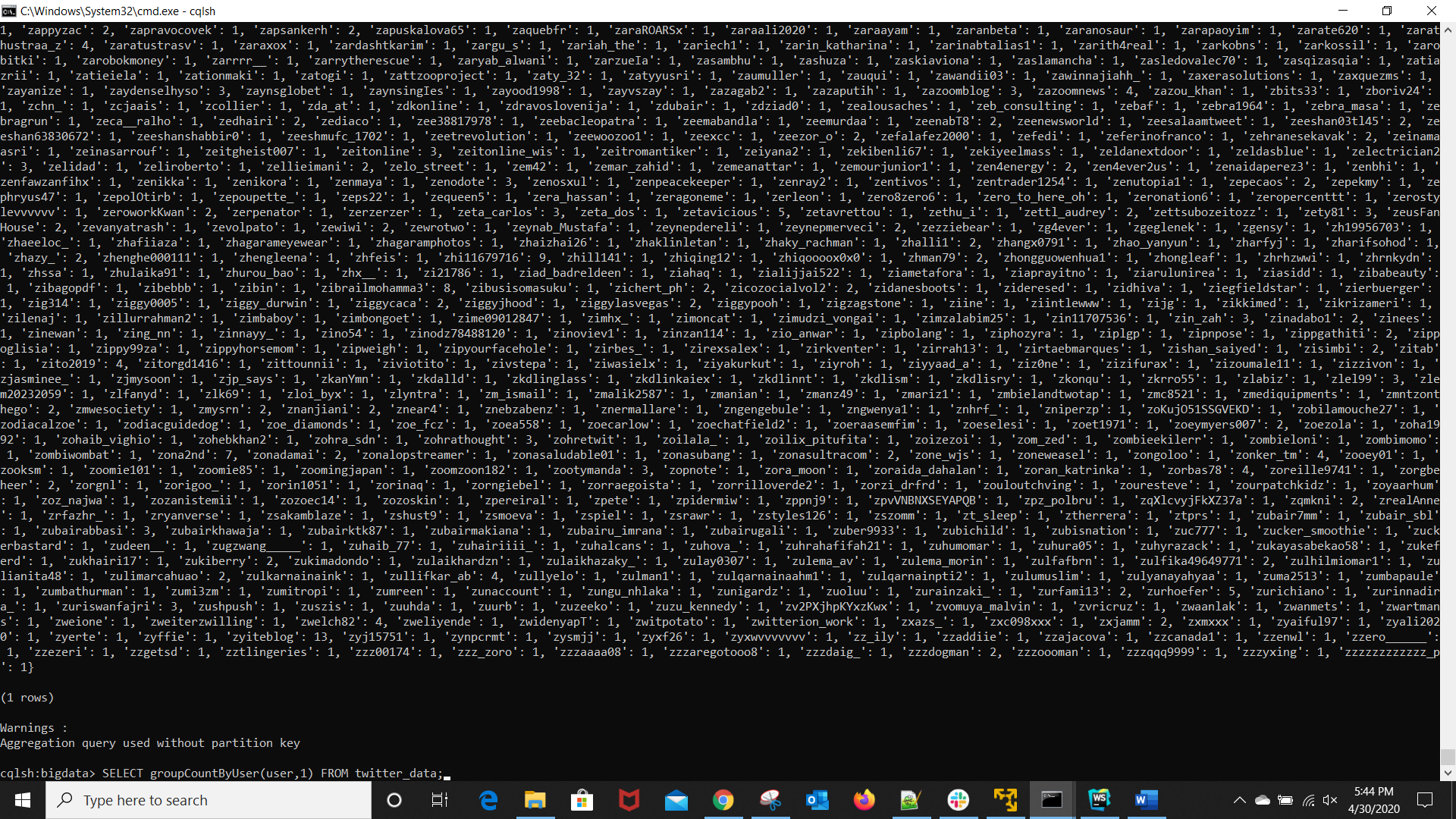
Analysis 1: Display the users and their tweet count

Cassandra does not support GROUPBY clause. To perform group by on the user column to count the tweets by each user, created UDFs- cumulateCounter and groupCountByUser. CumulateCounter maps each user to 1 if it is a new user else increments the existing count.

SELECT groupCountByUser(user,1) FROM twitter\_data;

We wanted to see the number of tweets coming from each user in general. From the results we noticed that most of the users are tweeting not more than 1 or 2 tweets in a span of day or two.



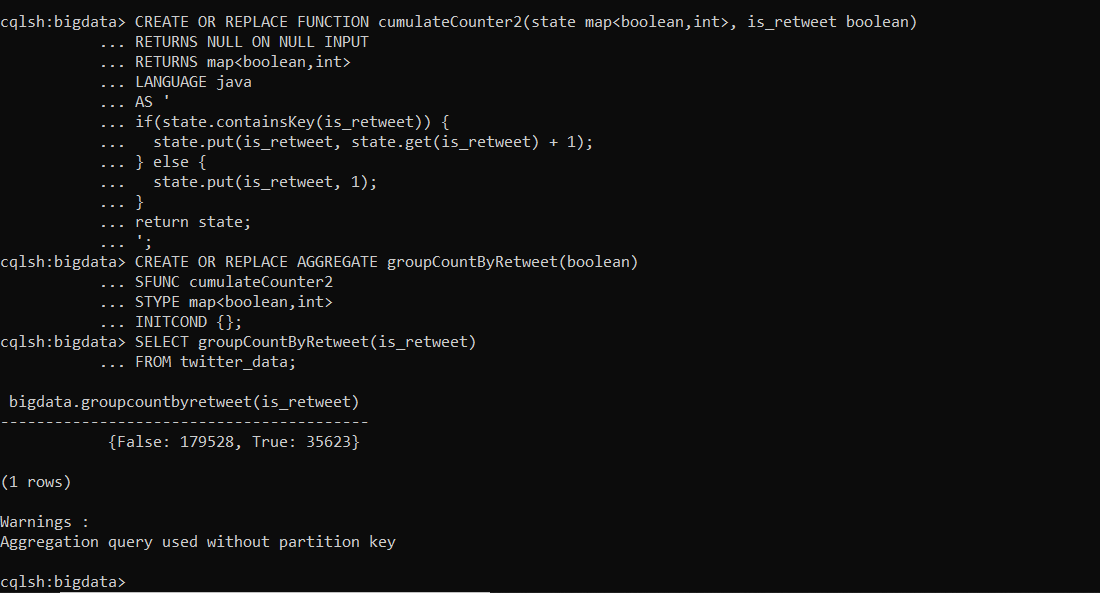


Analysis 2: Count the number of retweets

To perform GROUPBY aggregation on is\_retweet column, created UDFs- cumulateCounter2 and groupCountByRetweet. CumulateCounter2 aggregates retweets by their Boolean value- false and true.

SELECT groupCountByRetweet(is\_retweet) FROM twitter\_data;

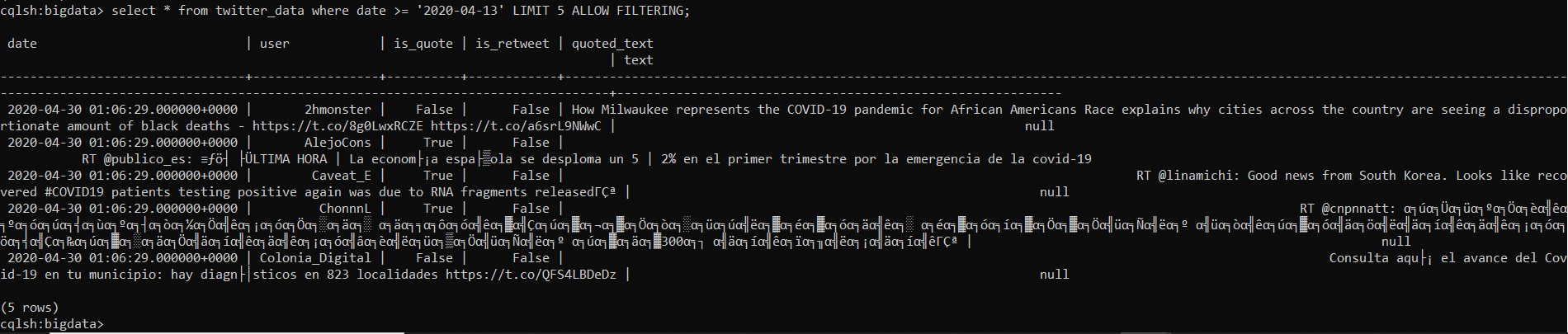
We are curious to know if the people are really interested to speak their opinions or just retweeting other tweets that are found worth sharing. From the analysis, we see that 179528 tweets are tweeted by the users and 35623 tweets are just retweets of others. We see that around 15% of the total tweets are retweeted and the rest 85% of the tweets are not retweeted.



Analysis 3: Display tweets that are tweeted after date- '2020-04-13'

SELECT \* FROM twitter\_data WHERE date >= ‘2020-04-13’ LIMIT 5 ALLOW FILTERING;

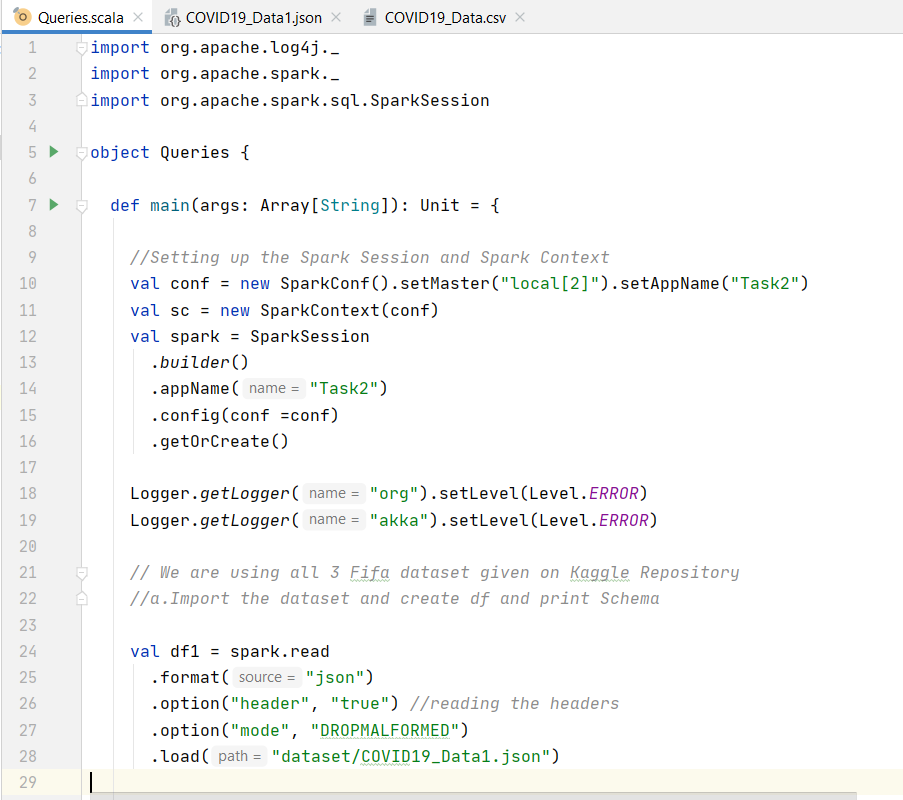
To see some actual tweet text we are displaying few tweets posted after April 13(the day we performed analysis).



**Use Case 5:** Twitter data analysis using Spark SQL

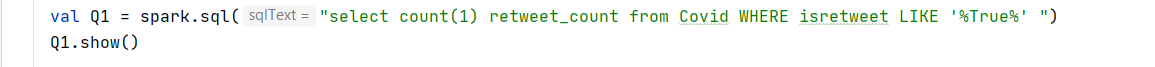
We have already performed the below analysis using Hive and Cassandra but we wanted to see how well spark is computing the results over others. We noticed that Spark is able to run the queries faster that the Cassandra and Hive because of in-memory computation.

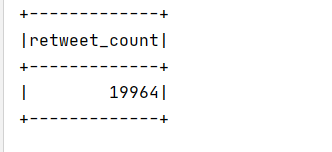
Load json data into SparkContext



Analysis 1: Display the count of the tweets that are retweeted

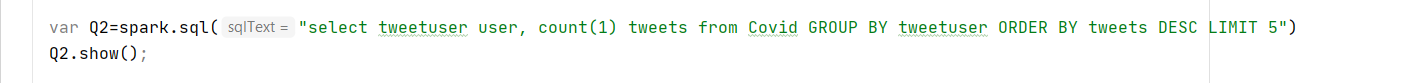
SELECT COUNT(1) retweet\_count FROM COVID WHERE isretweet LIKE ‘%True%’;

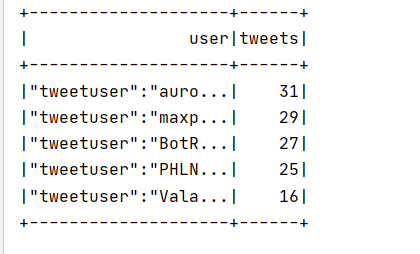




Analysis 2: Display the top 5 users with highest number of tweets along with count

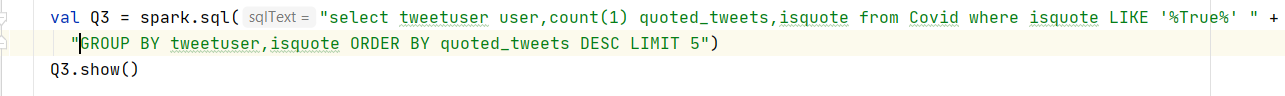
SELECT tweetuser user, COUNT(1) FROM Covid GROUP BY tweetuser ORDER BY tweets DESC LIMIT 5;

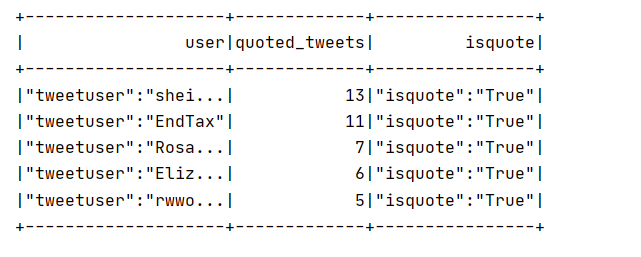




Analysis 3: Display the top 5 users with highest number of quoted tweets along with count

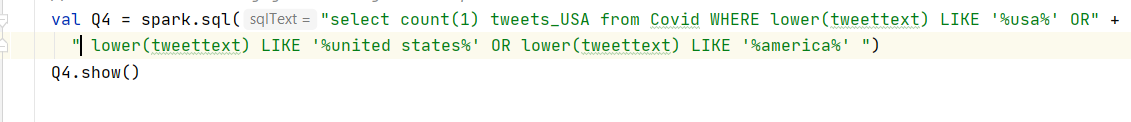
SELECT tweetuser user, COUNT(1) quoted\_tweets, isquote FROM Covid WHERE isquote LIKE ‘%True%’ GROUP BY tweetuser, isquote ORDER BY quoted\_tweets DESC LIMIT 5;

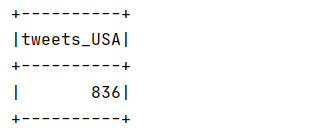




Analysis 4: Display the number of tweets taking about ‘United States’

SELECT COUNT(1) tweets\_USA FROM covid WHERE lower(tweettext) LIKE ‘%usa%’ OR lower(tweettext) LIKE ‘%united states%’ OR lower(tweettext) LIKE ‘%america%’;





Work Completed (100%):

* Collected Data
* Analyzed data using map reduce and hive
* Analyzed data using Cassandra
* Performed sentiment analysis using map reduce
* Analyzed data using Spark SQL

Responsibility:

Vidyullatha Lakshmi Kaza- 34%

Aparna Manda- 33%

Lohitha Yenugu- 33%

References:

1. <https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json>
2. <https://cassandra.apache.org/doc/latest/cql/dml.html>
3. <https://stackoverflow.com/questions/17342176/max-distinct-and-group-by-in-cassandra>
4. <https://docs.datastax.com/en/cql-oss/3.3/cql/cql_reference/cqlCreateAggregate.html>