Twitter Data Analysis

DSCI 5350

Suhail Bari - sb1164

Question 1 - Correctly process and store the files in Hive. All tables created for the solution must have your student_id as a prefix to table name. For example, if I were to store the dictionary table, I would name it dictionary_ks0776 (5 points)

To create and import data from the Json file in hive, we can use multiple methods. I opted to use the Serde. In this way, we use the Apache JSON Hive Serde. In order to do this, we have to add the jar file to the resources on Hive. (shown below)

```
sb1164
File  Edit  View  Search  Terminal  Help
[training@localhost ~]$ add jar Desktop/hive-serdes-1.0-SNAPSHOT.jar
bash: add: command not found
[training@localhost ~]$ hive | add jar Desktop/hive-serdes-1.0-SNAPSHOT.jar
bash: add: command not found
^C
[training@localhost ~]$ clear;
[training@localhost ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.p
roperties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> add jar Desktop/hive-serdes-1.0-SNAPSHOT.jar
    > ;
Added [Desktop/hive-serdes-1.0-SNAPSHOT.jar] to class path
Added resources: [Desktop/hive-serdes-1.0-SNAPSHOT.jar]
hive> create table as
    > ;
FAILED: SemanticException [Error 10043]: Either list of columns or a custom seri
alizer should be specified
hive> create table tweets_sb1164(
    > id BIGINT,
    > text STRING,
    > created_at STRING)
    > ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe'
```

After the Serde is added, we start creating tables. The first table is the staging table where the entire JSON file is imported in one column. (shown below)

```
sb1164
File  Edit  View  Search  Terminal  Help
OK
2016-07-29 05:59:31
Time taken: 0.187 seconds, Fetched: 1 row(s)
hive> select DATE(date) from tweets_date_sb1164 limit 1;
OK
NULL
Time taken: 0.22 seconds, Fetched: 1 row(s)
hive> create table stage_sb1164(jstring STRING);
OK
Time taken: 0.299 seconds
hive> load data inpath 'rawdata/tweets/Twitter.json' overwrite into table stage_
sb1164;
Loading data to table default.stage_sb1164
chgrp: changing ownership of 'hdfs://localhost:8020/user/hive/warehouse/stage_sb
1164/Twitter.json': User does not belong to hive
Table default.stage_sb1164 stats: [numFiles=1, numRows=0, totalSize=443237, rawD
ataSize=0]
OK
Time taken: 0.652 seconds
hive> select * from stage_sb1164;
OK
{"retweet_count": 7,"created_at": "Fri Jul 29 12:59:31 +0000 2016","text": "It i
s being reported by virtually everyone, and is a fact,that the media pile on aga
inst me is the worst in American political history!","id": 641766061380228000,"s
```

As per the other requirements of this project, we need to change the format of the date column. We create a new table tweets_date where we define the column format as the unix time stamp. (shown below)

```
sb1164
File  Edit  View  Search  Terminal  Help
 68003 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 680 msec
OK
Time taken: 55.284 seconds
hive> select date from tweets_date_sb1164 limit 1;
OK
NULL
Time taken: 0.183 seconds, Fetched: 1 row(s)
hive> drop table tweets_date_sb1164;
OK
Time taken: 1.219 seconds
hive> create table tweets_date_sb1164 as
    > id as tweet_id,
    > text,
    > from_unixtime(UNIX_TIMESTAMP(created_at,"EEE MMM d HH:mm:ss Z yyyy")) as d
ate
    > from tweets_sb1164;
NoViableAltException(26@[])
        at org.apache.hadoop.hive.ql.parse.HiveParser_SelectClauseParser.selectC
lause(HiveParser_SelectClauseParser.java:759)
        at org.apache.hadoop.hive.ql.parse.HiveParser.selectClause(HiveParser.ja
va:44508)
        at org.apache.hadoop.hive.ql.parse.HiveParser.singleSelectStatement(Hive
Parser.java:41634)
```

In the next step, we create a new table users where use the CAST function to turn the user id from a string to a BIGINT data type. The get_json_object function is used to extract data out of a json file since json files have complex datatypes. (shown below)

```
sb1164
File  Edit  View  Search  Terminal  Help
    > select
    > Cast(get_json_object("$.id") AS BIGINT) AS tweet_id,
    > get_json_object("$.user.name") AS user_name
    > from stage_sb1164;
FAILED: SemanticException [Error 10014]: Line 3:5 Wrong arguments '"$.id"': No m
atching method for class org.apache.hadoop.hive.ql.udf.UDFJson with (string). Po
ssible choices: _FUNC_(string, string)
hive> create table users_sb1164 as
    > select
    > Cast(get_json_object(jstring,"$.id") AS BIGINT) AS tweet_id,
    > get_json_object(jstring,"$.user.name") AS user_name
    > from stage_sb1164;
Query ID = training_20200225160606_f954ff94-951c-409d-b9b4-b8333bc06c2a
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1582652563506_0007, Tracking URL = http://localhost:8088/prox
y/application_1582652563506_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1582652563506_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-02-25 16:06:32,329 Stage-1 map = 0%,  reduce = 0%
2020-02-25 16:06:54,104 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.01 se
c
MapReduce Total cumulative CPU time: 5 seconds 10 msec
```

In the next step, we create a tweets_token table in which we use the regexp_replace function. This function is used to find and replace a particular value in a string.

```
sb1164
File  Edit  View  Search  Terminal  Help
ts_token_sb1164.date, w.word)
hive> drop table tweets_token_sb1164;
OK
Time taken: 0.265 seconds
hive> create table tweets_token_sb1164 as
    > select tweet_id,
    > regexp_replace(
    > regexp_replace(
    > regexp_replace(text,"#"," #"),
    > "@", " @"),
    > "[\!\?\.,:-]", " ") as text,
"[!\?\.,:-]", " ") as text,
    > date
    > from tweets_date_sb1164;
Query ID = training_20200225163737_7e6fd9e2-5371-4599-b3de-438af1812d89
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1582652563506_0011, Tracking URL = http://localhost:8088/prox
y/application_1582652563506_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1582652563506_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-02-25 16:37:46,400 Stage-1 map = 0%,  reduce = 0%
2020-02-25 16:38:06,677 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.31 se
```

We create the words table where we split each word and by used LATERAL VIEW explode, we convert the rows to columns.

```
sb1164
File  Edit  View  Search  Terminal  Help
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 4.31 sec   HDFS Read: 80591 HDFS Write:
77813 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 310 msec
OK
Time taken: 43.675 seconds
hive> create table words_sb1164 as
    > select tweet_id,
    > regexp_replace(word," ","") as word
    > FROM tweets_token_sb1164 LATERAL VIEW explode(split(text, " ")) w as word
    > WHERE word <> "";
Query ID = training_20200225163939_211f328d-bb8f-4fb0-871a-83483785a267
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1582652563506_0012, Tracking URL = http://localhost:8088/prox
y/application_1582652563506_0012/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1582652563506_0012
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2020-02-25 16:39:27,469 Stage-1 map = 0%,  reduce = 0%
2020-02-25 16:39:49,122 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.53 se
c
MapReduce Total cumulative CPU time: 6 seconds 530 msec
Ended Job = job_1582652563506_0012
```

The next step is to include the outlier hashtags and give them a defined value. We use the join to get data from two tables – tweet_tokens and words to create the hashtag table (shown below)

```
sb1164
File  Edit  View  Search  Terminal  Help
OK
date                    string
_c1                     string
Time taken: 0.258 seconds, Fetched: 2 row(s)
hive> drop table hashtag_sb1164;
OK
Time taken: 0.468 seconds
hive> create table hashtag_sb1164 as
    > select
    > t.date,
    > regexp_replace(
    > regexp_replace(wo.word, "#MAGA[a-zA-Z0-9]{1,}","#MAGA"),
    > "#ImWithYou[a-zA-Z0-9]{1,}","#ImWithYou")
    > as word
    > from tweets_token_sb1164 t
    > join words_sb1164 wo
    > on t.tweet_id = wo.tweet_id
    > where wo.word Like "#%";
Query ID = training_20200225182525_b5c6eaae-1d53-45eb-b76c-2314c51e4d57
Total jobs = 1
Execution log at: /tmp/training/training_20200225182525_b5c6eaae-1d53-45eb-b76c-
2314c51e4d57.log
2020-02-25 06:26:05     Starting to launch local task to process map join;     m
aximum memory = 1013645312
```

Question 2.a)

What were the hashtags used in the file, and how many times each hashtag was used?

Total hashtags – 234

```
sb1164
File  Edit  View  Search  Terminal  Help
2016-07-11 04:57:45     #TrumpTrain
2011-09-10 15:23:38     #MakeAmericaGreatAgain
2016-07-11 04:57:45     #CrookedHillary
2016-07-11 04:57:45     #ThrowbackThursday
Time taken: 0.179 seconds, Fetched: 20 row(s)
hive> select * from hashtag_sb1164;
OK
2016-08-23 06:53:11     #tcdisrupt
2016-08-23 06:53:11     #tcdisrupt
2016-01-25 20:25:18     #Trump2016
2016-07-29 05:59:31     #TeamTrump
2016-07-29 05:59:31     #MAGA
2016-07-29 05:59:31     #TrumpPence16
2011-09-10 15:23:38     #NotoTrump
2016-07-29 05:59:31     #MAGA
2016-07-29 05:59:31     #TrumpPence16
2011-09-10 15:23:38     #TrumpPence16
2011-09-10 15:23:38     #MAGA
2011-09-10 15:23:38     #AlwaysTrump
2011-09-10 15:23:38     #StandWithLouisiana
2011-09-10 15:23:38     #WheresHillary
2016-07-11 04:57:45     #TrumpPence16
2016-07-11 04:57:45     #ImWithYou
2016-07-11 04:57:45     #TrumpTrain
```

```
sb1164
File  Edit  View  Search  Terminal  Help
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1582652563506_0024, Tracking URL = http://localhost:8088/pro
y/application_1582652563506_0024/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1582652563506_0024
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-02-25 19:10:11,092 Stage-1 map = 0%,  reduce = 0%
2020-02-25 19:10:28,150 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.55 se
c
2020-02-25 19:10:46,510 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.1 s
ec
MapReduce Total cumulative CPU time: 6 seconds 100 msec
Ended Job = job_1582652563506_0024
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.1 sec   HDFS Read: 14148 HD
FS Write: 4 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 100 msec
OK
234
Time taken: 58.423 seconds, Fetched: 1 row(s)
hive>
```

Unique hashtags - 59

```
sb1164
File  Edit  View  Search  Terminal  Help
#USA    1
#VoterFraud     1
#WCS16  1
#WakeUpAmerica  1
#WheresHillary  1
#smallbiz       1
#taxplan        1
#tcdisrupt      2
Time taken: 61.499 seconds, Fetched: 59 row(s)
hive> select count(word) from hashtag_sb1164;
Query ID = training_20200225190909_2dbc0e2b-efb1-4459-accb-75feae689d7d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1582652563506_0024, Tracking URL = http://localhost:8088/prox
y/application_1582652563506_0024/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1582652563506_0024
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
```

```
sb1164
File  Edit  View  Search  Terminal  Help
#tcdisrupt      2
#ThrowbackThursday      2
#Obamacare      2
#ICYMI  2
#TrumpTrain     2
#MakeAmericaWorkAgain   2
#LawandOrder    2
#LESM   3
#Hillary        3
#Trump  4
#2A     4
#GOPConvention  5
#MakeAmericaSafeAgain   5
#NeverHillary   5
#RNCinCLE       10
#Trump2016      15
#CrookedHillary 17
#AmericaFirst   18
#TrumpPence16   19
#MAGA   23
#ImWithYou      26
#MakeAmericaGreatAgain  26
Time taken: 123.49 seconds, Fetched: 59 row(s)
hive>
```

Question 2.b)

Identify the most trending hashtag by the day.

```
sb1164
File  Edit  View  Search  Terminal  Help
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:606)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
FAILED: ParseException line 1:57 cannot recognize input near 'groub' 'by' 'date
 in table source
hive> select date, word, count(word) from hashtag_sb1164 group by date;
FAILED: SemanticException [Error 10025]: Line 1:13 Expression not in GROUP BY ke
y 'word'
hive> select date, word, count(word) from hashtag_sb1164 group by word;
FAILED: SemanticException Line 0:-1 Expression not in GROUP BY key 'date'
hive> select date, word, count(word) from hashtag_sb1164 order by word;
FAILED: SemanticException Line 0:-1 Expression not in GROUP BY key 'date'
hive> select date, word, count(*) from hashtag_sb1164 group by date,word;
Query ID = training_20200225185858_0e6730d8-074c-41be-b982-c5d90802368d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
```

```
sb1164
File  Edit  View  Search  Terminal  Help
2016-07-08 16:26:56     #DefendtheSecond        1
2016-07-08 16:26:56     #ElectionDay    1
2016-07-08 16:26:56     #MakeAmericaGreatAgain  1
2016-07-08 16:26:56     #NeverHillary   2
2016-07-08 16:26:56     #SCOTUS 1
2016-07-11 04:57:45     #CrookedHillary 1
2016-07-11 04:57:45     #Hillary        1
2016-07-11 04:57:45     #ImWithYou      1
2016-07-11 04:57:45     #MAGA   1
2016-07-11 04:57:45     #MakeAmericaSafeAgain   1
2016-07-11 04:57:45     #ThrowbackThursday      1
2016-07-11 04:57:45     #Trump  1
2016-07-11 04:57:45     #TrumpPence16   1
2016-07-11 04:57:45     #TrumpTrain     1
2016-07-24 13:45:31     #1      1
2016-07-24 13:45:31     #CrookedHillary 2
2016-07-24 13:45:31     #ICYMI  1
2016-07-29 05:59:31     #MAGA   2
2016-07-29 05:59:31     #TeamTrump      1
2016-07-29 05:59:31     #TrumpPence16   2
2016-08-23 06:53:11     #tcdisrupt      2
Time taken: 34.613 seconds, Fetched: 109 row(s)
hive> Display all 465 possibilities? (y or n)
!                       !=                      $ELEM$
```

How many times the most trending hashtag was tweeted?

```
sb1164
File  Edit  View  Search  Terminal  Help
hive> select quantity  from tex_sb1164 where quantity = max(quantity);
FAILED: SemanticException [Error 10128]: Line 1:50 Not yet supported place for U
DAF 'max'
hive> SELECT date,word,quantity FROM ( SELECT date,word,quantity,row_number() ov
er (partition by date order by count desc) as rn from tex_sb1164) sq WHERE sq.rn
 = 1;
FAILED: SemanticException Failed to breakup Windowing invocations into Groups. A
t least 1 group must only depend on input columns. Also check for circular depen
dencies.
Underlying error: org.apache.hadoop.hive.ql.parse.SemanticException: Line 1:105
Invalid table alias or column reference 'count': (possible column names are: dat
e, word, quantity)
hive> SELECT date,word,quantity FROM ( SELECT date,word,quantity,row_number() ov
er (partition by date order by quantity desc) as rn from tex_sb1164) sq WHERE sq
.rn = 1;
Query ID = training_20200226174444_4d5ceeea-3eaf-40a5-917b-be5127232acc
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
```

```
sb1164
File  Edit  View  Search  Terminal  Help
OK
2009-03-18 06:46:38     #CrookedHillary 2
2011-09-10 15:23:38     #MakeAmericaGreatAgain  20
2015-12-15 06:47:56     #RiggedSystem   1
2015-12-18 15:14:08     #CrookedHillary 2
2015-12-22 14:13:13     #ImWithYou      3
2015-12-24 15:28:54     #Trump2016      1
2016-01-19 08:19:49     #RNCinCLE       2
2016-01-25 20:25:18     #Hillary        1
2016-01-29 17:41:51     #ImWithYou      2
2016-02-20 07:20:27     #MakeAmericaGreatAgain  1
2016-03-20 04:55:27     #CrookedHillary 2
2016-04-04 14:07:04     #DNC    1
2016-04-26 18:02:47     #Trumpforlife   1
2016-05-21 06:34:02     #MAGA   2
2016-05-24 22:42:59     #Trump2016      3
2016-06-07 17:41:42     #MakeAmericaGreatAgain  2
2016-07-08 16:26:56     #2A     2
2016-07-11 04:57:45     #MAGA   1
2016-07-24 13:45:31     #CrookedHillary 2
2016-07-29 05:59:31     #MAGA   2
2016-08-23 06:53:11     #tcdisrupt      2
Time taken: 34.966 seconds, Fetched: 21 row(s)
hive>
```

Question 2.c)

Determine the score for each tweet that was posted? Identify whether the tweet had a positive or negative sentiment? Use the dictionary.txt file for determining the score. Note: Include the date ('yyyy-mm-dd'), tweet_id, user_name, and the score in the resulting query.

In order to do sentiment analysis, we have to pull the data from dictionary.txt into table dt.
We create a new table words_join using a left outer join on words table and dictionary table.

```
sb1164
File  Edit  View  Search  Terminal  Help
Stage-Stage-4: Map: 1   Cumulative CPU: 6.54 sec   HDFS Read: 234632 HDFS Write:
 256458 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 540 msec
OK
Time taken: 63.408 seconds
hive> create table word_joinsb1164 as select words_sb1164.tweet_id,words_sb1164.
word,dt.t_id as score from words_sb1164 LEFT OUTER JOIN dt ON(words_sb1164.word
=dt.word);
Query ID = training_20200227143737_64e4e262-aaa9-42a9-a73f-58653bae9ab8
Total jobs = 1
Execution log at: /tmp/training/training_20200227143737_64e4e262-aaa9-42a9-a73f-
58653bae9ab8.log
2020-02-27 02:37:16     Starting to launch local task to process map join;     m
aximum memory = 1013645312
2020-02-27 02:37:19     Dump the side-table for tag: 1 with group count: 2477 in
to file: file:/tmp/training/b992d61e-5bc0-4235-9603-fd284712355f/hive_2020-02-27
_14-37-02_790_2401545389252572653-1/-local-10003/HashTable-Stage-4/MapJoin-mapfi
le21--.hashtable
2020-02-27 02:37:19     Uploaded 1 File to: file:/tmp/training/b992d61e-5bc0-423
5-9603-fd284712355f/hive_2020-02-27_14-37-02_790_2401545389252572653-1/-local-10
003/HashTable-Stage-4/MapJoin-mapfile21--.hashtable (69200 bytes)
2020-02-27 02:37:19     End of local task; Time Taken: 3.252 sec.
Execution completed successfully
MapredLocal task succeeded
```

We have to perform a group by operation on tweet_id and average the of all the ratings having the same tweet id. This will give us an average of each tweet id on the basis of the dictionary table dt.

```
sb1164
File  Edit  View  Search  Terminal  Help
Time taken: 0.171 seconds, Fetched: 3 row(s)
hive> select tweet_id, score from word_joinsb1164 where score=AVG(score) group b
y word_joinsb1164.tweet_id order by score ASC;
FAILED: SemanticException [Error 10128]: Line 1:56 Not yet supported place for U
DAF 'AVG'
hive> select tweet_id, AVG(score) from word_joinsb1164 group by word_joinsb1164.
tweet_id order by score DESC;
FAILED: SemanticException [Error 10004]: Line 1:92 Invalid table alias or column
 reference 'score': (possible column names are: tweet_id, _c1)
hive> select tweet_id, AVG(score) from word_joinsb1164 group by word_joinsb1164.
tweet_id order by tweet_id DESC;
Query ID = training_20200227145050_a6ae04c2-5ef3-4471-9c2b-a2f728116fcb
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1582652563506_0039, Tracking URL = http://localhost:8088/prox
y/application_1582652563506_0039/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1582652563506_0039
```

```
sb1164
File  Edit  View  Search  Terminal  Help
OK
899285588251612000      2.0
899120405062922000      -3.0
898147844068158000      2.0
897897234225893000      1.0
894589045629596000      0.3333333333333333
893860208370497000      3.0
892993179592271000      -2.6666666666666665
892812529934080000      -2.0
892786063807571000      -2.0
892340354124439000      1.3333333333333333
890745718284415000      0.0
886494948390315000      NULL
883849857287617000      -2.0
883318030923533000      NULL
883029445050689000      -0.5
881919550707052000      2.0
880050504499670000      -3.0
878934128370149000      3.0
878424466087027000      NULL
876852986037848000      -2.0
876557870030974000      -3.0
876517185424031000      NULL
873021816475835000      3.0
```

Question 3)

Propose a better solution for the sentiment analysis as compared to 1(c). Cite the source. (5 points)

We used the tool Hive for a sentiment analysis of twitter data. Some of the problems faced during this were getting the right serde to work for the JSON file, slightly different flavor of SQL (HQL) which did not support some very important functions to answer the proposed questions and a few other issues like having to wait for the data to be processed by the map reduce. As the complexity of the queries and tables got higher, the processing engine got slower.

A better solution for doing this particular analysis is by using Spark. Spark is faster at handling Hadoop's MapReduce. It has a huge arsenal of data mining and NLP tools like SQL, Scala, Java, more importantly its very own Pyspark Python terminal.

References:

1. https://acadgild.com/blog/twitter-sentiment-analysis-using-spark
2. https://www.dezyre.com/apache-spark-tutorial/pyspark-tutorial