CIS5200 Term Project Tutorial

Authors: Divya Pakhale, Sanchita Gawand, Siddhi Udani, Tanvi Gawade

Instructor: Jongwook Woo

Date: 12/10/2018

Lab Tutorial

Divya Pakhale (dpakhal@calstatela.edu)

Sanchita Gawand (sgawand@calstatela.edu)

Siddhi Udani (sudani2@calstatela.edu)

Tanvi Gawade (tgawade@calstatela.edu)

12/10/2018

New York Times Data Analysis using Hive

OBJECTIVE

The New York Times (NYT) has a large reader base and plays an important role in shaping public opinion and outlook on current affairs and in setting the tone of the public discourse, especially in the U.S. The comments sections for articles in the NYT are quite active and gives insights to reader's opinion on the subject matter of the articles. Each comment can receive other reader's recommendations in the form of upvotes. This project aims at performing data analysis and sheds lights on New York Times Dataset using HIVEQL queries and presenting visualization to see the insights using Power BI, Tableau, & Excel 3-D Maps.

- To find out count of document type by type of material for the year 2017 and 2018.
- To determine reply count for the document type month wise for year 2017 and 2018.
- To find out reply count for each comment type for year 2017 and 2018.
- To uncover frequency of type of material with respect to article word count for both the years.
- To find out the degree of polarity to reveal the most positive as well negative headlines for the year 2017 based on public comments.
- To identify the most common words in the headlines for article year 2017 and 2018.
- Geo map to show recommendations by the user's location for the year 2017 and 2018.
- Also, to show a comparative analysis for the above objectives between the year 2017 and 2018.

INTRODUCTION

This Project aims at performing data analysis and providing insights on New York Times Comments (NYT) using HIVE and presenting the visualization in Tableau and Microsoft Power BI.

In this hands-on lab, you will learn how to:

- Load data from local desktop(windows) to Linux shell.
- Download and upload files to HDFS.
- Extract TXT file using Hive.
- Data cleaning using Hive.
- Create Hive tables to query the NYT dataset for analysis.
- Create Hive queries to analyze the sentiment of data
- Use Tableau, Power BI, Excel 3D Maps for visualization of the analyzed data.

PLATFORM SPECIFICATIONS

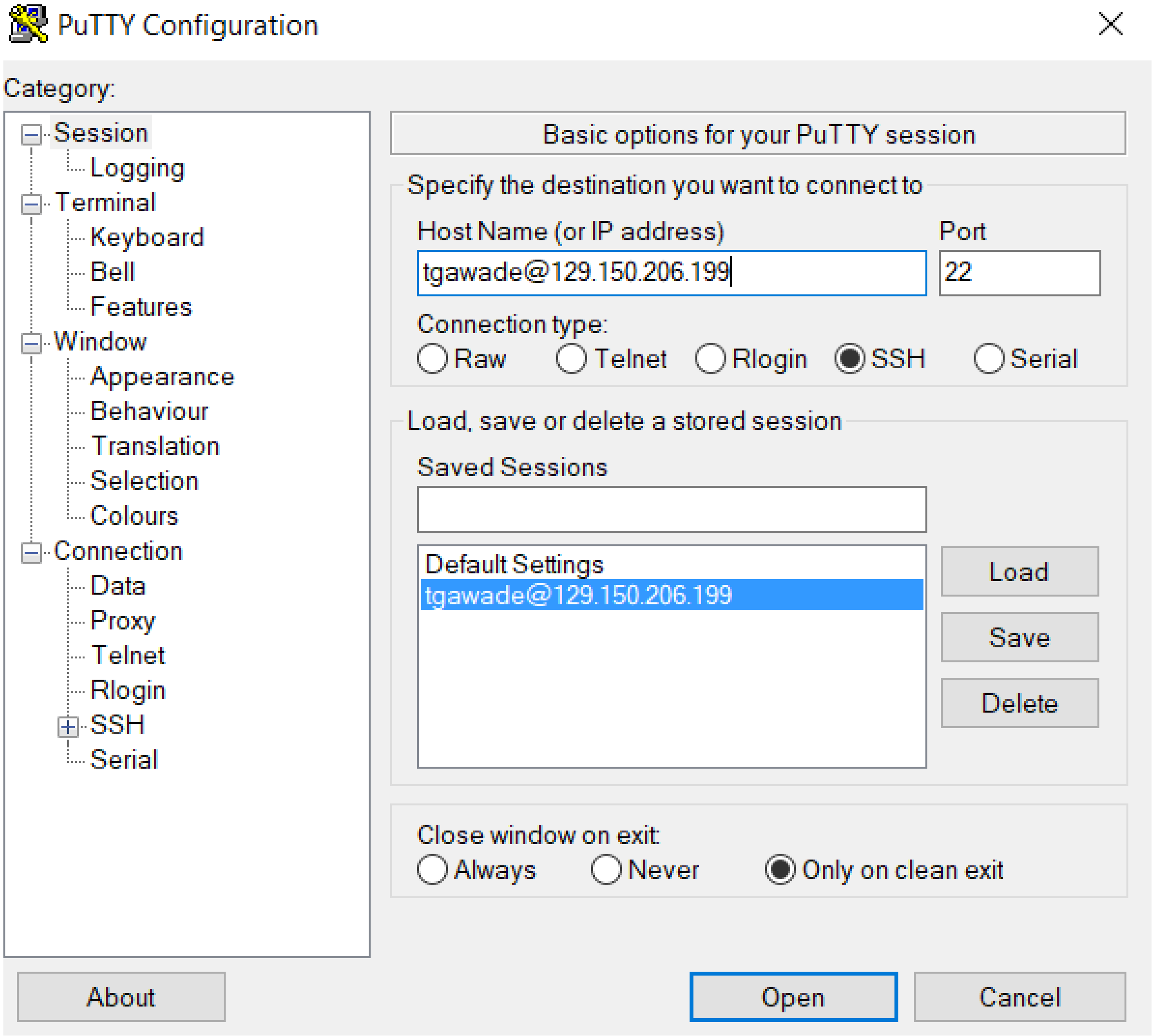
- Oracle Big Data Compute Edition: 5 nodes
- CPUs: 10
- CPU speed: 2.20GHz
- Memory: 150 GB
- Storage: 678 BG
- HDFS Capacity: 147 GB

PREREQUISITES

- You must have Microsoft Excel 2010, 2013 or 2016 installed.
- You must have your Excel 3D-Map enabled.
- Tableau 10.3 installed for visualization of the analyzed data.
- Power BI Desktop Version
- Oracle Big Data Compute Edition: 5 nodes

DOWNLOAD THE DATASET

This step is to get data manually. You need to remotely access your Oracle Cloud Big Data Compute Editions that you executed in your Oracle Cloud account using ssh using the information - ip address and connect command in beeline CLI-



Specify the destination you want to connect to

Host Name (or IP address): tgawade@129.150.206.199

Port: 22

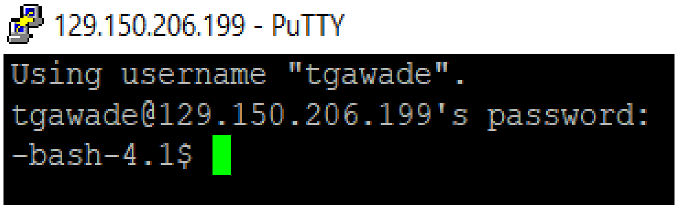
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
tgawade@129.150.206.199

Close window on exit: ○ Always ○ Never ● Only on clean exit



```
Using username "tgawade".
tgawade@129.150.206.199's password:
-bash-4.1$
```

1. ArticleYear2017- https://raw.githubusercontent.com/tanvigawade/April2017/master/ArticleYear2017.txt

2. ArticleYear2018- https://raw.githubusercontent.com/tanvigawade/April2017/master/ArticleYear2018.txt

3. CommentYear2017- https://www.dropbox.com/s/v0zqfog8pmque6g/CommentYear2017%20.txt?dl=0

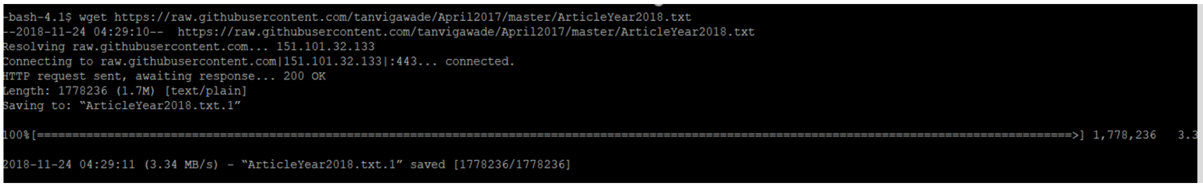
4. CommentYear2018- https://www.dropbox.com/s/mj4by2421kptba2/CommentYear2018.txt?dl=0

UPLOAD TXT FILE TO HADOOP DIRECTORY

Before uploading the TXT file to Hadoop directory, we need to first transfer it to local directory using below commands.

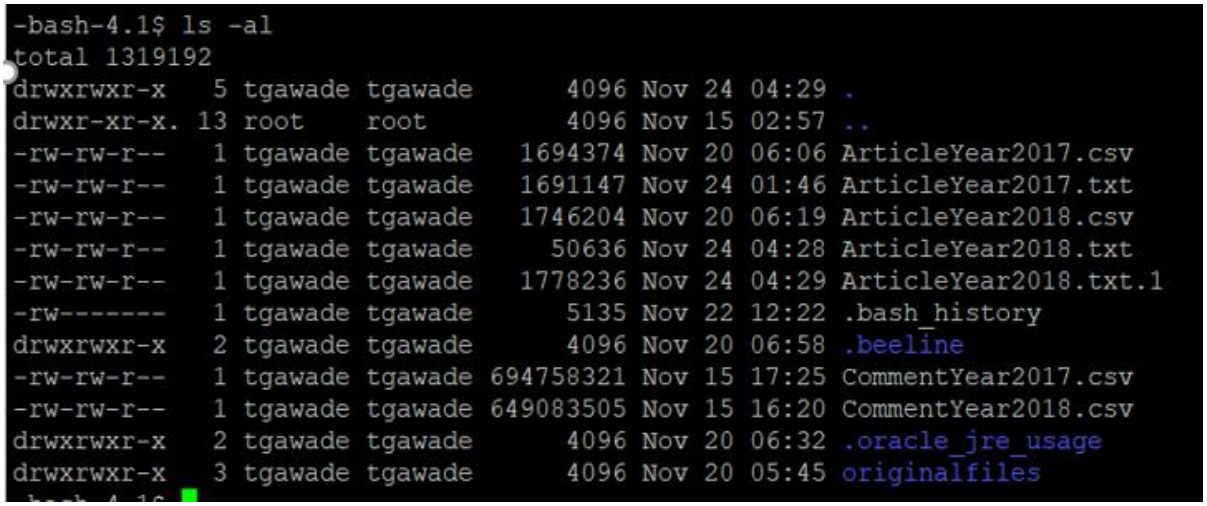
Note: Change the path and username.

```
wget https://raw.githubusercontent.com/tanvigawade/April2017/master/ArticleYear2017.txt

wget https://raw.githubusercontent.com/tanvigawade/April2017/master/ArticleYear2018.txt
```



```
-bash-4.1$ wget https://raw.githubusercontent.com/tanvigawade/April2017/master/ArticleYear2018.txt
--2018-11-24 04:29:10--  https://raw.githubusercontent.com/tanvigawade/April2017/master/ArticleYear2018.txt
Resolving raw.githubusercontent.com... 151.101.32.133
Connecting to raw.githubusercontent.com|151.101.32.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1778236 (1.7M) [text/plain]
Saving to: “ArticleYear2018.txt.1”

100%[=====================================================================================================================>] 1,778,236   3.3

2018-11-24 04:29:11 (3.34 MB/s) - “ArticleYear2018.txt.1” saved [1778236/1778236]
```

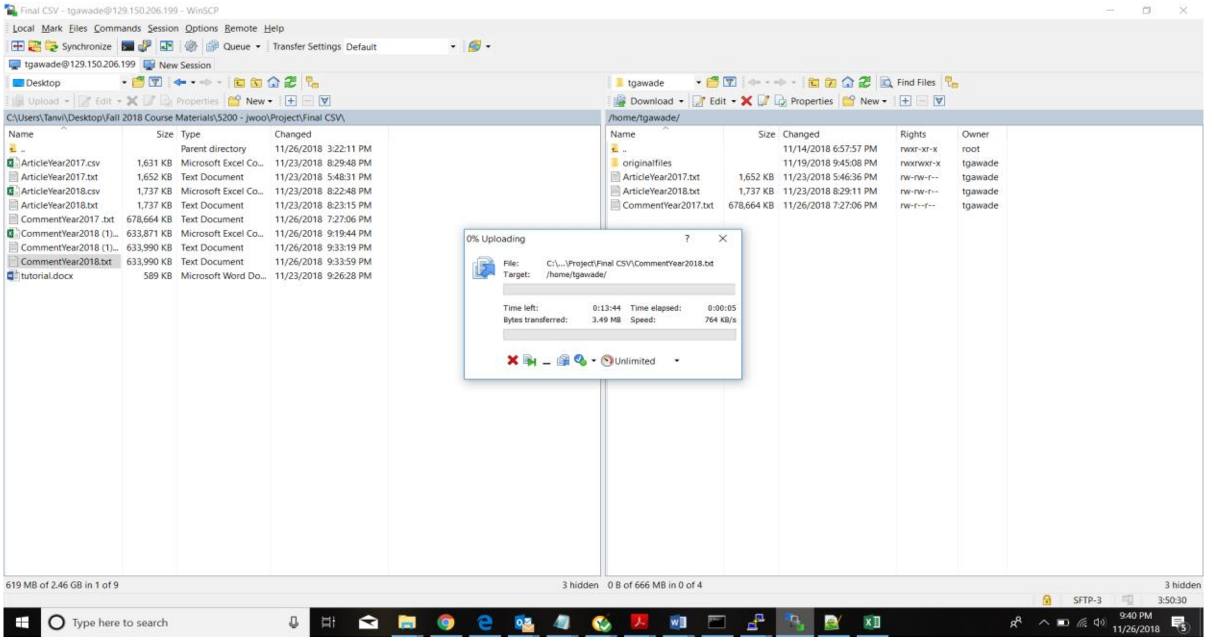
```
$ ls -al
```



```
-bash-4.1$ ls -al
total 1319192
drwxrwxr-x   5 tgawade tgawade      4096 Nov 24 04:29 .
drwxr-xr-x. 13 root    root         4096 Nov 15 02:57 ..
-rw-rw-r--   1 tgawade tgawade   1694374 Nov 20 06:06 ArticleYear2017.csv
-rw-rw-r--   1 tgawade tgawade   1691147 Nov 24 01:46 ArticleYear2017.txt
-rw-rw-r--   1 tgawade tgawade   1746204 Nov 20 06:19 ArticleYear2018.csv
-rw-rw-r--   1 tgawade tgawade     50636 Nov 24 04:28 ArticleYear2018.txt
-rw-rw-r--   1 tgawade tgawade   1778236 Nov 24 04:29 ArticleYear2018.txt.1
-rw-------   1 tgawade tgawade      5135 Nov 22 12:22 .bash_history
drwxrwxr-x   2 tgawade tgawade      4096 Nov 20 06:58 .beeline
-rw-rw-r--   1 tgawade tgawade 694758321 Nov 15 17:25 CommentYear2017.csv
-rw-rw-r--   1 tgawade tgawade 649083505 Nov 15 16:20 CommentYear2018.csv
drwxrwxr-x   2 tgawade tgawade      4096 Nov 20 06:32 .oracle_jre_usage
drwxrwxr-x   3 tgawade tgawade      4096 Nov 20 05:45 originalfiles
-bash-4.1$
```

Repeat “Step 2” for ArticleYear2018.

Since the CommentYear2017 and CommentYear2018 are more than 25MB we downloaded the datasets to local directory using WinSCP software. We also uploaded the dictionary data set using the same methodology.


Final CSV - tgawade@129.150.206.199 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
tgawade@129.150.206.199
New Session
Desktop
Upload
Edit
Properties
New
tgawade
Find Files
Download
Edit
Properties
New

C:\Users\Tanvi\Desktop\Fall 2018 Course Materials\5200 - jwoo\Project\Final CSV\

Name	Size	Type	Changed
..		Parent directory	11/26/2018 3:22:11 PM
ArticleYear2017.csv	1,631 KB	Microsoft Excel Co...	11/23/2018 8:29:48 PM
ArticleYear2017.txt	1,652 KB	Text Document	11/23/2018 5:48:31 PM
ArticleYear2018.csv	1,737 KB	Microsoft Excel Co...	11/23/2018 8:22:48 PM
ArticleYear2018.txt	1,737 KB	Text Document	11/23/2018 8:23:15 PM
CommentYear2017 .txt	678,664 KB	Text Document	11/26/2018 7:27:06 PM
CommentYear2018 (1)...	633,871 KB	Microsoft Excel Co...	11/26/2018 9:19:44 PM
CommentYear2018 (1)...	633,990 KB	Text Document	11/26/2018 9:33:19 PM
CommentYear2018.txt	633,990 KB	Text Document	11/26/2018 9:33:59 PM
tutorial.docx	589 KB	Microsoft Word Do...	11/23/2018 9:26:28 PM

/home/tgawade/

Name	Size	Changed	Rights	Owner
..		11/14/2018 6:57:57 PM	rwxr-xr-x	root
originalfiles		11/19/2018 9:45:08 PM	rwxrwxr-x	tgawade
ArticleYear2017.txt	1,652 KB	11/23/2018 5:46:36 PM	rw-rw-r--	tgawade
ArticleYear2018.txt	1,737 KB	11/23/2018 8:29:11 PM	rw-rw-r--	tgawade
CommentYear2017.txt	678,664 KB	11/26/2018 7:27:06 PM	rw-r--r--	tgawade

0% Uploading

File: C:\...\Project\Final CSV\CommentYear2018.txt
Target: /home/tgawade/

Time left: 0:13:44 Time elapsed: 0:00:05
Bytes transferred: 3.49 MB Speed: 764 KB/s

Unlimited

619 MB of 2.46 GB in 1 of 9 3 hidden 0 B of 666 MB in 0 of 4 3 hidden

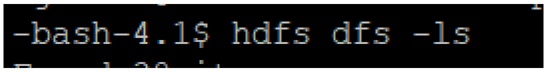
SFTP-3 3:50:30

Type here to search 9:40 PM 11/26/2018

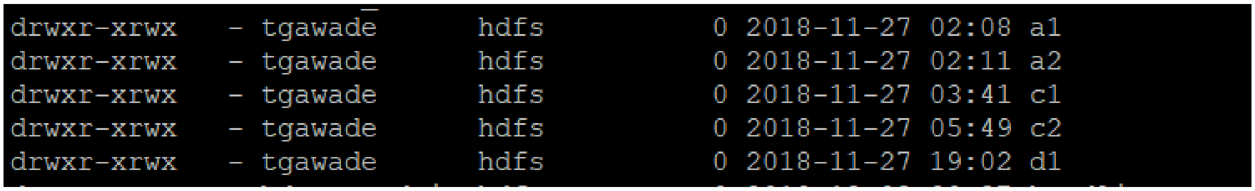
Now we have to upload all the TXT files to HDFS folder. Run the following HDFS commands to create and list the a1,a2,c1,c2,d1 and d directories in HDFS.

```
Hdfs dfs -mkdir /user/tgawade/a1

Hdfs dfs -mkdir /user/tgawade/a2

Hdfs dfs -mkdir /user/tgawade/c1

Hdfs dfs -mkdir /user/tgawade/c2

Hdfs dfs -mkdir /user/tgawade/d1

hdfs dfs -put CommentYear2017.txt /user/tgawade/c1/

hdfs dfs -put CommentYear2018.txt /user/tgawade/c2/

hdfs dfs -put ArticleYear2017.txt /user/tgawade/a1/

hdfs dfs -put ArticleYear2018.txt /user/tgawade/c2/

hdfs dfs -put dictionary.txt /user/tgawade/d1/
```

st



```
-bash-4.1$ hdfs dfs -ls
```



```
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 02:08 a1
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 02:11 a2
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 03:41 c1
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 05:49 c2
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 19:02 d1
```

Give permissions

Run the following HDFS command to make your beeline command works:

```
-bash-4.1$ hdfs dfs -chmod -R o+w /user/tgawade/c1/
```

```
-bash-4.1$ hdfs dfs -chmod -R o+w /user/tgawade/c2/
```

```
-bash-4.1$ hdfs dfs -chmod -R o+w /user/tgawade/a1/
```

```
-bash-4.1$ hdfs dfs -chmod -R o+w /user/tgawade/a1/
```

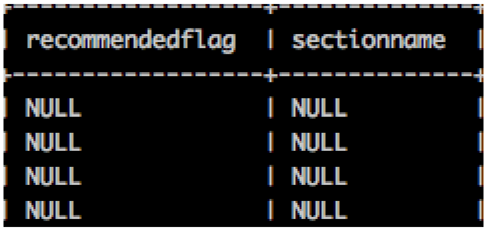
```
-bash-4.1$ hdfs dfs -chmod -R o+w /user/tgawade/d1/
```

DATA CLEANING

Removing Null Values

Null values were removed from tables. For example, section name and replycount columns had null values as shown below:

Before:

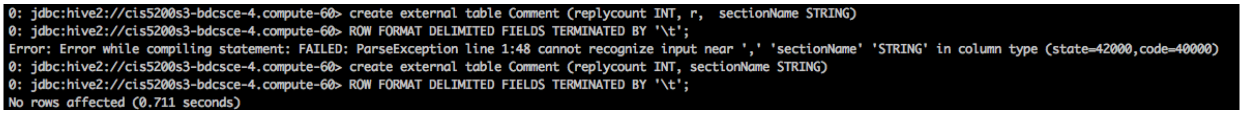


```
+------------------+---------------+
| recommendedflag  | sectionname   |
+------------------+---------------+
| NULL             | NULL          |
| NULL             | NULL          |
| NULL             | NULL          |
| NULL             | NULL          |
```

Below steps were performed to remove null values

Step 1:

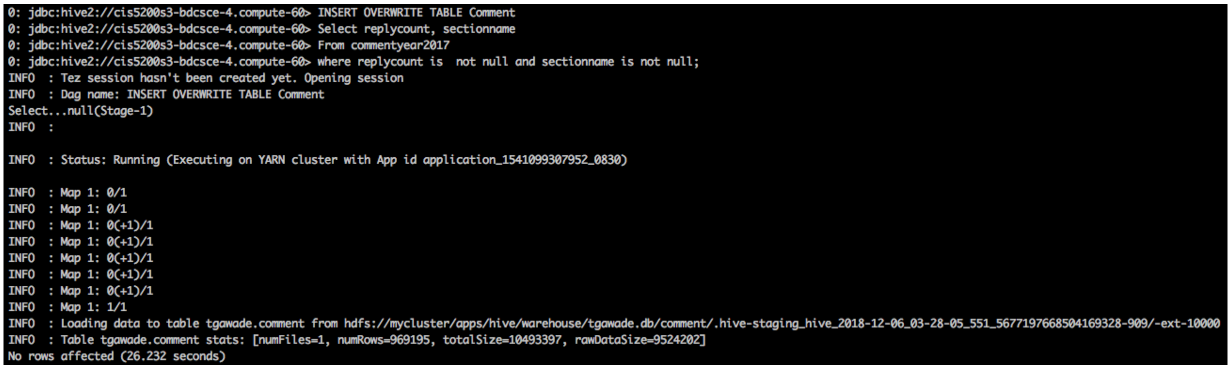
```
#External Table was created
create external table Comment (replycount INT, sectionName STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create external table Comment (replycount INT, r,  sectionName STRING)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
Error: Error while compiling statement: FAILED: ParseException line 1:48 cannot recognize input near ',' 'sectionName' 'STRING' in column type (state=42000,code=40000)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create external table Comment (replycount INT, sectionName STRING)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
No rows affected (0.711 seconds)
```

Step 2:

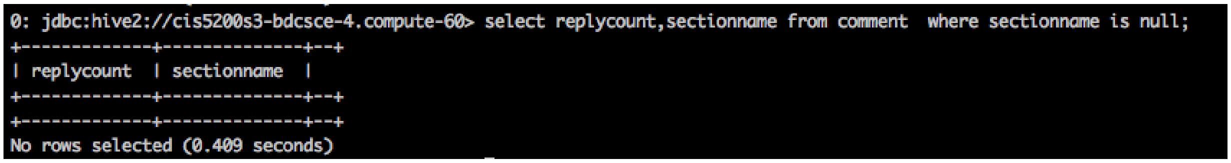
```
#Inserted data from original table

INSERT OVERWRITE TABLE Comment
Select replycount, sectionname
From commentyear2017
where replycount is not null and sectionname is not null;
```



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> INSERT OVERWRITE TABLE Comment
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> Select replycount, sectionname
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> From commentyear2017
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> where replycount is  not null and sectionname is not null;
INFO  : Tez session hasn't been created yet. Opening session
INFO  : Dag name: INSERT OVERWRITE TABLE Comment
Select...null(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0830)

INFO  : Map 1: 0/1
INFO  : Map 1: 0/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 1/1
INFO  : Loading data to table tgawade.comment from hdfs://mycluster/apps/hive/warehouse/tgawade.db/comment/.hive-staging_hive_2018-12-06_03-28-05_551_5677197668504169328-909/-ext-10000
INFO  : Table tgawade.comment stats: [numFiles=1, numRows=969195, totalSize=10493397, rawDataSize=9524202]
No rows affected (26.232 seconds)
```

All null values from columns were removed. Rechecked using below query:

```
select replycount,sectionname from comment  where sectionname is null;
```



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select replycount,sectionname from comment  where sectionname is null;
+-------------+--------------+--+
| replycount  | sectionname  |
+-------------+--------------+--+
+-------------+--------------+--+
No rows selected (0.409 seconds)
```

CREATE HIVE TABLE TO QUERY NEW YORK TIMES DATA

Open beeline CLI (Command Line Shell Interface) that is equivalent to hive CLI environment as follows,

which you have done in the previous lab.

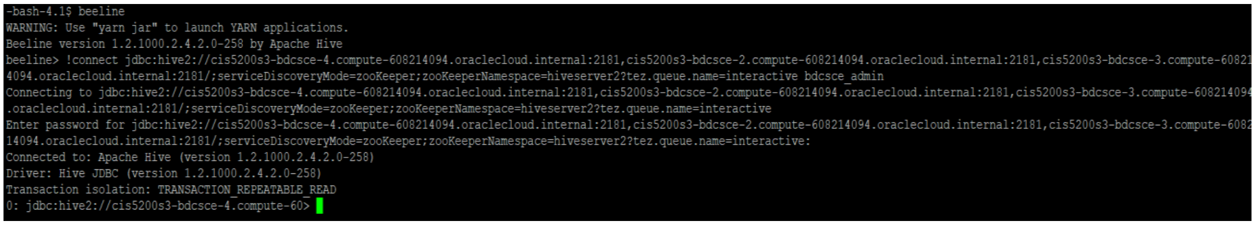
```
beeline
```

Beeline is for multiple user's access to Hive Server 2 of a Hadoop cluster.

Use the below command to connect to beeline:

!connect jdbc:hive2://cis5200s3-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-3.compute-608214094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcsce_admin.

NOTE: If you see "CLOSED" in the above beeline shell prompt, it is not connected to Hive Server 2.



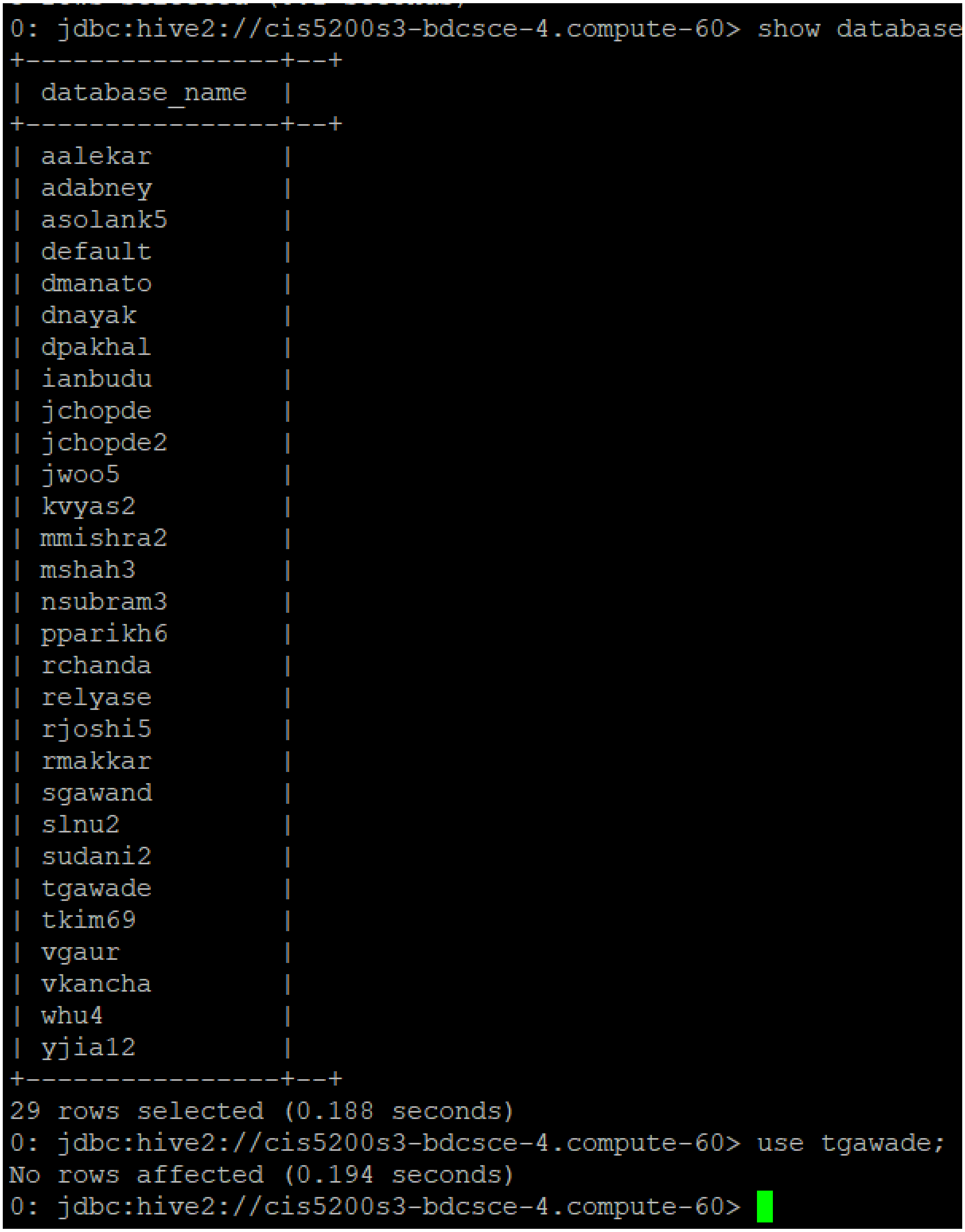
```
-bash-4.1$ beeline
WARNING: Use "yarn jar" to launch YARN applications.
Beeline version 1.2.1000.2.4.2.0-258 by Apache Hive
beeline> !connect jdbc:hive2://cis5200s3-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-3.compute-60821
4094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcsce_admin
Connecting to jdbc:hive2://cis5200s3-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-3.compute-608214094
.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive
Enter password for jdbc:hive2://cis5200s3-bdcsce-4.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-2.compute-608214094.oraclecloud.internal:2181,cis5200s3-bdcsce-3.compute-6082
14094.oraclecloud.internal:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive:
Connected to: Apache Hive (version 1.2.1000.2.4.2.0-258)
Driver: Hive JDBC (version 1.2.1000.2.4.2.0-258)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

NOTE: Now we have to create your own database with your username to separate your tables with other users you have to use your username. For example, the user should run the following command.

0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>CREATE DATABASE tgawade;

No rows affected (0.277 seconds)

0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> show databases;



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> show database
+-----------------+--+
| database_name   |
+-----------------+--+
| aalekar         |
| adabney         |
| asolank5        |
| default         |
| dmanato         |
| dnayak          |
| dpakhal         |
| ianbudu         |
| jchopde         |
| jchopde2        |
| jwoo5           |
| kvyas2          |
| mmishra2        |
| mshah3          |
| nsubram3        |
| pparikh6        |
| rchanda         |
| relyase         |
| rjoshi5         |
| rmakkar         |
| sgawand         |
| slnu2           |
| sudani2         |
| tgawade         |
| tkim69          |
| vgaur           |
| vkancha         |
| whu4            |
| yjia12          |
+-----------------+--+
29 rows selected (0.188 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> use tgawade;
No rows affected (0.194 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

The following hive statement creates an external table for ArticleYear2017, ArticleYear2018. External tables preserve the data in the original file format, while allowing Hive to perform queries against the data within the file.

In the hive shell CLI, you need to copy and paste the following HiveQL code to create an external table CommentYear2017.

```
create external table if not exists CommentYear2017(Month_Name STRING,approveDate
STRING,articleID STRING,articleWordCount BIGINT,commentBody STRING,commentID
STRING,commentSequence STRING,commentTitle STRING,commentType STRING,createDate
STRING,depth INT,editorsSelection INT,inReplyTo STRING,newDesk STRING,parentID
STRING,parentUserDisplayName STRING, permID STRING, picURL STRING, printPage INT,
recommendations INT, recommendedFlag INT, replyCount INT, reportAbuseFlag INT, sectionName
STRING, sharing INT, status STRING, timespeople INT, trusted INT, updateDate STRING,
userDisplayName STRING, userID STRING, userLocation STRING, userTitle STRING, userURL
STRING,typeofmaterial STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE location "/user/tgawade/c1/"
TBLPROPERTIES ('skip.header.line.count'='1');
```

In the hive shell CLI, you need to copy and paste the following HiveQL code to create an external table CommentYear2018.

```
create external table if not exists CommentYear2018(Month_Name STRING,approveDate
STRING,articleID STRING,articleWordCount BIGINT,commentBody STRING,commentID
STRING,commentSequence STRING,commentTitle STRING,commentType STRING,createDate
STRING,depth INT,editorsSelection INT,inReplyTo STRING,newDesk STRING,parentID
STRING,parentUserDisplayName STRING, permID STRING, picURL STRING, printPage INT,
recommendations INT, recommendedFlag INT, replyCount INT, reportAbuseFlag INT, sectionName
STRING, sharing INT, status STRING, timespeople INT, trusted INT, updateDate STRING,
userDisplayName STRING, userID STRING, userLocation STRING, userTitle STRING, userURL
STRING,typeofmaterial STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE location "/user/tgawade/c2/"
TBLPROPERTIES ('skip.header.line.count'='1');
```

In the hive shell CLI, you need to copy and paste the following HiveQL code to create an external table ArticleYear2017.

```
create external table if not exists articleyear2017(Month_Name STRING,articleID STRING,abstract
STRING,byline STRING,documentType STRING,headline STRING,keywords STRING,multimedia INT,
newDesk STRING,printPage INT,pubDate TIMESTAMP,source STRING,
typeOfMaterial STRING,
webURL STRING,
articleWordCount BIGINT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE location "/user/tgawade/a1/"
TBLPROPERTIES ('skip.header.line.count'='1');
```

In the hive shell CLI, you need to copy and paste the following HiveQL code to create an external table ArticleYear2018.

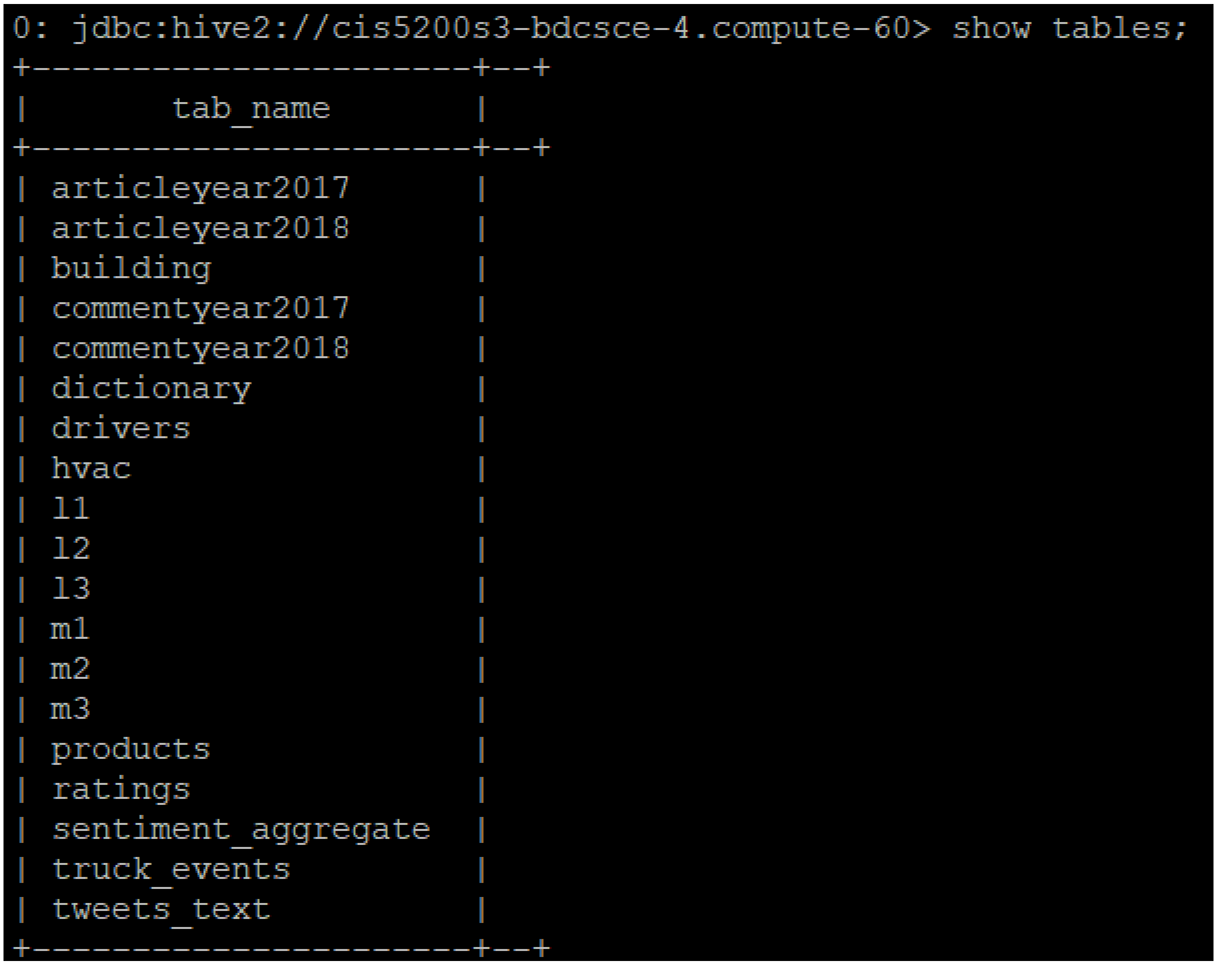
```
create external table if not exists articleyear2018(Month_Name STRING,articleID STRING,abstract
STRING,byline STRING,documentType STRING,headline STRING,keywords STRING,multimedia INT,
newDesk STRING,printPage INT,pubDate TIMESTAMP,source STRING,
typeOfMaterial STRING,
webURL STRING,
articleWordCount BIGINT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE location "/user/tgawade/a2/"
TBLPROPERTIES ('skip.header.line.count'='1');
```

In the hive shell CLI, you need to copy and paste the following HiveQL code to create an external table dictionary.

```
CREATE EXTERNAL TABLE if not exists dictionary (type string,length int,word string,pos string,
stemmed string,
polarity string )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION "/user/tgawade/d1/"
```

Now you may see if those tables are created with “show tables”:

0: jdbc:hive2://cis5200-bdcsce-4.compute-6082> show tables;



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> show tables;
+----------------------+--+
|       tab_name       |
+----------------------+--+
| articleyear2017      |
| articleyear2018      |
| building             |
| commentyear2017      |
| commentyear2018      |
| dictionary           |
| drivers              |
| hvac                 |
| l1                   |
| l2                   |
| l3                   |
| m1                   |
| m2                   |
| m3                   |
| products             |
| ratings              |
| sentiment_aggregate  |
| truck_events         |
| tweets_text          |
+----------------------+--+
```

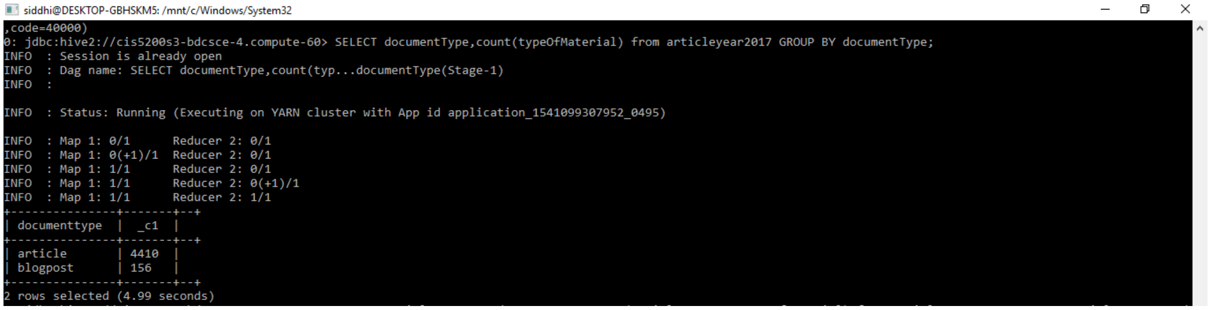
QUERYING ON THE DATASET

Query 1: Show the count of document type by type of material for the year 2017 and 2018?

In this query, we have tried to determine what number of articles and blogpost are present in NYT for both the years respectively.

For year 2017:

```
SELECT documentType,count(typeOfMaterial) from articleyear2017 GROUP BY documentType;
```

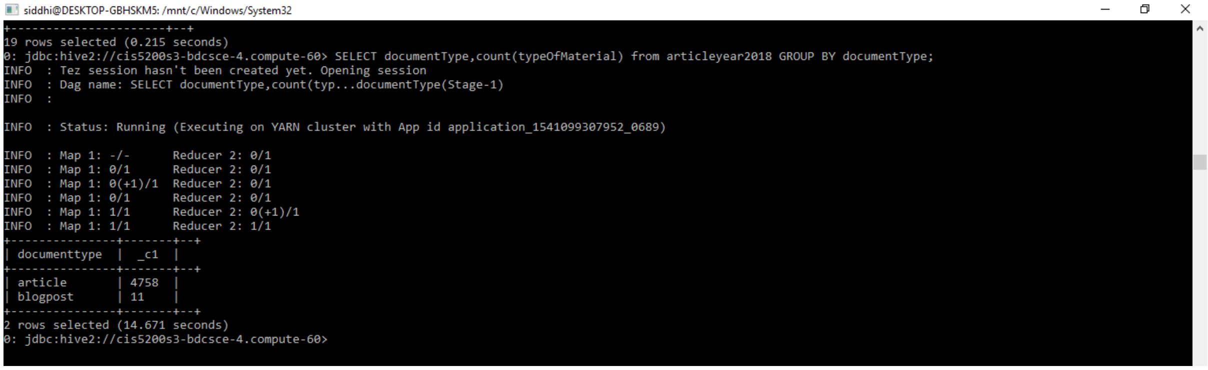


siddhi@DESKTOP-GBHSKM5: /mnt/c/Windows/System32

```
,code=40000)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT documentType,count(typeOfMaterial) from articleyear2017 GROUP BY documentType;
INFO  : Session is already open
INFO  : Dag name: SELECT documentType,count(typ...documentType(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0495)

INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1
+---------------+-------+--+
| documenttype  |  _c1  |
+---------------+-------+--+
| article       | 4410  |
| blogpost      | 156   |
+---------------+-------+--+
2 rows selected (4.99 seconds)
```

For year 2018:

```
SELECT documentType, count(typeOfMaterial) from articleyear2018 GROUP BY documentType;
```



siddhi@DESKTOP-GBHSKM5: /mnt/c/Windows/System32

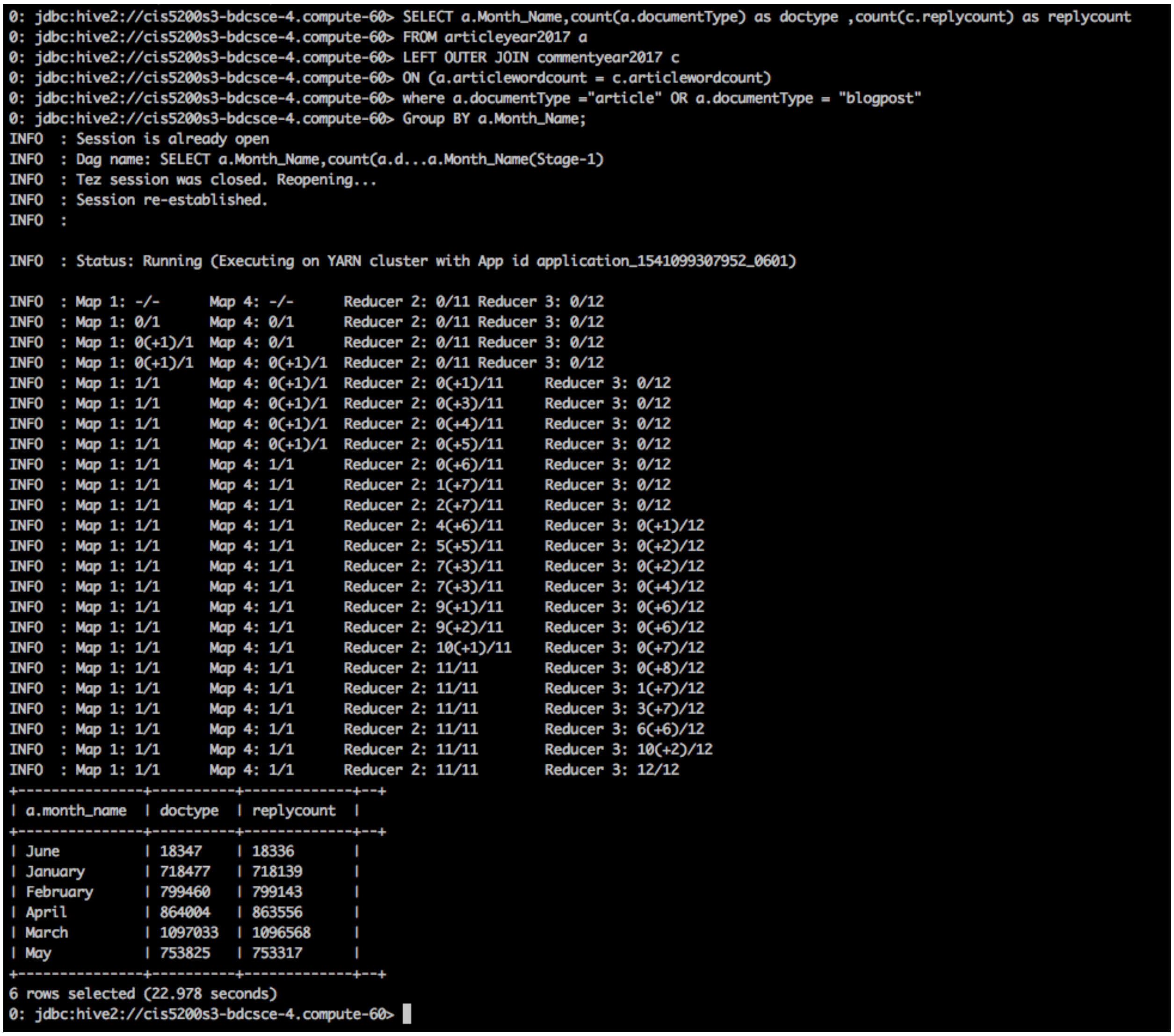
```
+----------------------+--+
19 rows selected (0.215 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT documentType,count(typeOfMaterial) from articleyear2018 GROUP BY documentType;
INFO  : Tez session hasn't been created yet. Opening session
INFO  : Dag name: SELECT documentType,count(typ...documentType(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0689)

INFO  : Map 1: -/-      Reducer 2: 0/1
INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1
INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1
+---------------+-------+--+
| documenttype  |  _c1  |
+---------------+-------+--+
| article       | 4758  |
| blogpost      | 11    |
+---------------+-------+--+
2 rows selected (14.671 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

Query 2: What is the reply count for the document type month wise for year 2017 & 2018?

Below query shows the reply count for each document type for month January to June for year 2017 & 2018. Month name is from articleyear2017 and reply count is from commentyear2017 table. Left outer join is used to get the desired output and is Grouped by month.

For Year 2017:

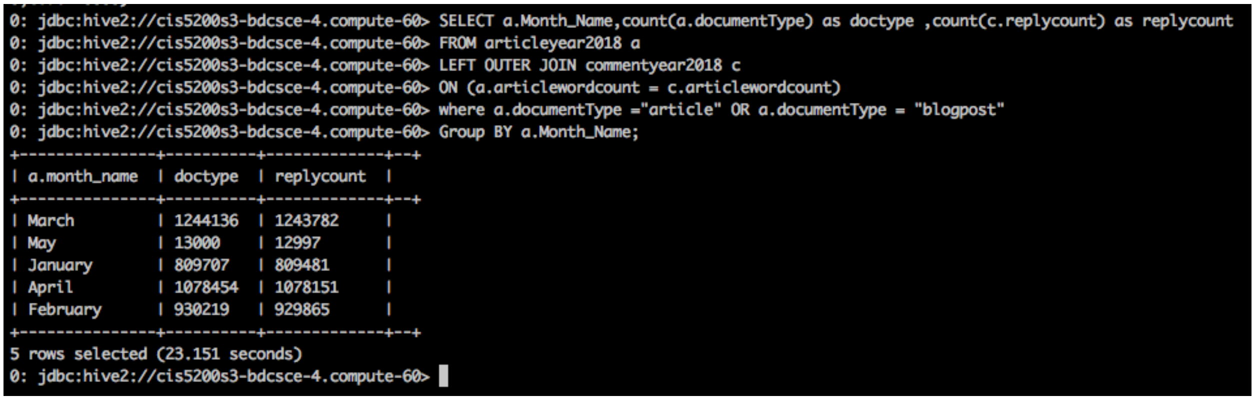
```
SELECT a.Month_Name,count(a.documentType) as doctype ,count(c.replycount) as replycount
FROM articleyear2017 a
LEFT OUTER JOIN commentyear2017 c
ON (a.articlewordcount = c.articlewordcount)
where a.documentType ="article" OR a.documentType = "blogpost"
Group BY a.Month_Name;
```



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT a.Month_Name,count(a.documentType) as doctype ,count(c.replycount) as replycount
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> FROM articleyear2017 a
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> LEFT OUTER JOIN commentyear2017 c
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> ON (a.articlewordcount = c.articlewordcount)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> where a.documentType ="article" OR a.documentType = "blogpost"
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> Group BY a.Month_Name;
INFO  : Session is already open
INFO  : Dag name: SELECT a.Month_Name,count(a.d...a.Month_Name(Stage-1)
INFO  : Tez session was closed. Reopening...
INFO  : Session re-established.
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0601)

INFO  : Map 1: -/-      Map 4: -/-      Reducer 2: 0/11 Reducer 3: 0/12
INFO  : Map 1: 0/1      Map 4: 0/1      Reducer 2: 0/11 Reducer 3: 0/12
INFO  : Map 1: 0(+1)/1  Map 4: 0/1      Reducer 2: 0/11 Reducer 3: 0/12
INFO  : Map 1: 0(+1)/1  Map 4: 0(+1)/1  Reducer 2: 0/11 Reducer 3: 0/12
INFO  : Map 1: 1/1      Map 4: 0(+1)/1  Reducer 2: 0(+1)/11     Reducer 3: 0/12
INFO  : Map 1: 1/1      Map 4: 0(+1)/1  Reducer 2: 0(+3)/11     Reducer 3: 0/12
INFO  : Map 1: 1/1      Map 4: 0(+1)/1  Reducer 2: 0(+4)/11     Reducer 3: 0/12
INFO  : Map 1: 1/1      Map 4: 0(+1)/1  Reducer 2: 0(+5)/11     Reducer 3: 0/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 0(+6)/11     Reducer 3: 0/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 1(+7)/11     Reducer 3: 0/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 2(+7)/11     Reducer 3: 0/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 4(+6)/11     Reducer 3: 0(+1)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 5(+5)/11     Reducer 3: 0(+2)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 7(+3)/11     Reducer 3: 0(+2)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 7(+3)/11     Reducer 3: 0(+4)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 9(+1)/11     Reducer 3: 0(+6)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 9(+2)/11     Reducer 3: 0(+6)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 10(+1)/11    Reducer 3: 0(+7)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 11/11        Reducer 3: 0(+8)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 11/11        Reducer 3: 1(+7)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 11/11        Reducer 3: 3(+7)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 11/11        Reducer 3: 6(+6)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 11/11        Reducer 3: 10(+2)/12
INFO  : Map 1: 1/1      Map 4: 1/1      Reducer 2: 11/11        Reducer 3: 12/12
+---------------+----------+-------------+--+
| a.month_name  | doctype  | replycount  |
+---------------+----------+-------------+--+
| June          | 18347    | 18336       |
| January       | 718477   | 718139      |
| February      | 799460   | 799143      |
| April         | 864004   | 863556      |
| March         | 1097033  | 1096568     |
| May           | 753825   | 753317      |
+---------------+----------+-------------+--+
6 rows selected (22.978 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

For Year 2018:

```
SELECT a.Month_Name,count(a.documentType) as doctype ,count(c.replycount) as replycount
FROM articleyear2018 a
LEFT OUTER JOIN commentyear2018 c
ON (a.articlewordcount = c.articlewordcount)
where a.documentType ="article" OR a.documentType = "blogpost"
Group BY a.Month_Name;
```



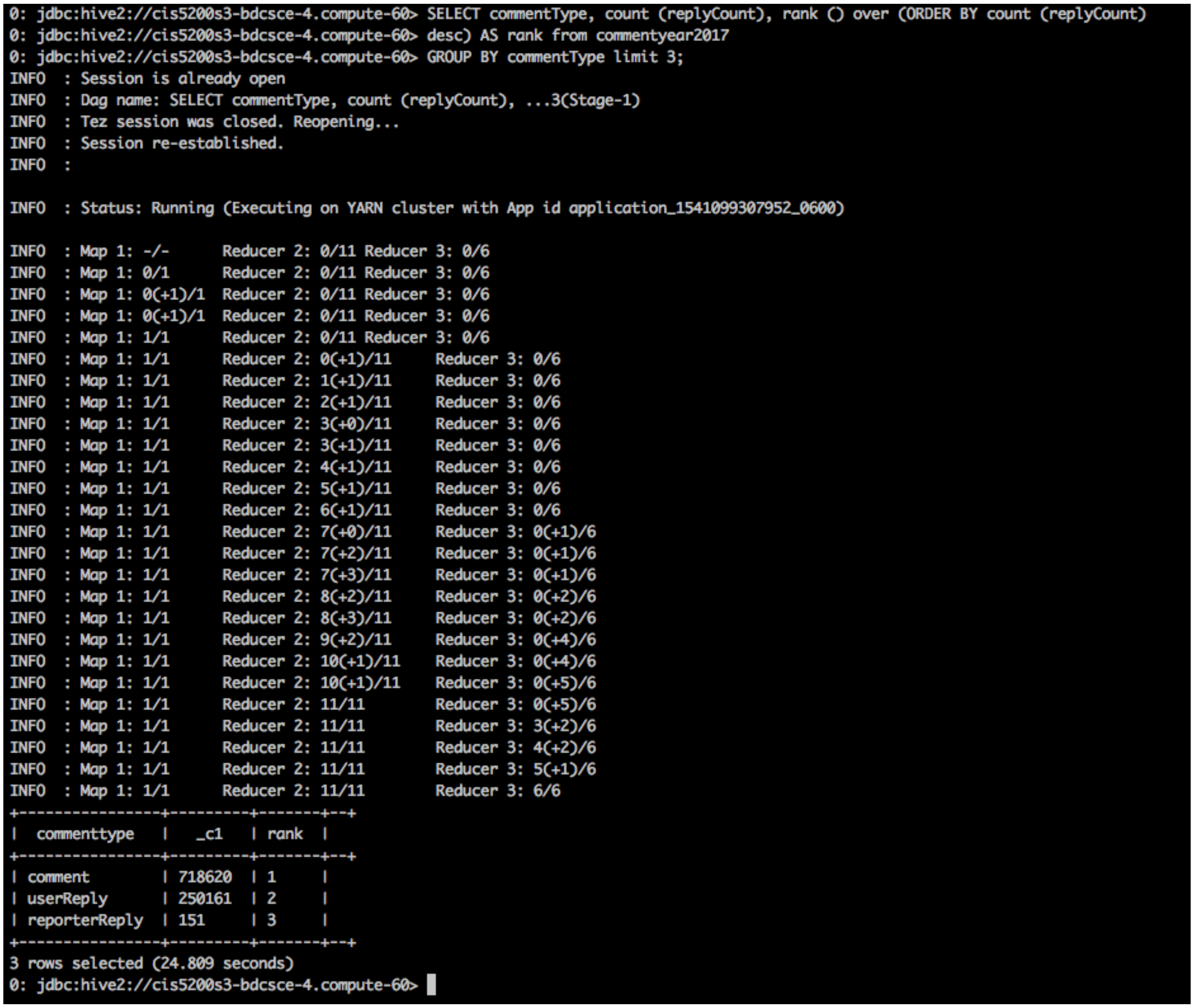
```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT a.Month_Name,count(a.documentType) as doctype ,count(c.replycount) as replycount
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> FROM articleyear2018 a
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> LEFT OUTER JOIN commentyear2018 c
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> ON (a.articlewordcount = c.articlewordcount)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> where a.documentType ="article" OR a.documentType = "blogpost"
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> Group BY a.Month_Name;
+---------------+----------+-------------+--+
| a.month_name  | doctype  | replycount  |
+---------------+----------+-------------+--+
| March         | 1244136  | 1243782     |
| May           | 13000    | 12997       |
| January       | 809707   | 809481      |
| April         | 1078454  | 1078151     |
| February      | 930219   | 929865      |
+---------------+----------+-------------+--+
5 rows selected (23.151 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

Query 3: What is the reply count for each comment type.

Below query shows the reply count received for top 3 comment type for year 2017 & 2018. Rank is used to get the desired output serially and is Grouped by comment type.

For Year 2017:

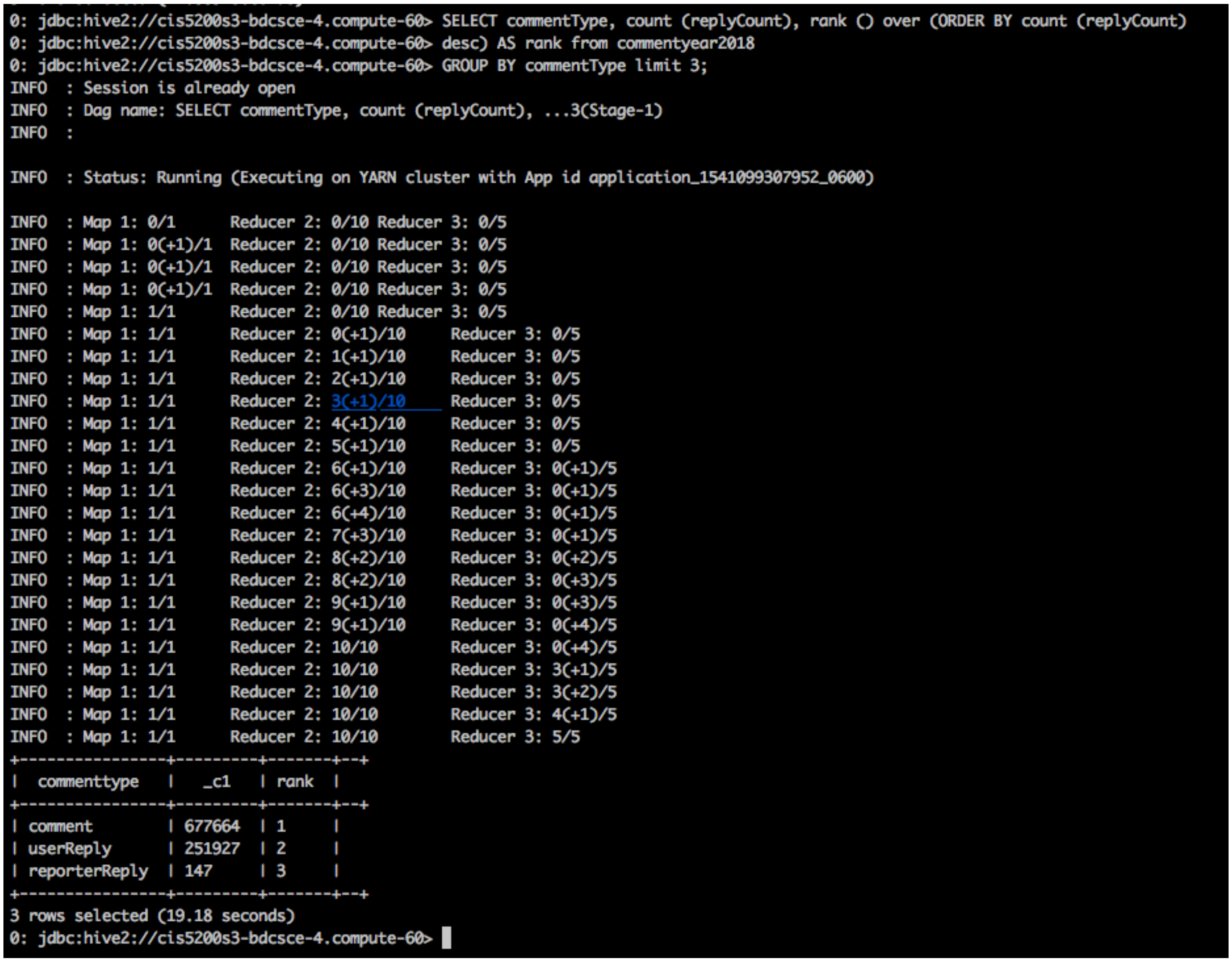
```
SELECT commentType, count (replyCount), rank () over (ORDER BY count (replyCount)
desc) AS rank from commentyear2017
GROUP BY commentType limit 3;
```



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT commentType, count (replyCount), rank () over (ORDER BY count (replyCount)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> desc) AS rank from commentyear2017
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> GROUP BY commentType limit 3;
INFO  : Session is already open
INFO  : Dag name: SELECT commentType, count (replyCount), ...3(Stage-1)
INFO  : Tez session was closed. Reopening...
INFO  : Session re-established.
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0600)

INFO  : Map 1: -/-      Reducer 2: 0/11 Reducer 3: 0/6
INFO  : Map 1: 0/1      Reducer 2: 0/11 Reducer 3: 0/6
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/11 Reducer 3: 0/6
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/11 Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 0/11 Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 1(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 2(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 3(+0)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 3(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 4(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 5(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 6(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 7(+0)/11     Reducer 3: 0(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 7(+2)/11     Reducer 3: 0(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 7(+3)/11     Reducer 3: 0(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 8(+2)/11     Reducer 3: 0(+2)/6
INFO  : Map 1: 1/1      Reducer 2: 8(+3)/11     Reducer 3: 0(+2)/6
INFO  : Map 1: 1/1      Reducer 2: 9(+2)/11     Reducer 3: 0(+4)/6
INFO  : Map 1: 1/1      Reducer 2: 10(+1)/11    Reducer 3: 0(+4)/6
INFO  : Map 1: 1/1      Reducer 2: 10(+1)/11    Reducer 3: 0(+5)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 0(+5)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 3(+2)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 4(+2)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 5(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 6/6
+---------------+---------+-------+--+
|  commenttype  |   _c1   | rank  |
+---------------+---------+-------+--+
| comment       | 718620  | 1     |
| userReply     | 250161  | 2     |
| reporterReply | 151     | 3     |
+---------------+---------+-------+--+
3 rows selected (24.809 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

For Year 2018:

```
SELECT commentType, count (replyCount), rank () over (ORDER BY count (replyCount)
desc) AS rank from commentyear2018
GROUP BY commentType limit 3;
```



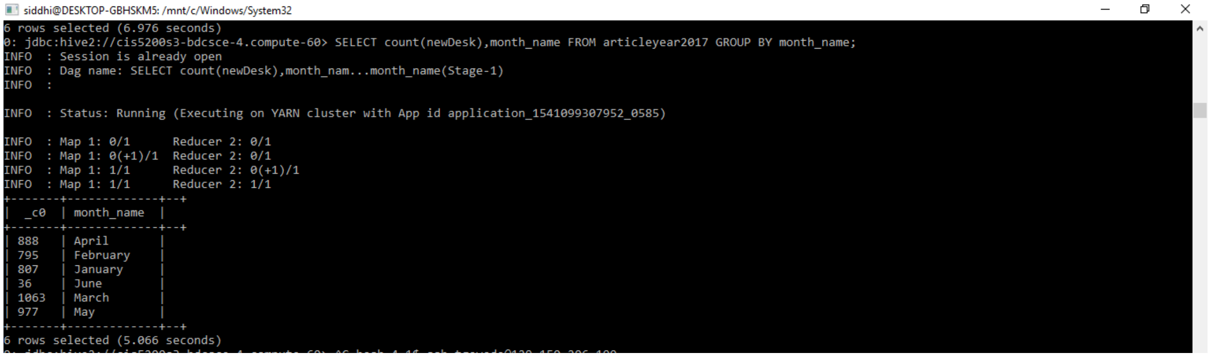
```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT commentType, count (replyCount), rank () over (ORDER BY count (replyCount)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> desc) AS rank from commentyear2018
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> GROUP BY commentType limit 3;
INFO  : Session is already open
INFO  : Dag name: SELECT commentType, count (replyCount), ...3(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0600)

INFO  : Map 1: 0/1      Reducer 2: 0/10 Reducer 3: 0/5
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/10 Reducer 3: 0/5
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/10 Reducer 3: 0/5
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/10 Reducer 3: 0/5
INFO  : Map 1: 1/1      Reducer 2: 0/10 Reducer 3: 0/5
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/10     Reducer 3: 0/5
INFO  : Map 1: 1/1      Reducer 2: 1(+1)/10     Reducer 3: 0/5
INFO  : Map 1: 1/1      Reducer 2: 2(+1)/10     Reducer 3: 0/5
INFO  : Map 1: 1/1      Reducer 2: 3(+1)/10     Reducer 3: 0/5
INFO  : Map 1: 1/1      Reducer 2: 4(+1)/10     Reducer 3: 0/5
INFO  : Map 1: 1/1      Reducer 2: 5(+1)/10     Reducer 3: 0/5
INFO  : Map 1: 1/1      Reducer 2: 6(+1)/10     Reducer 3: 0(+1)/5
INFO  : Map 1: 1/1      Reducer 2: 6(+3)/10     Reducer 3: 0(+1)/5
INFO  : Map 1: 1/1      Reducer 2: 6(+4)/10     Reducer 3: 0(+1)/5
INFO  : Map 1: 1/1      Reducer 2: 7(+3)/10     Reducer 3: 0(+1)/5
INFO  : Map 1: 1/1      Reducer 2: 8(+2)/10     Reducer 3: 0(+2)/5
INFO  : Map 1: 1/1      Reducer 2: 8(+2)/10     Reducer 3: 0(+3)/5
INFO  : Map 1: 1/1      Reducer 2: 9(+1)/10     Reducer 3: 0(+3)/5
INFO  : Map 1: 1/1      Reducer 2: 9(+1)/10     Reducer 3: 0(+4)/5
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 0(+4)/5
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 3(+1)/5
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 3(+2)/5
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 4(+1)/5
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 5/5
+----------------+---------+-------+--+
|  commenttype   |   _c1   | rank  |
+----------------+---------+-------+--+
| comment        | 677664  | 1     |
| userReply      | 251927  | 2     |
| reporterReply  | 147     | 3     |
+----------------+---------+-------+--+
3 rows selected (19.18 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

Query 4: What is the count of new desk month wise?

NewDesk is a column which has various field values like letter, foreign, editorial, brief, etc. In this query we have tried to find out the count of NewDesk received for both the years month wise.

For year 2017:

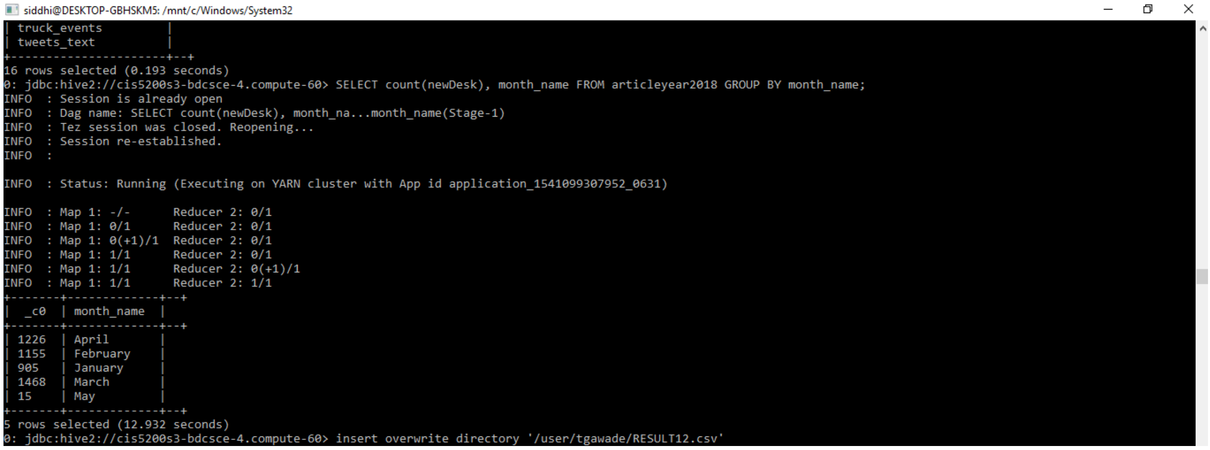
```
SELECT count(newDesk),month_name FROM articleyear2017 GROUP BY month_name;
```



```
6 rows selected (6.976 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT count(newDesk),month_name FROM articleyear2017 GROUP BY month_name;
INFO  : Session is already open
INFO  : Dag name: SELECT count(newDesk),month_nam...month_name(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0585)

INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1
+-------+------------+--+
|  _c0  | month_name  |
+-------+------------+--+
| 888   | April       |
| 795   | February    |
| 807   | January     |
| 36    | June        |
| 1063  | March       |
| 977   | May         |
+-------+------------+--+
6 rows selected (5.066 seconds)
```

For year 2018:

```
SELECT count(newDesk),month_name FROM articleyear2018 GROUP BY month_name;
```



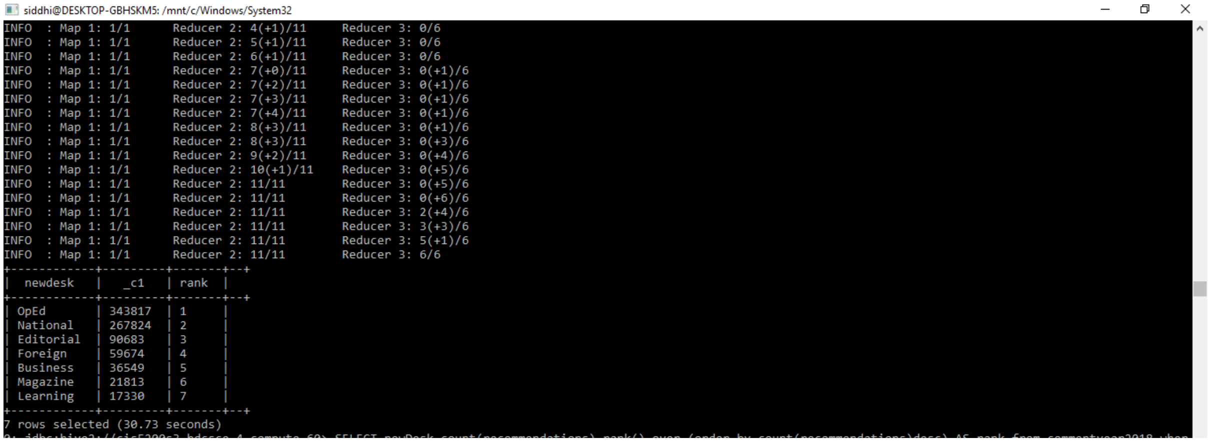
```
| truck_events          |
| tweets_text           |
+----------------------+--+
16 rows selected (0.193 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT count(newDesk), month_name FROM articleyear2018 GROUP BY month_name;
INFO  : Session is already open
INFO  : Dag name: SELECT count(newDesk), month_na...month_name(Stage-1)
INFO  : Tez session was closed. Reopening...
INFO  : Session re-established.
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0631)

INFO  : Map 1: -/-      Reducer 2: 0/1
INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1
+-------+------------+--+
|  _c0  | month_name  |
+-------+------------+--+
| 1226  | April       |
| 1155  | February    |
| 905   | January     |
| 1468  | March       |
| 15    | May         |
+-------+------------+--+
5 rows selected (12.932 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> insert overwrite directory '/user/tgawade/RESULT12.csv'
```

Query 5: What is the count of new desk based on recommendations?

As explained above that newDesk has various filed values and each of them receive some sort of recommendations from the people, which we have shown in the query below for both the years.

For year 2017:

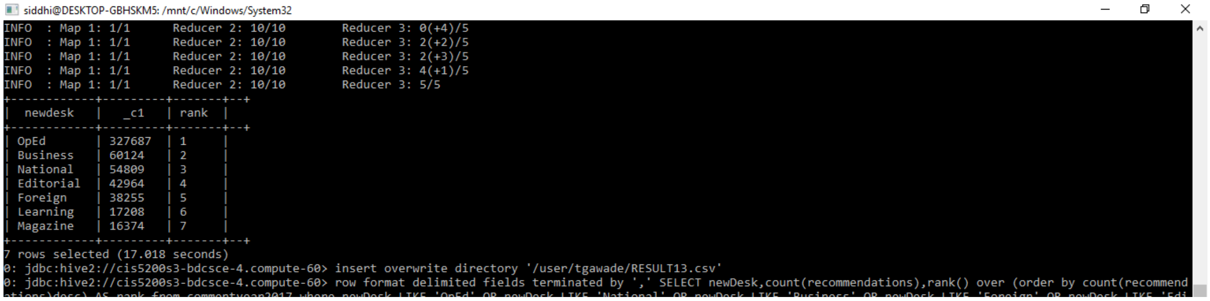
```
SELECT newDesk,count(recommendations),rank() over (order by  count(recommendations)desc)
AS rank from commentyear2017 where newDesk LIKE 'OpEd' OR newDesk LIKE 'National' OR
newDesk LIKE 'Business' OR newDesk LIKE 'Foreign' OR newDesk LIKE 'Editorial' OR newDesk LIKE
'Magazine' OR newDesk LIKE 'Learning'  GROUP BY newDesk;
```



```
INFO  : Map 1: 1/1      Reducer 2: 4(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 5(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 6(+1)/11     Reducer 3: 0/6
INFO  : Map 1: 1/1      Reducer 2: 7(+0)/11     Reducer 3: 0(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 7(+2)/11     Reducer 3: 0(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 7(+3)/11     Reducer 3: 0(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 7(+4)/11     Reducer 3: 0(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 8(+3)/11     Reducer 3: 0(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 8(+3)/11     Reducer 3: 0(+3)/6
INFO  : Map 1: 1/1      Reducer 2: 9(+2)/11     Reducer 3: 0(+4)/6
INFO  : Map 1: 1/1      Reducer 2: 10(+1)/11    Reducer 3: 0(+5)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 0(+5)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 0(+6)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 2(+4)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 3(+3)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 5(+1)/6
INFO  : Map 1: 1/1      Reducer 2: 11/11        Reducer 3: 6/6
+------------+---------+-------+--+
|  newdesk   |   _c1   | rank  |
+------------+---------+-------+--+
| OpEd       | 343817  | 1     |
| National   | 267824  | 2     |
| Editorial  | 90683   | 3     |
| Foreign    | 59674   | 4     |
| Business   | 36549   | 5     |
| Magazine   | 21813   | 6     |
| Learning   | 17330   | 7     |
+------------+---------+-------+--+
7 rows selected (30.73 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT newDesk,count(recommendations),rank() over (order by count(recommendations)desc) AS rank from commentyear2018 wher
```

For year 2018:

```
SELECT newDesk,count(recommendations),rank() over (order by count(recommendations)desc) AS rank from commentyear2018 where newDesk LIKE 'OpEd' OR newDesk LIKE 'National' OR newDesk LIKE 'Business' OR newDesk LIKE 'Foreign' OR newDesk LIKE 'Editorial' OR newDesk LIKE 'Magazine' OR newDesk LIKE 'Learning' GROUP BY newDesk;
```

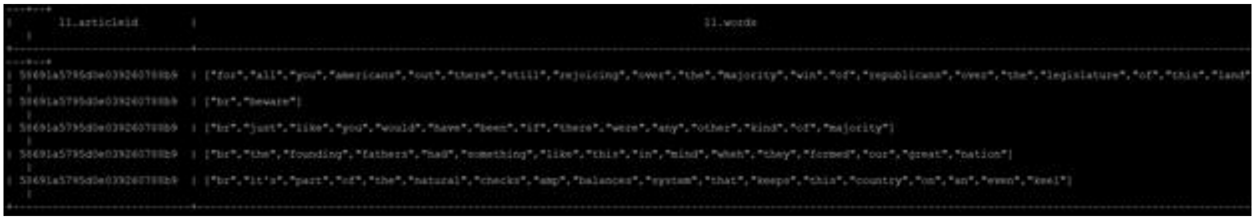


```
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 0(+4)/5
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 2(+2)/5
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 2(+3)/5
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 4(+1)/5
INFO  : Map 1: 1/1      Reducer 2: 10/10        Reducer 3: 5/5
+------------+---------+-------+--+
|  newdesk   |   _c1   | rank  |
+------------+---------+-------+--+
| OpEd       | 327687  | 1     |
| Business   | 60124   | 2     |
| National   | 54809   | 3     |
| Editorial  | 42964   | 4     |
| Foreign    | 38255   | 5     |
| Learning   | 17208   | 6     |
| Magazine   | 16374   | 7     |
+------------+---------+-------+--+
7 rows selected (17.018 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> insert overwrite directory '/user/tgawade/RESULT13.csv'
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> row format delimited fields terminated by ',' SELECT newDesk,count(recommendations),rank() over (order by count(recommend
ations)desc) AS rank from commentyear2017 where newDesk LIKE 'OpEd' OR newDesk LIKE 'National' OR newDesk LIKE 'Business' OR newDesk LIKE 'Foreign' OR newDesk LIKE 'Edi
```

Query 6: What is the degree of polarity by most positive headlines for the year 2017?

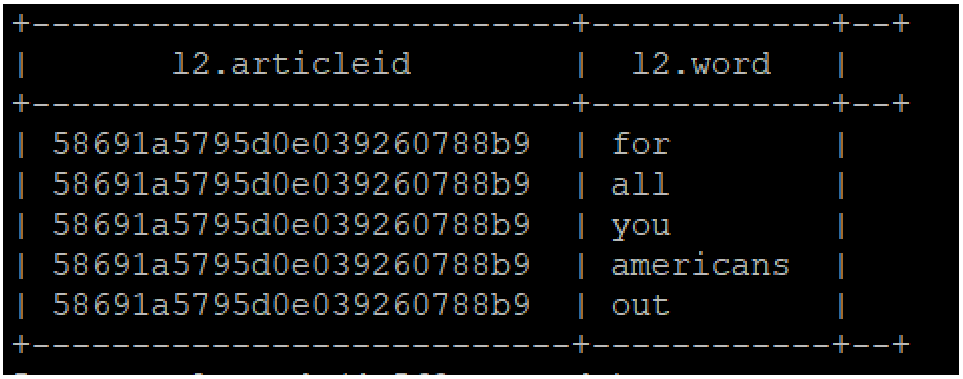
Here we created a view using the function Sentences() which splits the string present in the comment body into arrays of sentences , where each sentence is an array of words. You have your data set as arrays of words which are then lateral view exploded at the first level using the function Explode().

```
create view IF NOT EXISTS l1 as
select articleid,words
from commentyear2017
lateral view explode(sentences(lower(commentbody))) dummy as words;
```



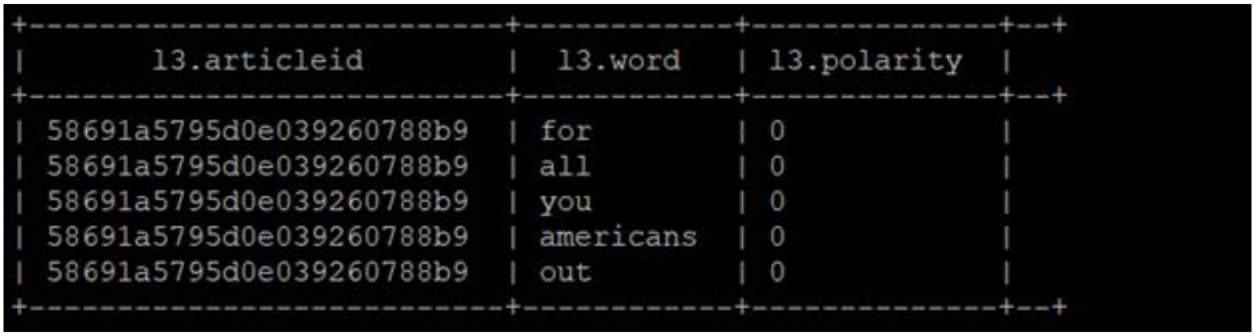
```
---+--+
|        l1.articleid        |                                                      l1.words
   |
+----------------------------+------------------------------------------------------------------------------------------------------------------------------------------
---+--+
| 58691a5795d0e039260788b9  | ["for","all","you","americans","out","there","still","rejoicing","over","the","majority","win","of","republicans","over","the","legislature","of","this","land"
|   |
| 58691a5795d0e039260788b9  | ["br","beware"]
   |
| 58691a5795d0e039260788b9  | ["br","just","like","you","would","have","been","if","there","were","any","other","kind","of","majority"]
   |
| 58691a5795d0e039260788b9  | ["br","the","founding","fathers","had","something","like","this","in","mind","whsh","they","formed","our","great","nation"]
   |
| 58691a5795d0e039260788b9  | ["br","it's","part","of","the","natural","checks","amp","balances","system","that","keeps","this","country","on","an","even","keel"]
   |
+----------------------------+------------------------------------------------------------------------------------------------------------------------------------------
```

```
create view IF NOT EXISTS l2 as
select articleid, word
from l1
lateral view explode(words) dummy as word;
```



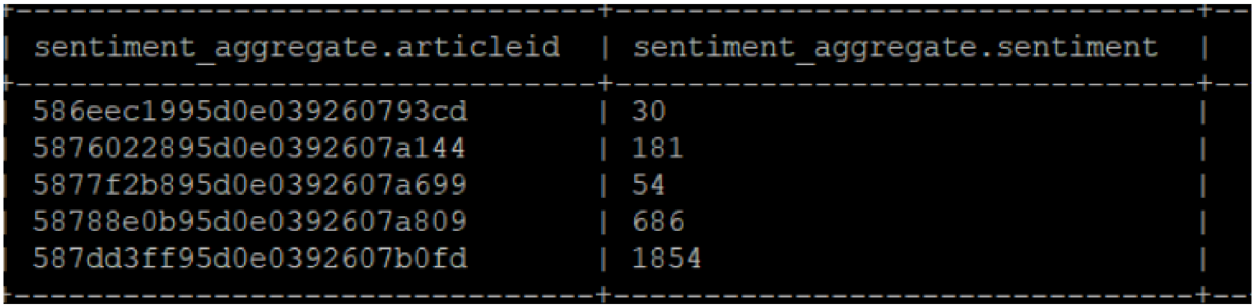
```
+---------------------------+------------+--+
|        l2.articleid       |  l2.word   |
+---------------------------+------------+--+
| 58691a5795d0e039260788b9  | for        |
| 58691a5795d0e039260788b9  | all        |
| 58691a5795d0e039260788b9  | you        |
| 58691a5795d0e039260788b9  | americans  |
| 58691a5795d0e039260788b9  | out        |
+---------------------------+------------+--+
```

```
create view IF NOT EXISTS l3 as select
articleid,
l2.word,
case d.polarity
when 'negative' then -1
when 'positive' then 1
else 0 end as polarity
from l2 left outer join dictionary d on l2.word = d.word;
```



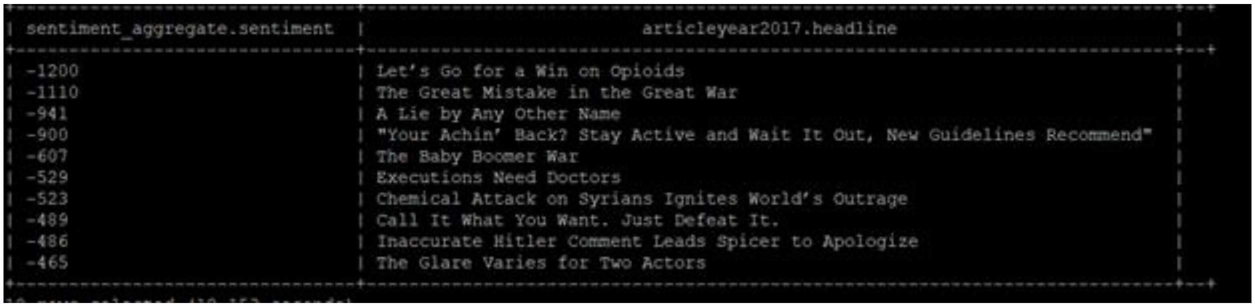
```
+-------------------------+-----------+--------------+--+
|       l3.articleid      |  l3.word  | l3.polarity  |
+-------------------------+-----------+--------------+--+
| 58691a5795d0e039260788b9  | for       | 0            |
| 58691a5795d0e039260788b9  | all       | 0            |
| 58691a5795d0e039260788b9  | you       | 0            |
| 58691a5795d0e039260788b9  | americans | 0            |
| 58691a5795d0e039260788b9  | out       | 0            |
+-------------------------+-----------+--------------+--+
```

```
create table IF NOT EXISTS sentiment_aggregate
stored as orc as select
articleid,sum( polarity ) sentiment
from l3 group by articleid;
```



```
+-----------------------------+-------------------------------+--
| sentiment_aggregate.articleid  | sentiment_aggregate.sentiment  |
+-----------------------------+-------------------------------+--
| 586eec1995d0e039260793cd       | 30                             |
| 5876022895d0e0392607a144       | 181                            |
| 5877f2b895d0e0392607a699       | 54                             |
| 58788e0b95d0e0392607a809       | 686                            |
| 587dd3ff95d0e0392607b0fd       | 1854                           |
+-----------------------------+-------------------------------+--
```

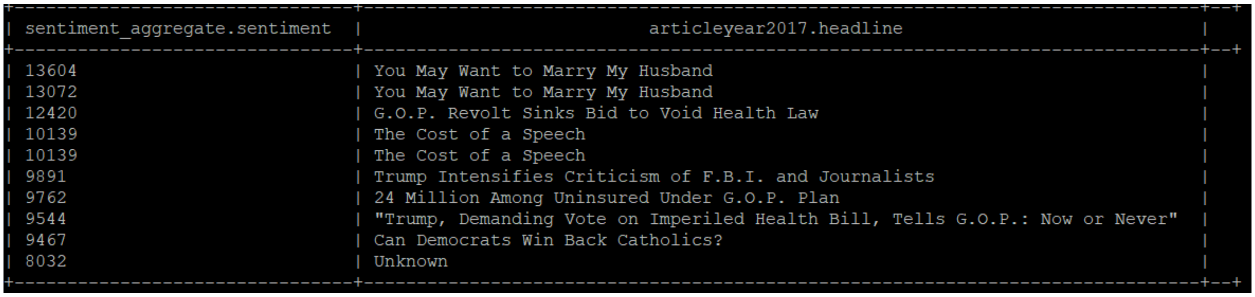
```
select sentiment_aggregate.sentiment,articleyear2017.headline from articleyear2017 inner join
sentiment_aggregate on sentiment_aggregate.articleid=articleyear2017.articleid order by
sentiment asc limit 10;
```



```
+--------------------------------+----------------------------------------------------------------------------+--+
| sentiment_aggregate.sentiment  |                          articleyear2017.headline                          |
+--------------------------------+----------------------------------------------------------------------------+--+
| -1200                          | Let’s Go for a Win on Opioids                                              |
| -1110                          | The Great Mistake in the Great War                                         |
| -941                           | A Lie by Any Other Name                                                    |
| -900                           | "Your Achin’ Back? Stay Active and Wait It Out, New Guidelines Recommend"  |
| -607                           | The Baby Boomer War                                                        |
| -529                           | Executions Need Doctors                                                    |
| -523                           | Chemical Attack on Syrians Ignites World’s Outrage                         |
| -489                           | Call It What You Want. Just Defeat It.                                     |
| -486                           | Inaccurate Hitler Comment Leads Spicer to Apologize                        |
| -465                           | The Glare Varies for Two Actors                                            |
+--------------------------------+----------------------------------------------------------------------------+--+
```

Query 7: What is the degree of polarity by most negative headlines?

```
select sentiment_aggregate.sentiment,articleyear2017.headline from articleyear2017 inner join
sentiment_aggregate on sentiment_aggregate.articleid=articleyear2017.articleid order by
sentiment desc limit 10;
```



```
+--------------------------------+-----------------------------------------------------------------------------+--+
| sentiment_aggregate.sentiment  |                          articleyear2017.headline                           |
+--------------------------------+-----------------------------------------------------------------------------+--+
| 13604                          | You May Want to Marry My Husband                                            |
| 13072                          | You May Want to Marry My Husband                                            |
| 12420                          | G.O.P. Revolt Sinks Bid to Void Health Law                                  |
| 10139                          | The Cost of a Speech                                                        |
| 10139                          | The Cost of a Speech                                                        |
| 9891                           | Trump Intensifies Criticism of F.B.I. and Journalists                       |
| 9762                           | 24 Million Among Uninsured Under G.O.P. Plan                                |
| 9544                           | "Trump, Demanding Vote on Imperiled Health Bill, Tells G.O.P.: Now or Never"  |
| 9467                           | Can Democrats Win Back Catholics?                                           |
| 8032                           | Unknown                                                                     |
+--------------------------------+-----------------------------------------------------------------------------+--+
```

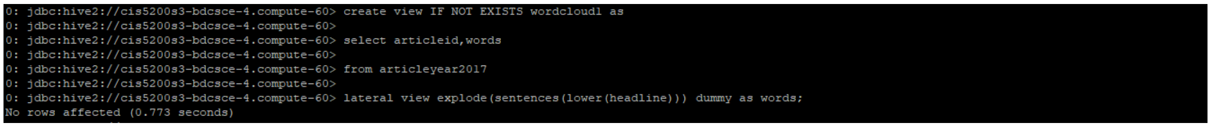
Query 8: What are the most common words in the headlines for article year 2017?

Here we created a view using the function Sentences() which splits the string present in the comment body into arrays of sentences , where each sentence is an array of words. You have your data set as arrays of words which are then lateral view exploded at the first level using the function Explode().

In this query, we are counting the most common words used in the headlines section.

```
create view IF NOT EXISTS wordcloud1 as
select articleid,words
from articleyear2017
lateral view explode(sentences(lower(headline))) dummy as words;
```

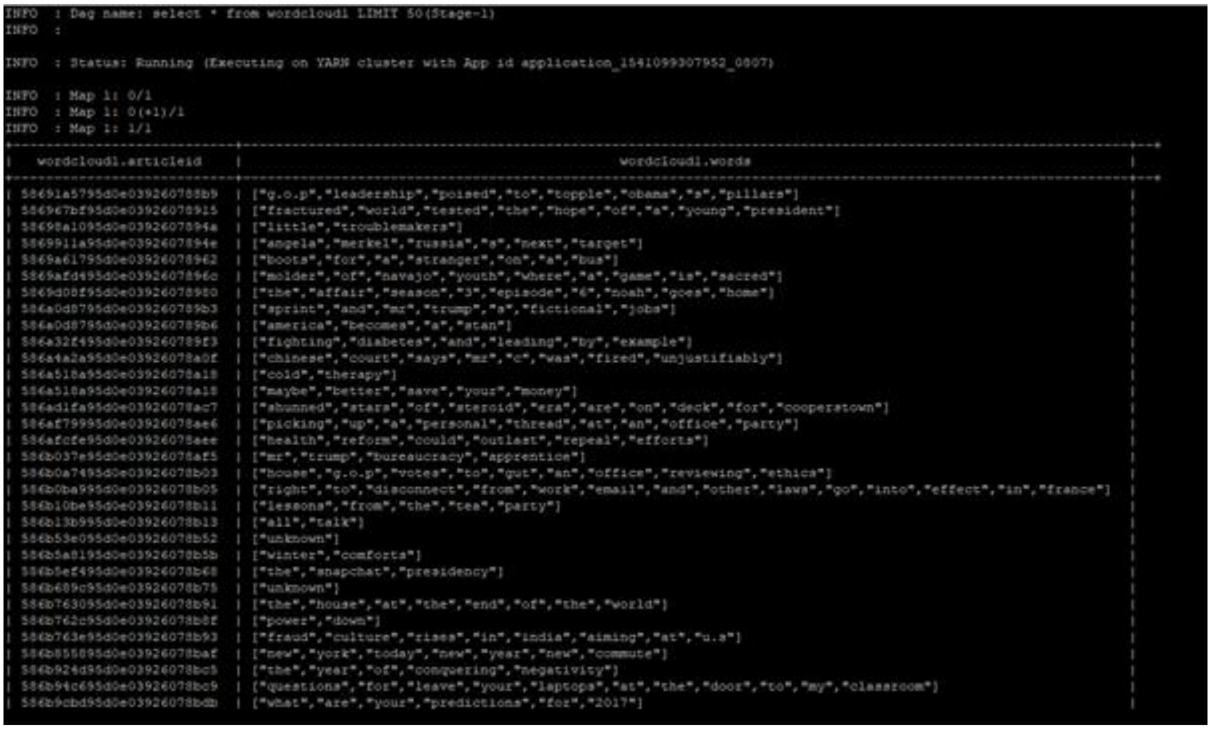
OUTPUT:



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create view IF NOT EXISTS wordcloud1 as
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select articleid,words
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> from articleyear2017
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> lateral view explode(sentences(lower(headline))) dummy as words;
No rows affected (0.773 seconds)
```

```
Select * from wordcloud1 LIMIT 50;
```

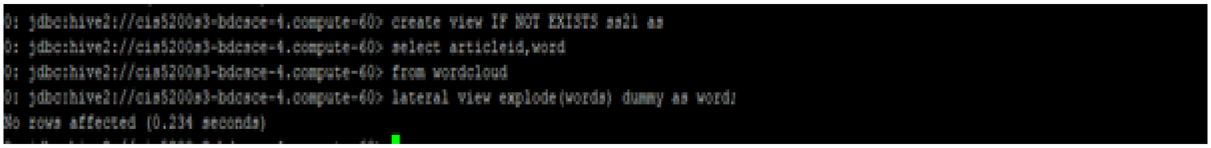
OUTPUT:



```
INFO  : Dag name: select * from wordcloud1 LIMIT 50(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0807)

INFO  : Map 1: 0/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 1/1
+---------------------------+------------------------------------------------------------------------------------------------------+--+
|   wordcloud1.articleid    |                                           wordcloud1.words                                           |
+---------------------------+------------------------------------------------------------------------------------------------------+--+
| 58691a5795d0e03926078Bb9  | ["g.o.p","leadership","poised","to","topple","obama","s","pillars"]                                  |
| 586967bf95d0e03926078915  | ["fractured","world","tested","the","hope","of","a","young","president"]                             |
| 58698a1095d0e0392607894a  | ["little","troublemakers"]                                                                           |
| 5869911a95d0e0392607894e  | ["angela","merkel","russia","s","next","target"]                                                     |
| 5869a61795d0e03926078962  | ["boots","for","a","stranger","on","a","bus"]                                                        |
| 5869afd495d0e0392607896c  | ["molder","of","navajo","youth","where","a","game","is","sacred"]                                    |
| 5869d08f95d0e03926078980  | ["the","affair","season","3","episode","6","noah","goes","home"]                                     |
| 586a0d8795d0e039260789b3  | ["sprint","and","mr","trump","s","fictional","jobs"]                                                 |
| 586a0d8795d0e039260789b6  | ["america","becomes","a","stan"]                                                                     |
| 586a32f495d0e039260789f3  | ["fighting","diabetes","and","leading","by","example"]                                               |
| 586a4a2a95d0e03926078a0f  | ["chinese","court","says","mr","c","was","fired","unjustifiably"]                                    |
| 586a518a95d0e03926078a18  | ["cold","therapy"]                                                                                   |
| 586a518a95d0e03926078a18  | ["maybe","better","save","your","money"]                                                             |
| 586ad1fa95d0e03926078ac7  | ["shunned","stars","of","steroid","era","are","on","deck","for","cooperstown"]                       |
| 586af79995d0e03926078ae6  | ["picking","up","a","personal","thread","at","an","office","party"]                                  |
| 586afcfe95d0e03926078aee  | ["health","reform","could","outlast","repeal","efforts"]                                             |
| 586b037e95d0e03926078af5  | ["mr","trump","bureaucracy","apprentice"]                                                            |
| 586b0a7495d0e03926078b03  | ["house","g.o.p","votes","to","gut","an","office","reviewing","ethics"]                              |
| 586b0ba995d0e03926078b05  | ["right","to","disconnect","from","work","email","and","other","laws","go","into","effect","in","france"] |
| 586b10be95d0e03926078b11  | ["lessons","from","the","tea","party"]                                                               |
| 586b13b995d0e03926078b13  | ["all","talk"]                                                                                       |
| 586b53e095d0e03926078b52  | ["unknown"]                                                                                          |
| 586b5a8195d0e03926078b5b  | ["winter","comforts"]                                                                                |
| 586b5ef495d0e03926078b68  | ["the","snapchat","presidency"]                                                                      |
| 586b689c95d0e03926078b75  | ["unknown"]                                                                                          |
| 586b763095d0e03926078b91  | ["the","house","at","the","end","of","the","world"]                                                  |
| 586b762c95d0e03926078b8f  | ["power","down"]                                                                                     |
| 586b763e95d0e03926078b93  | ["fraud","culture","rises","in","india","aiming","at","u.s"]                                         |
| 586b855895d0e03926078baf  | ["new","york","today","new","year","new","commute"]                                                  |
| 586b924d95d0e03926078bc5  | ["the","year","of","conquering","negativity"]                                                        |
| 586b94c695d0e03926078bc9  | ["questions","for","leave","your","laptops","at","the","door","to","my","classroom"]                 |
| 586b9cbd95d0e03926078bdb  | ["what","are","your","predictions","for","2017"]                                                     |
```

```
create view IF NOT EXISTS ss21 as
select articleid,word
from wordcloud
lateral view explode(words) dummy as word;
```

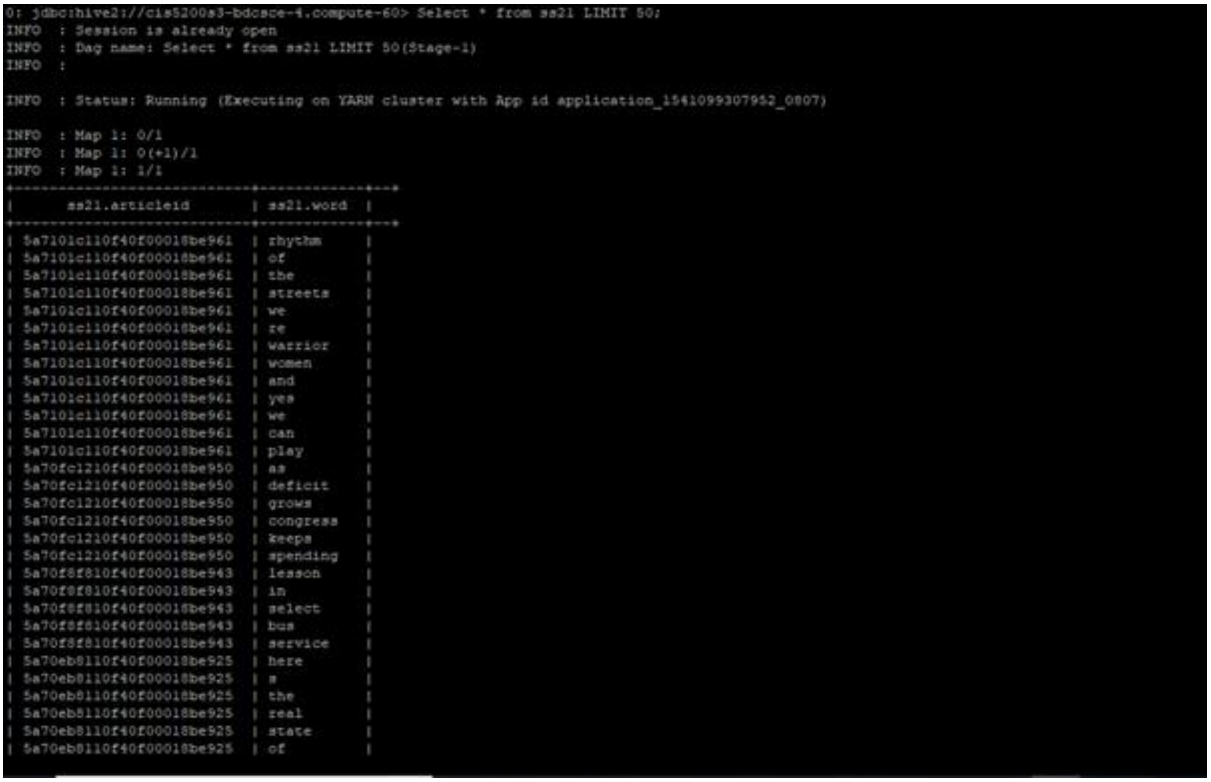
OUTPUT:



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create view IF NOT EXISTS ss21 as
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select articleid,word
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> from wordcloud
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> lateral view explode(words) dummy as word;
No rows affected (0.234 seconds)
```

```
Select * from ss21 LIMIT 50;
```

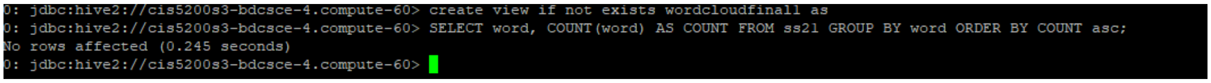
OUTPUT:



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> Select * from ss21 LIMIT 50;
INFO  : Session is already open
INFO  : Dag name: Select * from ss21 LIMIT 50(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0807)

INFO  : Map 1: 0/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 1/1
+---------------------------+-------------+--+
|      ss21.articleid       | ss21.word   |
+---------------------------+-------------+--+
| 5a7101c110f40f00018be961  | rhythm      |
| 5a7101c110f40f00018be961  | of          |
| 5a7101c110f40f00018be961  | the         |
| 5a7101c110f40f00018be961  | streets     |
| 5a7101c110f40f00018be961  | we          |
| 5a7101c110f40f00018be961  | re          |
| 5a7101c110f40f00018be961  | warrior     |
| 5a7101c110f40f00018be961  | women       |
| 5a7101c110f40f00018be961  | and         |
| 5a7101c110f40f00018be961  | yes         |
| 5a7101c110f40f00018be961  | we          |
| 5a7101c110f40f00018be961  | can         |
| 5a7101c110f40f00018be961  | play        |
| 5a70fc1210f40f00018be950  | as          |
| 5a70fc1210f40f00018be950  | deficit     |
| 5a70fc1210f40f00018be950  | grows       |
| 5a70fc1210f40f00018be950  | congress    |
| 5a70fc1210f40f00018be950  | keeps       |
| 5a70fc1210f40f00018be950  | spending    |
| 5a70f8f810f40f00018be943  | lesson      |
| 5a70f8f810f40f00018be943  | in          |
| 5a70f8f810f40f00018be943  | select      |
| 5a70f8f810f40f00018be943  | bus         |
| 5a70f8f810f40f00018be943  | service     |
| 5a70eb8110f40f00018be925  | here        |
| 5a70eb8110f40f00018be925  | s           |
| 5a70eb8110f40f00018be925  | the         |
| 5a70eb8110f40f00018be925  | real        |
| 5a70eb8110f40f00018be925  | state       |
| 5a70eb8110f40f00018be925  | of          |
```

```
create view if not exists wordcloudfinal1 as
SELECT word, COUNT(word) AS COUNT FROM ss21 GROUP BY word ORDER BY COUNT asc;
```

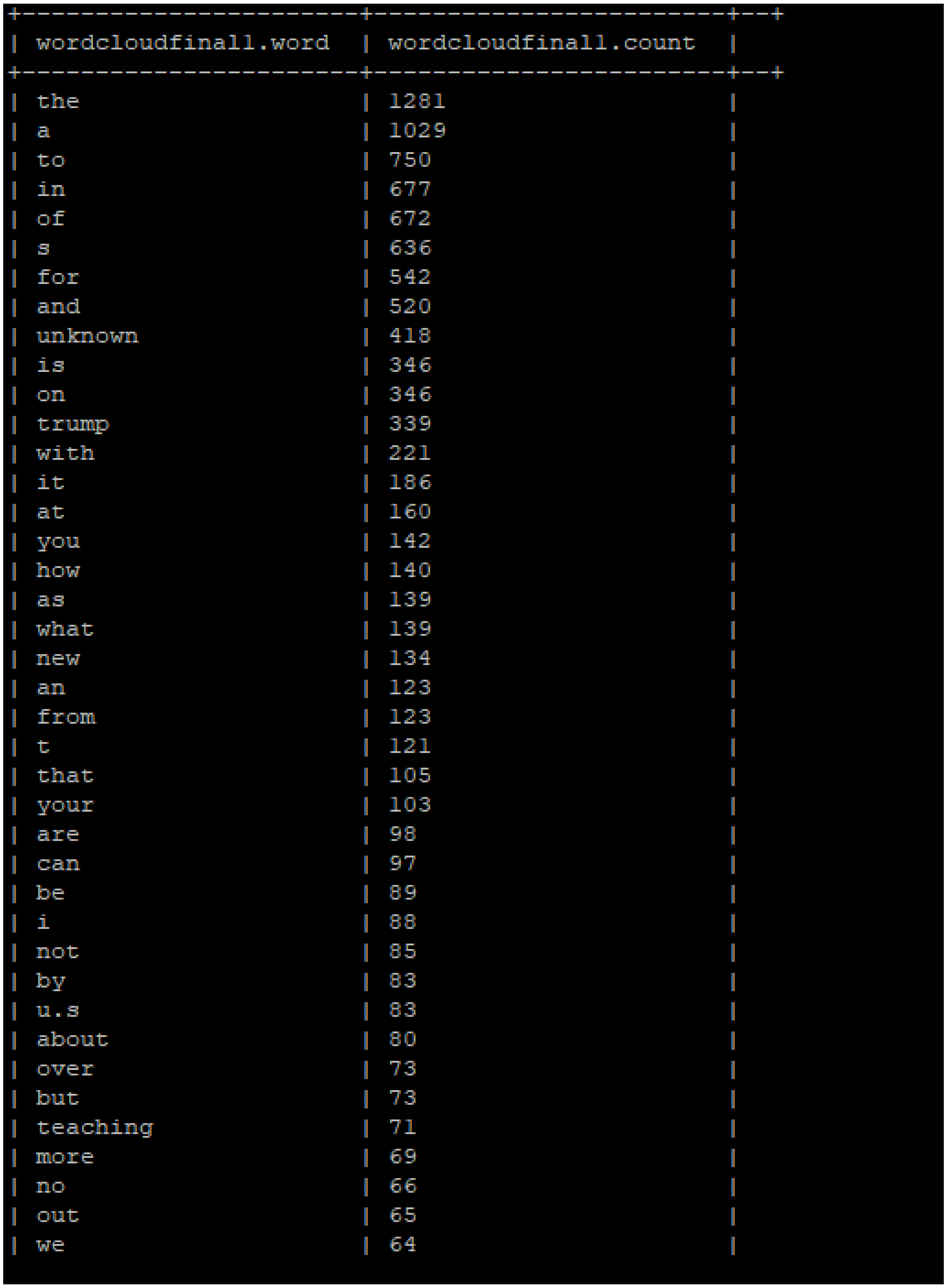
OUTPUT



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create view if not exists wordcloudfinal1 as
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT word, COUNT(word) AS COUNT FROM ss21 GROUP BY word ORDER BY COUNT asc;
No rows affected (0.245 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

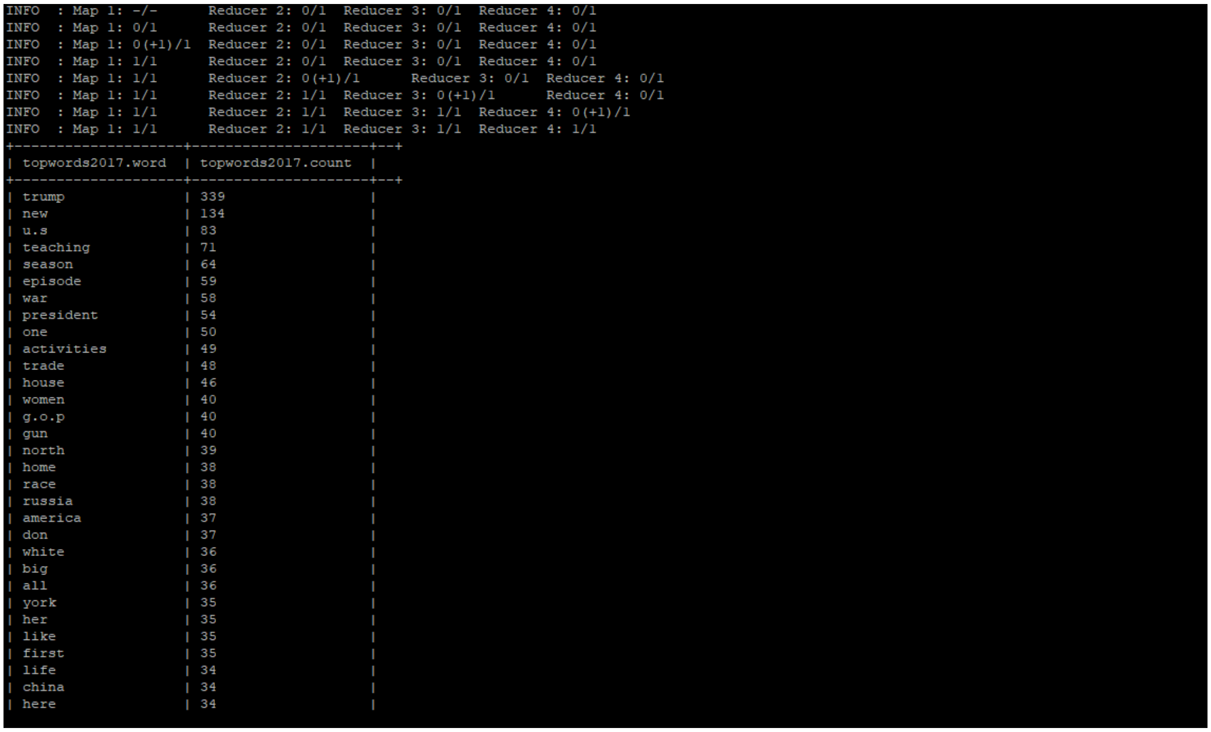
```
select * from wordcloudfinal1 order by count desc limit 100;
```

OUTPUT



```
+-----------------------+------------------------+--+
| wordcloudfinall.word  | wordcloudfinall.count  |
+-----------------------+------------------------+--+
| the                   | 1281                   |
| a                     | 1029                   |
| to                    | 750                    |
| in                    | 677                    |
| of                    | 672                    |
| s                     | 636                    |
| for                   | 542                    |
| and                   | 520                    |
| unknown               | 418                    |
| is                    | 346                    |
| on                    | 346                    |
| trump                 | 339                    |
| with                  | 221                    |
| it                    | 186                    |
| at                    | 160                    |
| you                   | 142                    |
| how                   | 140                    |
| as                    | 139                    |
| what                  | 139                    |
| new                   | 134                    |
| an                    | 123                    |
| from                  | 123                    |
| t                     | 121                    |
| that                  | 105                    |
| your                  | 103                    |
| are                   | 98                     |
| can                   | 97                     |
| be                    | 89                     |
| i                     | 88                     |
| not                   | 85                     |
| by                    | 83                     |
| u.s                   | 83                     |
| about                 | 80                     |
| over                  | 73                     |
| but                   | 73                     |
| teaching              | 71                     |
| more                  | 69                     |
| no                    | 66                     |
| out                   | 65                     |
| we                    | 64                     |
create view if not exists topwords2017 as select * from wordcloudfinal1 where not word in('the','a','to','in','of','s','for','and','unknown','is','on','from','by','i','l','t','with','it','at','you','how','as','what','an','that','your','are','can','be','not','about','but','no','out','we','over','more','now','has','who','up','this','will','do','his','he','after','may','why','when','was','into','get','its','my','or','says','should','they','have','our','1') order by count desc limit 100;
```

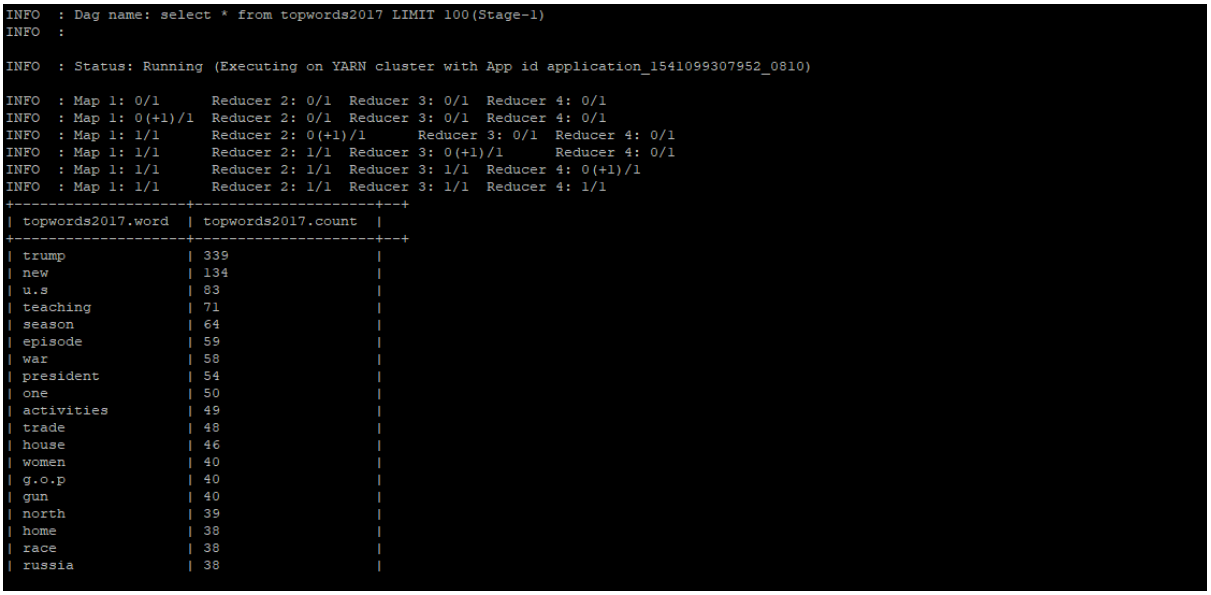
OUTPUT



```
INFO  : Map 1: -/-      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 0/1      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 0(+1)/1      Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 1/1
+--------------------+----------------------+--+
| topwords2017.word  | topwords2017.count   |
+--------------------+----------------------+--+
| trump              | 339                  |
| new                | 134                  |
| u.s                | 83                   |
| teaching           | 71                   |
| season             | 64                   |
| episode            | 59                   |
| war                | 58                   |
| president          | 54                   |
| one                | 50                   |
| activities         | 49                   |
| trade              | 48                   |
| house              | 46                   |
| women              | 40                   |
| g.o.p              | 40                   |
| gun                | 40                   |
| north              | 39                   |
| home               | 38                   |
| race               | 38                   |
| russia             | 38                   |
| america            | 37                   |
| don                | 37                   |
| white              | 36                   |
| big                | 36                   |
| all                | 36                   |
| york               | 35                   |
| her                | 35                   |
| like               | 35                   |
| first              | 35                   |
| life               | 34                   |
| china              | 34                   |
| here               | 34                   |
```

```
Select * from topwords2017 LIMIT 100;
```

OUTPUT

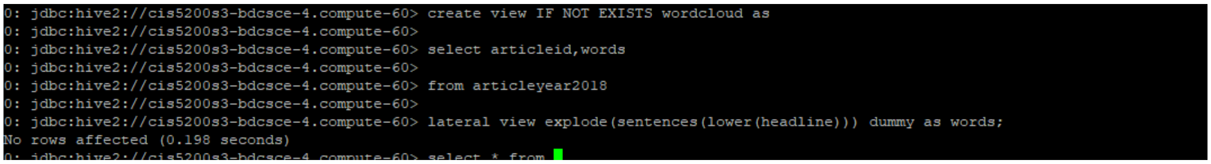


```
INFO  : Dag name: select * from topwords2017 LIMIT 100(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0810)

INFO  : Map 1: 0/1      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 0(+1)/1      Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 1/1
+---------------------+----------------------+--+
| topwords2017.word  | topwords2017.count  |
+---------------------+----------------------+--+
| trump              | 339                 |
| new                | 134                 |
| u.s                | 83                  |
| teaching           | 71                  |
| season             | 64                  |
| episode            | 59                  |
| war                | 58                  |
| president          | 54                  |
| one                | 50                  |
| activities         | 49                  |
| trade              | 48                  |
| house              | 46                  |
| women              | 40                  |
| g.o.p              | 40                  |
| gun                | 40                  |
| north              | 39                  |
| home               | 38                  |
| race               | 38                  |
| russia             | 38                  |
```

Query 9: What are the most common words in the headlines for article year 2018?

```
create view IF NOT EXISTS wordcloud as
select articleid,words
from articleyear2018
lateral view explode(sentences(lower(headline))) dummy as words;
```

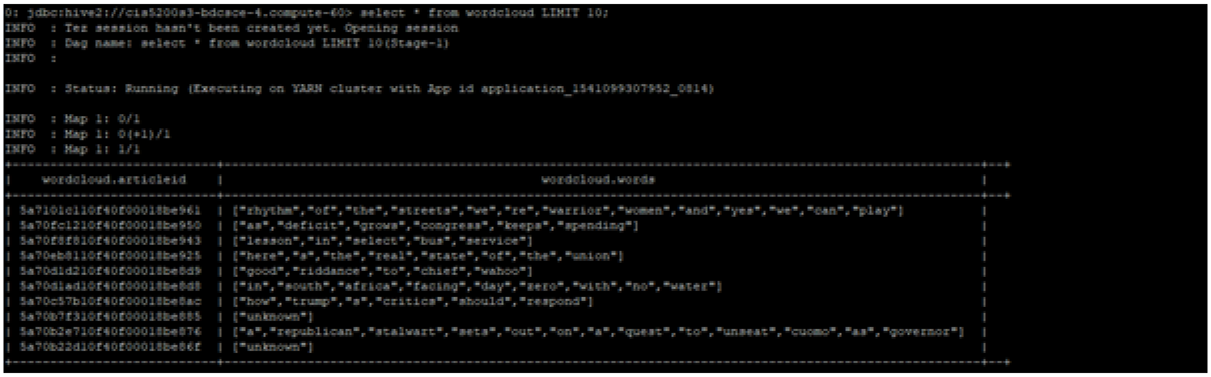
OUTPUT



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create view IF NOT EXISTS wordcloud as
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select articleid,words
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> from articleyear2018
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> lateral view explode(sentences(lower(headline))) dummy as words;
No rows affected (0.198 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select * from
```

```
Select * from wordcloud LIMIT 10;
```

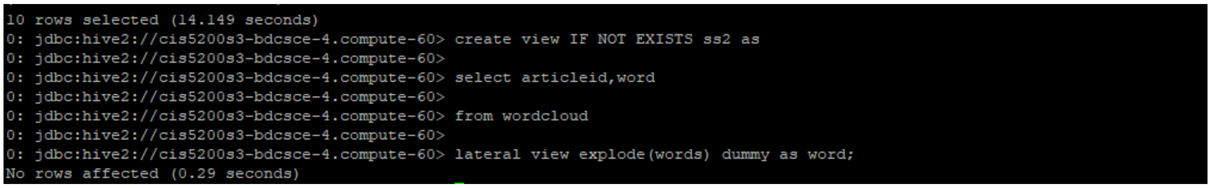
OUTPUT



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select * from wordcloud LIMIT 10;
INFO  : Tez session hasn't been created yet. Opening session
INFO  : Dag name: select * from wordcloud LIMIT 10(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0814)

INFO  : Map 1: 0/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 1/1
+---------------------------+-------------------------------------------------------------------------------------------------------------+--+
|    wordcloud.articleid    |                                               wordcloud.words                                               |
+---------------------------+-------------------------------------------------------------------------------------------------------------+--+
| 5a7101c110f40f00018be961  | ["rhythm","of","the","streets","we","re","warrior","women","and","yes","we","can","play"]                   |
| 5a70fc1210f40f00018be950  | ["as","deficit","grows","congress","keeps","spending"]                                                      |
| 5a70f8f810f40f00018be943  | ["lesson","in","select","bus","service"]                                                                    |
| 5a70eb8110f40f00018be925  | ["here","s","the","real","state","of","the","union"]                                                        |
| 5a70d1d210f40f00018be8d9  | ["good","riddance","to","chief","wahoo"]                                                                    |
| 5a70d1ad10f40f00018be8d8  | ["in","south","africa","facing","day","zero","with","no","water"]                                           |
| 5a70c57b10f40f00018be8ac  | ["how","trump","s","critics","should","respond"]                                                            |
| 5a70b7f310f40f00018be885  | ["unknown"]                                                                                                 |
| 5a70b2e710f40f00018be876  | ["a","republican","stalwart","sets","out","on","a","quest","to","unseat","cuomo","as","governor"]           |
| 5a70b22d10f40f00018be86f  | ["unknown"]                                                                                                 |
+---------------------------+-------------------------------------------------------------------------------------------------------------+--+
```

```
create view IF NOT EXISTS ss2 as
select articleid,word
from wordcloud
lateral view explode(words) dummy as word;
```

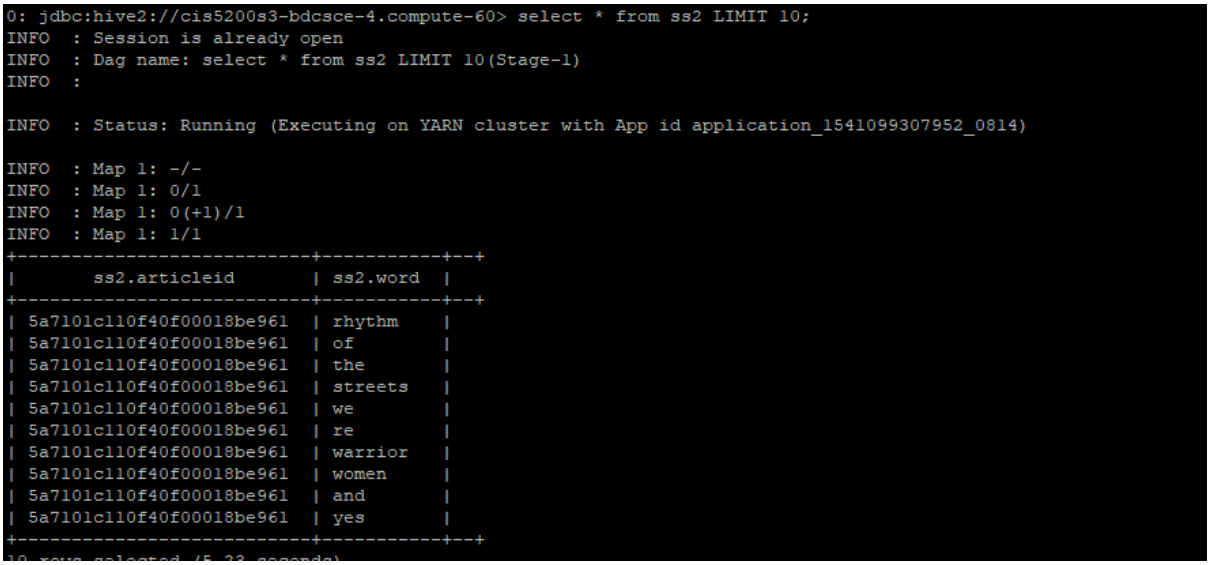
OUTPUT



```
10 rows selected (14.149 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create view IF NOT EXISTS ss2 as
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select articleid,word
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> from wordcloud
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> lateral view explode(words) dummy as word;
No rows affected (0.29 seconds)
```

```
Select * from ss2 LIMIT 10;
```

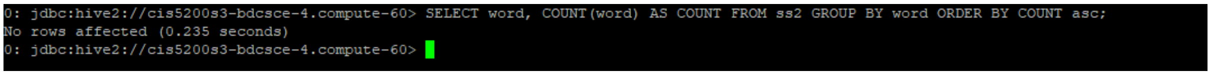
OUTPUT



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select * from ss2 LIMIT 10;
INFO  : Session is already open
INFO  : Dag name: select * from ss2 LIMIT 10(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0814)

INFO  : Map 1: -/-
INFO  : Map 1: 0/1
INFO  : Map 1: 0(+1)/1
INFO  : Map 1: 1/1
+---------------------------+-----------+--+
|       ss2.articleid       | ss2.word  |
+---------------------------+-----------+--+
| 5a7101c110f40f00018be961  | rhythm    |
| 5a7101c110f40f00018be961  | of        |
| 5a7101c110f40f00018be961  | the       |
| 5a7101c110f40f00018be961  | streets   |
| 5a7101c110f40f00018be961  | we        |
| 5a7101c110f40f00018be961  | re        |
| 5a7101c110f40f00018be961  | warrior   |
| 5a7101c110f40f00018be961  | women     |
| 5a7101c110f40f00018be961  | and       |
| 5a7101c110f40f00018be961  | yes       |
+---------------------------+-----------+--+
10 rows selected (5.23 seconds)
```

```
create view if not exists wordcloudfinal as
SELECT word, COUNT(word) AS COUNT FROM ss2 GROUP BY word ORDER BY COUNT asc;
```

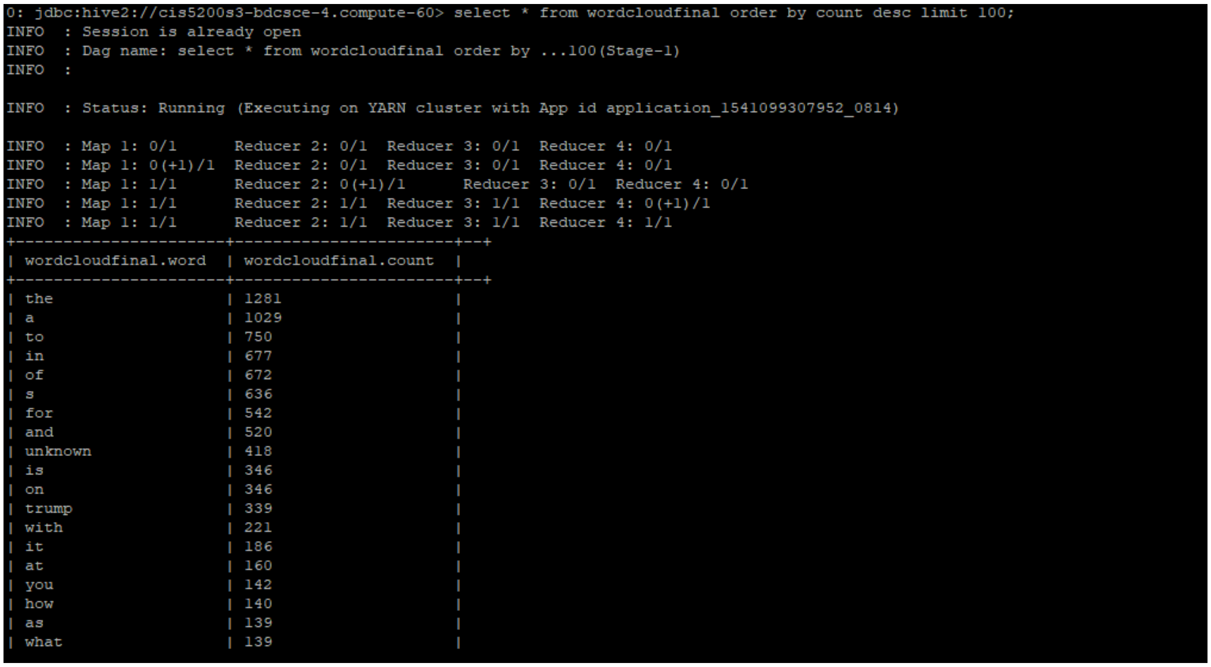
OUTPUT



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> SELECT word, COUNT(word) AS COUNT FROM ss2 GROUP BY word ORDER BY COUNT asc;
No rows affected (0.235 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

```
select * from wordcloudfinal order by count desc limit 100;
```

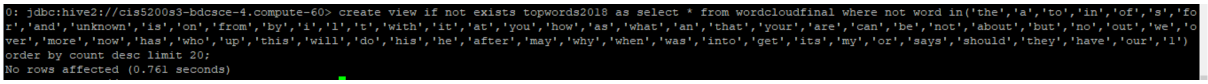
OUTPUT



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select * from wordcloudfinal order by count desc limit 100;
INFO  : Session is already open
INFO  : Dag name: select * from wordcloudfinal order by ...100(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0814)

INFO  : Map 1: 0/1      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 1/1
+----------------------+------------------------+--+
| wordcloudfinal.word  | wordcloudfinal.count   |
+----------------------+------------------------+--+
| the                  | 1281                   |
| a                    | 1029                   |
| to                   | 750                    |
| in                   | 677                    |
| of                   | 672                    |
| s                    | 636                    |
| for                  | 542                    |
| and                  | 520                    |
| unknown              | 418                    |
| is                   | 346                    |
| on                   | 346                    |
| trump                | 339                    |
| with                 | 221                    |
| it                   | 186                    |
| at                   | 160                    |
| you                  | 142                    |
| how                  | 140                    |
| as                   | 139                    |
| what                 | 139                    |
```

```
create view if not exists topwords2018 as select * from wordcloudfinal where not word in('the','a','to','in','of','s','for','and','unknown','is','on','from','by','i','l','t','with','it','at','you','how','as','what','an','that','your','are','can','be','not','about','but','no','out','we','over','more','now','has','who','up','this','will','do','his','he','after','may','why','when','was','into','get','its','my','or','says','should','they','have','our','1') order by count desc limit 20;
```

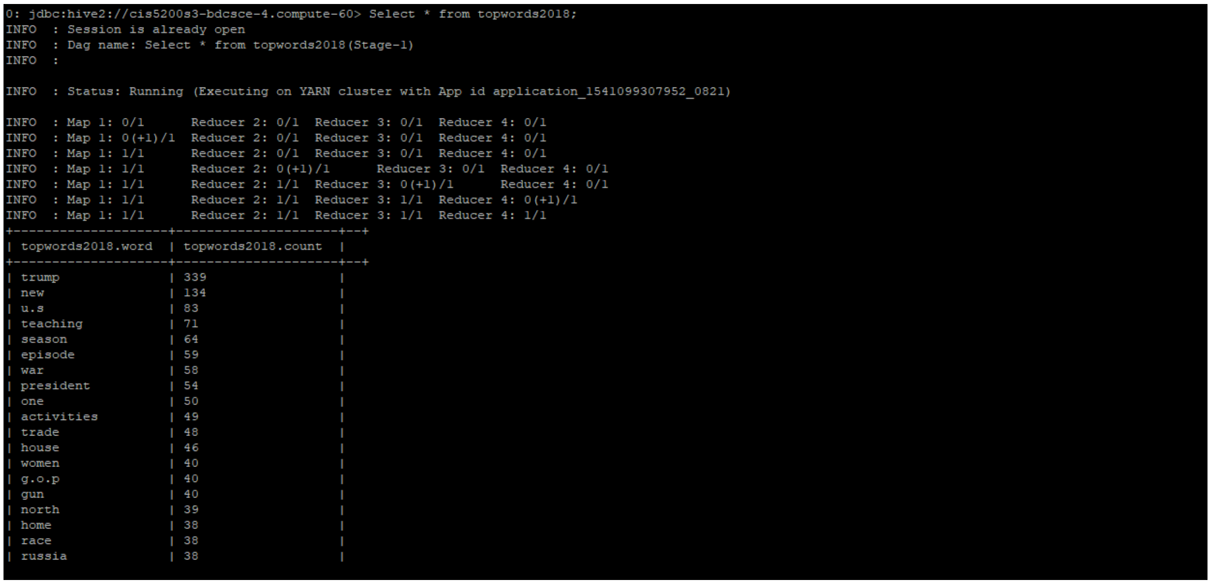
OUTPUT



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create view if not exists topwords2018 as select * from wordcloudfinal where not word in('the','a','to','in','of','s','fo
r','and','unknown','is','on','from','by','i','l','t','with','it','at','you','how','as','what','an','that','your','are','can','be','not','about','but','no','out','we','o
ver','more','now','has','who','up','this','will','do','his','he','after','may','why','when','was','into','get','its','my','or','says','should','they','have','our','1')
order by count desc limit 20;
No rows affected (0.761 seconds)
```

```
Select * from topwords2018 LIMIT 100;
```

OUTPUT



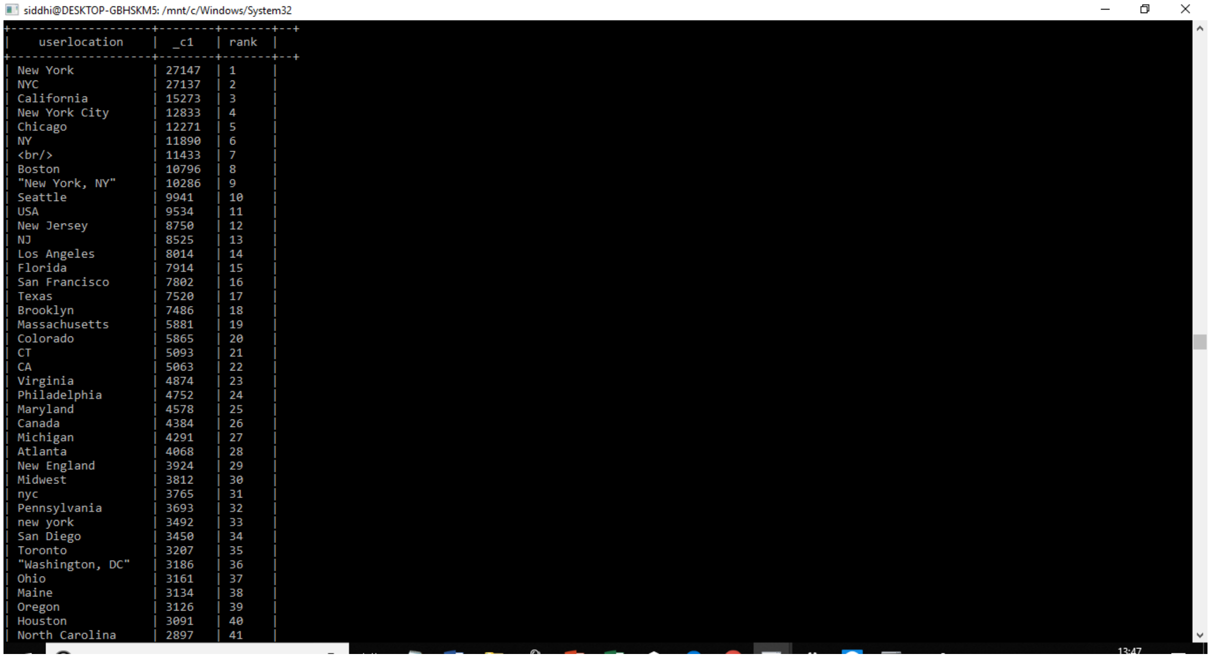
```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> Select * from topwords2018;
INFO  : Session is already open
INFO  : Dag name: Select * from topwords2018(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0821)

INFO  : Map 1: 0/1      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 0(+1)/1      Reducer 4: 0/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 1/1
+--------------------+----------------------+--+
| topwords2018.word  | topwords2018.count   |
+--------------------+----------------------+--+
| trump              | 339                  |
| new                | 134                  |
| u.s                | 83                   |
| teaching           | 71                   |
| season             | 64                   |
| episode            | 59                   |
| war                | 58                   |
| president          | 54                   |
| one                | 50                   |
| activities         | 49                   |
| trade              | 48                   |
| house              | 46                   |
| women              | 40                   |
| g.o.p              | 40                   |
| gun                | 40                   |
| north              | 39                   |
| home               | 38                   |
| race               | 38                   |
| russia             | 38                   |
```

Query 10: What are the recommendations by the user's location for the year 2017?

Recommendations are received from people and various users of NYT. These people and users could be present at different locations and so we have tried to get a count of recommendations that we receive from different locations across the U.S for both the years.

For year 2017:

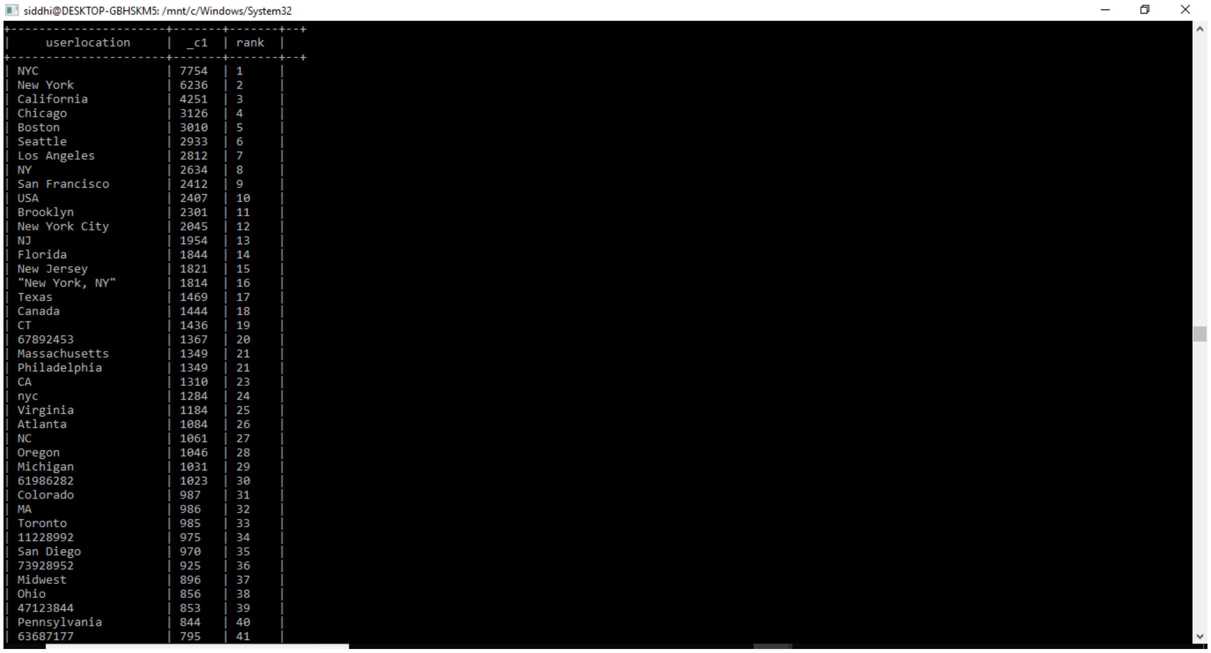
```
select userLocation, count(recommendations), rank() over (order by
count(recommendations)desc) AS
rank from commentyear2017
group by userLocation limit 100;
```



```
+---------------------+--------+-------+--+
|    userlocation     |  _c1   | rank  |
+---------------------+--------+-------+--+
| New York            | 27147  | 1     |
| NYC                 | 27137  | 2     |
| California          | 15273  | 3     |
| New York City       | 12833  | 4     |
| Chicago             | 12271  | 5     |
| NY                  | 11890  | 6     |
| <br/>               | 11433  | 7     |
| Boston              | 10796  | 8     |
| "New York, NY"      | 10286  | 9     |
| Seattle             | 9941   | 10    |
| USA                 | 9534   | 11    |
| New Jersey          | 8750   | 12    |
| NJ                  | 8525   | 13    |
| Los Angeles         | 8014   | 14    |
| Florida             | 7914   | 15    |
| San Francisco       | 7802   | 16    |
| Texas               | 7520   | 17    |
| Brooklyn            | 7486   | 18    |
| Massachusetts       | 5881   | 19    |
| Colorado            | 5865   | 20    |
| CT                  | 5093   | 21    |
| CA                  | 5063   | 22    |
| Virginia            | 4874   | 23    |
| Philadelphia        | 4752   | 24    |
| Maryland            | 4578   | 25    |
| Canada              | 4384   | 26    |
| Michigan            | 4291   | 27    |
| Atlanta             | 4068   | 28    |
| New England         | 3924   | 29    |
| Midwest             | 3812   | 30    |
| nyc                 | 3765   | 31    |
| Pennsylvania        | 3693   | 32    |
| new york            | 3492   | 33    |
| San Diego           | 3450   | 34    |
| Toronto             | 3207   | 35    |
| "Washington, DC"    | 3186   | 36    |
| Ohio                | 3161   | 37    |
| Maine               | 3134   | 38    |
| Oregon              | 3126   | 39    |
| Houston             | 3091   | 40    |
| North Carolina      | 2897   | 41    |
```

Query 11: What are the recommendations by the user’s location for the year 2018?

```
select userLocation, count(recommendations), rank() over (order by
count(recommendations)desc) AS
rank from commentyear2018
group by userLocation limit 100;
```

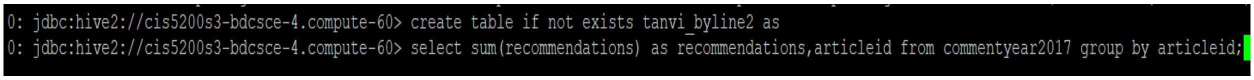


```
+----------------------+-------+-------+--+
|     userlocation     |  _c1  | rank  |
+----------------------+-------+-------+--+
| NYC                  | 7754  | 1     |
| New York             | 6236  | 2     |
| California           | 4251  | 3     |
| Chicago              | 3126  | 4     |
| Boston               | 3010  | 5     |
| Seattle              | 2933  | 6     |
| Los Angeles          | 2812  | 7     |
| NY                   | 2634  | 8     |
| San Francisco        | 2412  | 9     |
| USA                  | 2407  | 10    |
| Brooklyn             | 2301  | 11    |
| New York City        | 2045  | 12    |
| NJ                   | 1954  | 13    |
| Florida              | 1844  | 14    |
| New Jersey           | 1821  | 15    |
| "New York, NY"       | 1814  | 16    |
| Texas                | 1469  | 17    |
| Canada               | 1444  | 18    |
| CT                   | 1436  | 19    |
| 67892453             | 1367  | 20    |
| Massachusetts        | 1349  | 21    |
| Philadelphia         | 1349  | 21    |
| CA                   | 1310  | 23    |
| nyc                  | 1284  | 24    |
| Virginia             | 1184  | 25    |
| Atlanta              | 1084  | 26    |
| NC                   | 1061  | 27    |
| Oregon               | 1046  | 28    |
| Michigan             | 1031  | 29    |
| 61986282             | 1023  | 30    |
| Colorado             | 987   | 31    |
| MA                   | 986   | 32    |
| Toronto              | 985   | 33    |
| 11228992             | 975   | 34    |
| San Diego            | 970   | 35    |
| 73928952             | 925   | 36    |
| Midwest              | 896   | 37    |
| Ohio                 | 856   | 38    |
| 47123844             | 853   | 39    |
| Pennsylvania         | 844   | 40    |
| 63687177             | 795   | 41    |
```

Query 12: Most Popular Author(byline) with respect to recommendations of public for the year 2017 ?

First, we find out the sum of recommendations as per each unique articleid with help of following query

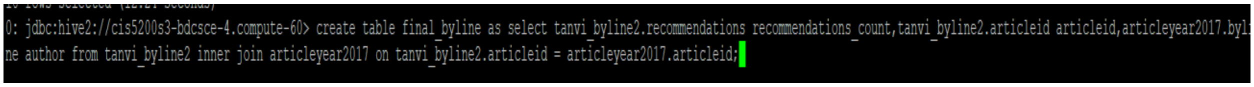
```
create table if not exists tanvi_byline2 as select sum(recommendations) as
recommendations,articleid from commentyear2017 group by articleid;
```



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create table if not exists tanvi_byline2 as
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select sum(recommendations) as recommendations,articleid from commentyear2017 group by articleid;
```

We create a new table which will store the results of the above output as well map the author with the help of inner join.

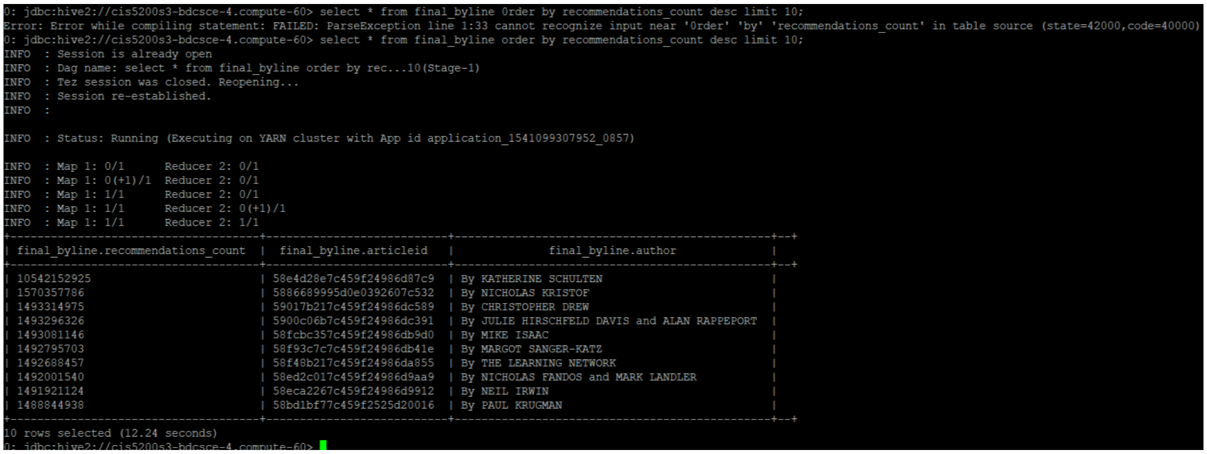
```
create table final_byline as select tanvi_byline2.recommendations
recommendations_count,tanvi_byline2.articleid articleid,articleyear2017.byline author from
tanvi_byline2 inner join articleyear2017 on tanvi_byline2.articleid = articleyear2017.articleid;
```



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create table final_byline as select tanvi_byline2.recommendations recommendations_count,tanvi_byline2.articleid articleid,articleyear2017.byli
ne author from tanvi_byline2 inner join articleyear2017 on tanvi_byline2.articleid = articleyear2017.articleid;
```

Lastly, we find out the most popular author - byline with help of below query

```
select * from final_byline order by recommendations_count desc limit 10;
```

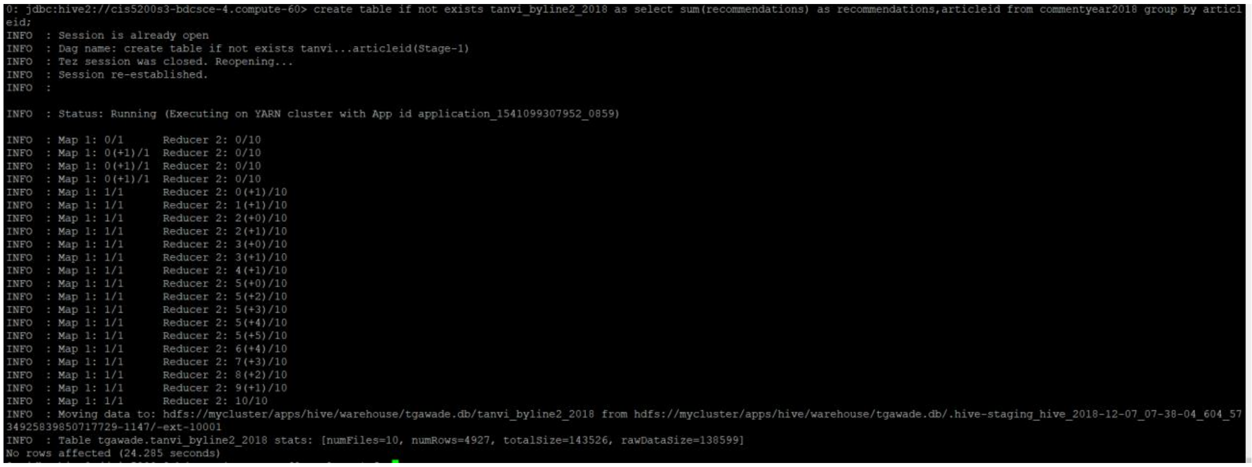


```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select * from final_byline 0rder by recommendations_count desc limit 10;
Error: Error while compiling statement: FAILED: ParseException line 1:33 cannot recognize input near '0rder' 'by' 'recommendations_count' in table source (state=42000,code=40000)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> select * from final_byline order by recommendations_count desc limit 10;
INFO  : Session is already open
INFO  : Dag name: select * from final_byline order by rec...10(Stage-1)
INFO  : Tez session was closed. Reopening...
INFO  : Session re-established.
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0857)

INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1
+-------------------------------------+---------------------------+-----------------------------------------------+--+
| final_byline.recommendations_count  |  final_byline.articleid   |              final_byline.author              |
+-------------------------------------+---------------------------+-----------------------------------------------+--+
| 10542152925                         | 58e4d28e7c459f24986d87c9  | By KATHERINE SCHULTEN                         |
| 1570357786                          | 5886689995d0e0392607c532  | By NICHOLAS KRISTOF                           |
| 1493314975                          | 59017b217c459f24986dc589  | By CHRISTOPHER DREW                           |
| 1493296326                          | 5900c06b7c459f24986dc391  | By JULIE HIRSCHFELD DAVIS and ALAN RAPPEPORT  |
| 1493081146                          | 58fcbc357c459f24986db9d0  | By MIKE ISAAC                                 |
| 1492795703                          | 58f93c7c7c459f24986db41e  | By MARGOT SANGER-KATZ                         |
| 1492688457                          | 58f48b217c459f24986da855  | By THE LEARNING NETWORK                       |
| 1492001540                          | 58ed2c017c459f24986d9aa9  | By NICHOLAS FANDOS and MARK LANDLER           |
| 1491921124                          | 58eca2267c459f24986d9912  | By NEIL IRWIN                                 |
| 1488844938                          | 58bd1bf77c459f2525d20016  | By PAUL KRUGMAN                               |
+-------------------------------------+---------------------------+-----------------------------------------------+--+
10 rows selected (12.24 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

Query 13: Most Popular Author(byline) with respect to recommendations of public for the year 2018?

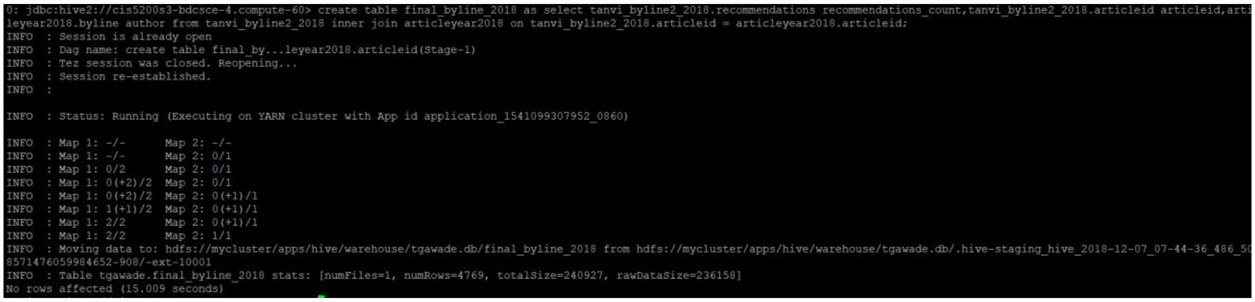
Similarly follow the above commands to find out the most popular author - byline with respect to the recommendations of public in the year 2018.

```
create table if not exists tanvi_byline2_2018 as select sum(recommendations) as
recommendations,articleid from commentyear2018 group by articleid;
```



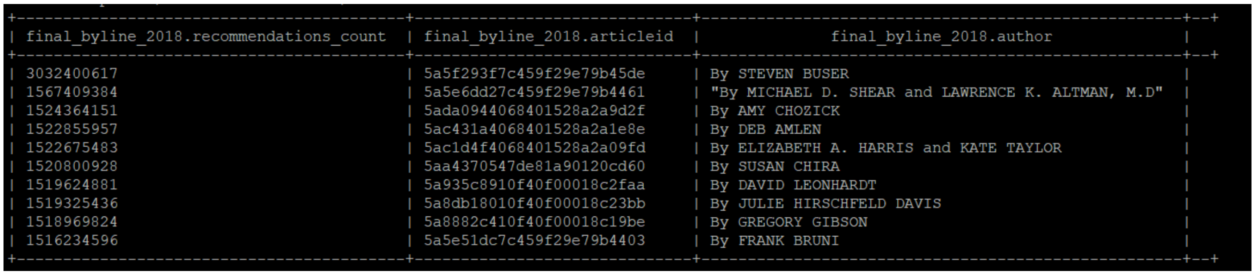
```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create table if not exists tanvi_byline2_2018 as select sum(recommendations) as recommendations,articleid from commentyear2018 group by articl
eid;
INFO  : Session is already open
INFO  : Dag name: create table if not exists tanvi...articleid(Stage-1)
INFO  : Tez session was closed. Reopening...
INFO  : Session re-established.
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0859)

INFO  : Map 1: 0/1      Reducer 2: 0/10
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/10
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/10
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/10
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/10
INFO  : Map 1: 1/1      Reducer 2: 1(+1)/10
INFO  : Map 1: 1/1      Reducer 2: 2(+0)/10
INFO  : Map 1: 1/1      Reducer 2: 2(+1)/10
INFO  : Map 1: 1/1      Reducer 2: 3(+0)/10
INFO  : Map 1: 1/1      Reducer 2: 3(+1)/10
INFO  : Map 1: 1/1      Reducer 2: 4(+1)/10
INFO  : Map 1: 1/1      Reducer 2: 5(+0)/10
INFO  : Map 1: 1/1      Reducer 2: 5(+2)/10
INFO  : Map 1: 1/1      Reducer 2: 5(+3)/10
INFO  : Map 1: 1/1      Reducer 2: 5(+4)/10
INFO  : Map 1: 1/1      Reducer 2: 5(+5)/10
INFO  : Map 1: 1/1      Reducer 2: 6(+4)/10
INFO  : Map 1: 1/1      Reducer 2: 7(+3)/10
INFO  : Map 1: 1/1      Reducer 2: 8(+2)/10
INFO  : Map 1: 1/1      Reducer 2: 9(+1)/10
INFO  : Map 1: 1/1      Reducer 2: 10/10
INFO  : Moving data to: hdfs://mycluster/apps/hive/warehouse/tgawade.db/tanvi_byline2_2018 from hdfs://mycluster/apps/hive/warehouse/tgawade.db/.hive-staging_hive_2018-12-07_07-38-04_604_57
34925839850717729-1147/-ext-10001
INFO  : Table tgawade.tanvi_byline2_2018 stats: [numFiles=10, numRows=4927, totalSize=143526, rawDataSize=138599]
No rows affected (24.285 seconds)
```

```
create table final_byline_2018 as select tanvi_byline2_2018.recommendations
recommendations_count,tanvi_byline2_2018.articleid articleid,articleyear2018.byline author from
tanvi_byline2_2018 inner join articleyear2018 on tanvi_byline2_2018.articleid =
articleyear2018.articleid;
```



```
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> create table final_byline_2018 as select tanvi_byline2_2018.recommendations recommendations_count,tanvi_byline2_2018.articleid articleid,arti
leyear2018.byline author from tanvi_byline2_2018 inner join articleyear2018 on tanvi_byline2_2018.articleid = articleyear2018.articleid;
INFO  : Session is already open
INFO  : Dag name: create table final_by...leyear2018.articleid(Stage-1)
INFO  : Tez session was closed. Reopening...
INFO  : Session re-established.
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0860)

INFO  : Map 1: -/-      Map 2: -/-
INFO  : Map 1: -/-      Map 2: 0/1
INFO  : Map 1: 0/2      Map 2: 0/1
INFO  : Map 1: 0(+2)/2  Map 2: 0/1
INFO  : Map 1: 0(+2)/2  Map 2: 0(+1)/1
INFO  : Map 1: 1(+1)/2  Map 2: 0(+1)/1
INFO  : Map 1: 2/2      Map 2: 0(+1)/1
INFO  : Map 1: 2/2      Map 2: 1/1
INFO  : Moving data to: hdfs://mycluster/apps/hive/warehouse/tgawade.db/final_byline_2018 from hdfs://mycluster/apps/hive/warehouse/tgawade.db/.hive-staging_hive_2018-12-07_07-44-36_486_50
8571476059984652-908/-ext-10001
INFO  : Table tgawade.final_byline_2018 stats: [numFiles=1, numRows=4769, totalSize=240927, rawDataSize=236158]
No rows affected (15.009 seconds)
```

```
select * from final_byline order by recommendations_count desc limit 10;
```



```
+-----------------------------------------+-----------------------------+---------------------------------------------------+--+
| final_byline_2018.recommendations_count  | final_byline_2018.articleid |               final_byline_2018.author              |
+-----------------------------------------+-----------------------------+---------------------------------------------------+--+
| 3032400617                              | 5a5f293f7c459f29e79b45de    | By STEVEN BUSER                                   |
| 1567409384                              | 5a5e6dd27c459f29e79b4461    | "By MICHAEL D. SHEAR and LAWRENCE K. ALTMAN, M.D" |
| 1524364151                              | 5ada0944068401528a2a9d2f    | By AMY CHOZICK                                    |
| 1522855957                              | 5ac431a4068401528a2a1e8e    | By DEB AMLEN                                      |
| 1522675483                              | 5ac1d4f4068401528a2a09fd    | By ELIZABETH A. HARRIS and KATE TAYLOR            |
| 1520800928                              | 5aa4370547de81a90120cd60    | By SUSAN CHIRA                                    |
| 1519624881                              | 5a935c8910f40f00018c2faa    | By DAVID LEONHARDT                                |
| 1519325436                              | 5a8db18010f40f00018c23bb    | By JULIE HIRSCHFELD DAVIS                         |
| 1518969824                              | 5a8882c410f40f00018c19be    | By GREGORY GIBSON                                 |
| 1516234596                              | 5a5e51dc7c459f29e79b4403    | By FRANK BRUNI                                    |
+-----------------------------------------+-----------------------------+---------------------------------------------------+--+
```

Hence the results for the most popular author - byline with respect to recommendations of public as given above for the year 2018.

DOWNLOADING DATA (OUTPUT FILES) INTO YOUR PC

After the Hive tables are created, we can download it to our personal PC/laptop as follows:

(The following is an example to download the output file for one query, similarly all the output files for all the queries have been downloaded in the same manner)

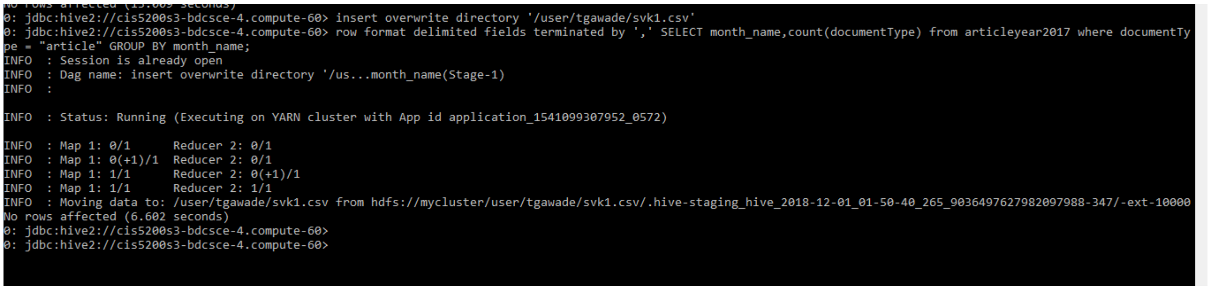
Step 1: Open another terminal Bash and connect it to Beeline which is connected to the Oracle cloud in order to download the output files and type in the following command at beeline:

```
insert overwrite directory  '/user/tgawade/svk1.csv'

row format delimited fields terminated by ',' SELECT month_name,count(documentType) from articleyear2017 where documentType = "article" GROUP BY month_name;
```

For the field marked in Green: Note: svk1.csv here is just a sample file name. you can name it anything and accordingly file with that name will be created)

For the field marked in Red: Here, which ever query you wish to run, copy and paste it here, in the field marked red above)

The following will be displayed an output on your screen:



```
No rows affected (13.009 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> insert overwrite directory '/user/tgawade/svk1.csv'
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60> row format delimited fields terminated by ',' SELECT month_name,count(documentType) from articleyear2017 where documentTy
pe = "article" GROUP BY month_name;
INFO  : Session is already open
INFO  : Dag name: insert overwrite directory '/us...month_name(Stage-1)
INFO  :

INFO  : Status: Running (Executing on YARN cluster with App id application_1541099307952_0572)

INFO  : Map 1: 0/1      Reducer 2: 0/1
INFO  : Map 1: 0(+1)/1  Reducer 2: 0/1
INFO  : Map 1: 1/1      Reducer 2: 0(+1)/1
INFO  : Map 1: 1/1      Reducer 2: 1/1
INFO  : Moving data to: /user/tgawade/svk1.csv from hdfs://mycluster/user/tgawade/svk1.csv/.hive-staging_hive_2018-12-01_01-50-40_265_9036497627982097988-347/-ext-10000
No rows affected (6.602 seconds)
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
0: jdbc:hive2://cis5200s3-bdcsce-4.compute-60>
```

Follow the rest steps as given below:

Step 2 : -bash-4.1$ hdfs dfs -ls /user/tgawade/svk1.csv

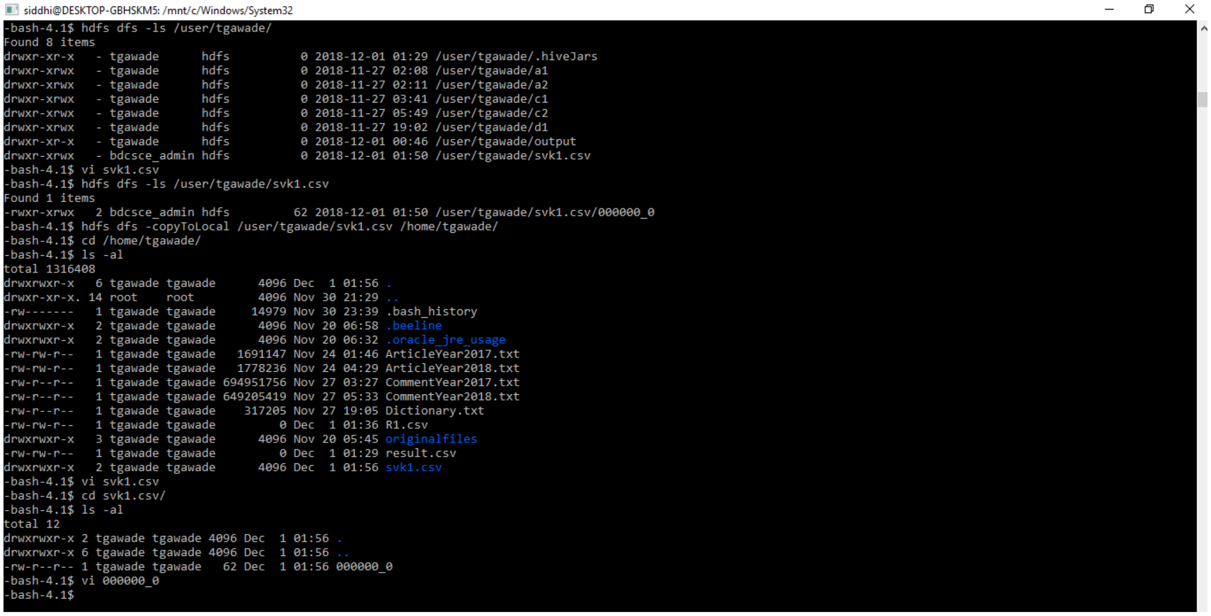
: -bash-4.1$ hdfs dfs -copyToLocal /user/tgawade/svk1.csv /home/tgawade/

: -bash-4.1$ cd /home/tgawade/

: -bash-4.1$ ls -al

: -bash-4.1$ cd svk1.csv/

: -bash-4.1$ vi 000000_0

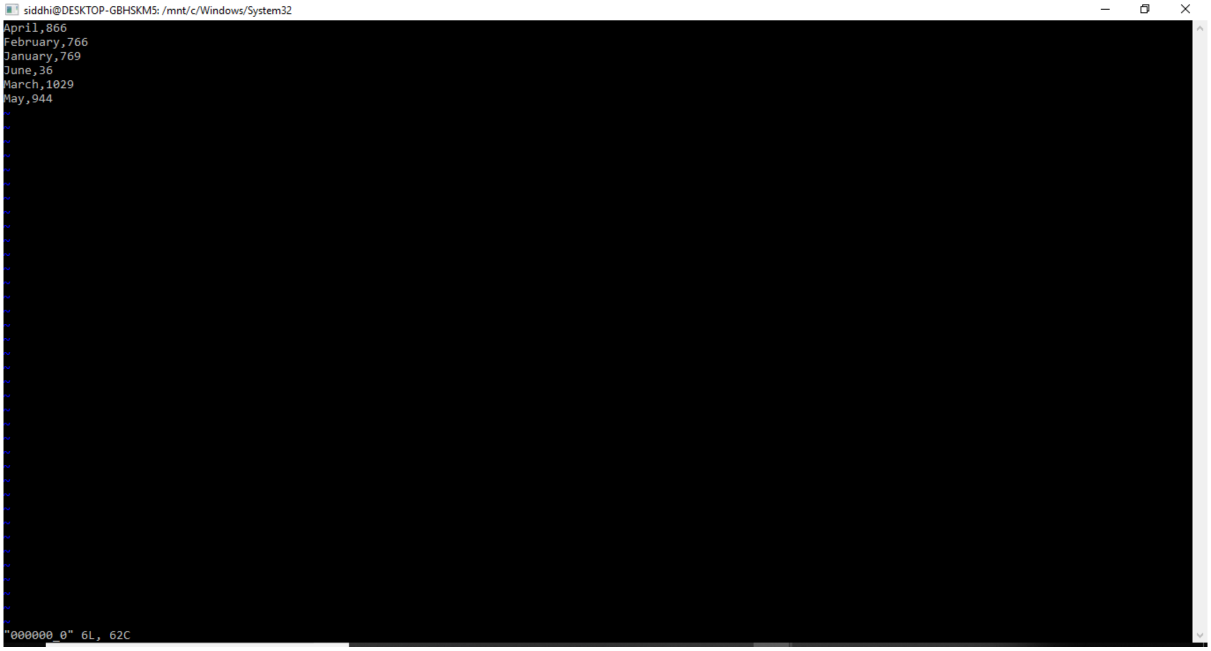


siddhi@DESKTOP-GBHSKM5: /mnt/c/Windows/System32

```
-bash-4.1$ hdfs dfs -ls /user/tgawade/
Found 8 items
drwxr-xr-x   - tgawade      hdfs          0 2018-12-01 01:29 /user/tgawade/.hiveJars
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 02:08 /user/tgawade/a1
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 02:11 /user/tgawade/a2
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 03:41 /user/tgawade/c1
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 05:49 /user/tgawade/c2
drwxr-xrwx   - tgawade      hdfs          0 2018-11-27 19:02 /user/tgawade/d1
drwxr-xr-x   - tgawade      hdfs          0 2018-12-01 00:46 /user/tgawade/output
drwxr-xrwx   - bdcsce_admin hdfs          0 2018-12-01 01:50 /user/tgawade/svk1.csv
-bash-4.1$ vi svk1.csv
-bash-4.1$ hdfs dfs -ls /user/tgawade/svk1.csv
Found 1 items
-rwxr-xrwx   2 bdcsce_admin hdfs         62 2018-12-01 01:50 /user/tgawade/svk1.csv/000000_0
-bash-4.1$ hdfs dfs -copyToLocal /user/tgawade/svk1.csv /home/tgawade/
-bash-4.1$ cd /home/tgawade/
-bash-4.1$ ls -al
total 1316408
drwxrwxr-x   6 tgawade tgawade      4096 Dec  1 01:56 .
drwxr-xr-x. 14 root    root         4096 Nov 30 21:29 ..
-rw-------   1 tgawade tgawade     14979 Nov 30 23:39 .bash_history
drwxrwxr-x   2 tgawade tgawade      4096 Nov 20 06:58 .beeline
drwxrwxr-x   2 tgawade tgawade      4096 Nov 20 06:32 .oracle_jre_usage
-rw-rw-r--   1 tgawade tgawade   1691147 Nov 24 01:46 ArticleYear2017.txt
-rw-rw-r--   1 tgawade tgawade   1778236 Nov 24 04:29 ArticleYear2018.txt
-rw-r--r--   1 tgawade tgawade 694951756 Nov 27 03:27 CommentYear2017.txt
-rw-r--r--   1 tgawade tgawade 649205419 Nov 27 05:33 CommentYear2018.txt
-rw-r--r--   1 tgawade tgawade    317205 Nov 27 19:05 Dictionary.txt
-rw-rw-r--   1 tgawade tgawade         0 Dec  1 01:36 R1.csv
drwxrwxr-x   3 tgawade tgawade      4096 Nov 20 05:45 originalfiles
-rw-rw-r--   1 tgawade tgawade         0 Dec  1 01:29 result.csv
drwxrwxr-x   2 tgawade tgawade      4096 Dec  1 01:56 svk1.csv
-bash-4.1$ vi svk1.csv
-bash-4.1$ cd svk1.csv/
-bash-4.1$ ls -al
total 12
drwxrwxr-x 2 tgawade tgawade 4096 Dec  1 01:56 .
drwxrwxr-x 6 tgawade tgawade 4096 Dec  1 01:56 ..
-rw-r--r-- 1 tgawade tgawade   62 Dec  1 01:56 000000_0
-bash-4.1$ vi 000000_0
-bash-4.1$
```

When you run the last command of the screenshot, that is, “ -bash-4.1$ vi 000000_0” , the output should be the result of your query :

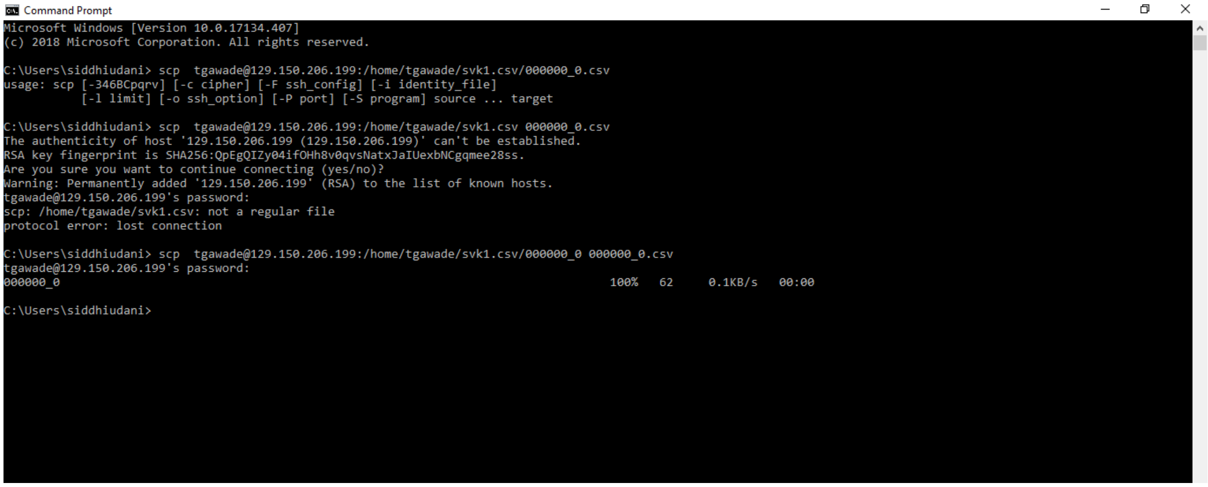
Example: For the query that I have run above, the output is as follows:



```
April,866
February,766
January,769
June,36
March,1029
May,944
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"000000_0" 6L, 62C
```

Step 3: Now open Command Prompt/Putty and run the following steps.

(We have done this using Command Prompt)



```
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\siddhiudani> scp  tgawade@129.150.206.199:/home/tgawade/svk1.csv/000000_0.csv
usage: scp [-346BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
           [-l limit] [-o ssh_option] [-P port] [-S program] source ... target

C:\Users\siddhiudani> scp  tgawade@129.150.206.199:/home/tgawade/svk1.csv 000000_0.csv
The authenticity of host '129.150.206.199 (129.150.206.199)' can't be established.
RSA key fingerprint is SHA256:QpEgQIZy04ifOHh8v0qvsNatxJaIUexbNCgqmee28ss.
Are you sure you want to continue connecting (yes/no)?
Warning: Permanently added '129.150.206.199' (RSA) to the list of known hosts.
tgawade@129.150.206.199's password:
scp: /home/tgawade/svk1.csv: not a regular file
protocol error: lost connection

C:\Users\siddhiudani> scp  tgawade@129.150.206.199:/home/tgawade/svk1.csv/000000_0 000000_0.csv
tgawade@129.150.206.199's password:
000000_0                                                                      100%   62     0.1KB/s   00:00

C:\Users\siddhiudani>
```

Step 4: After this go to your local machine and you will find the output file there.

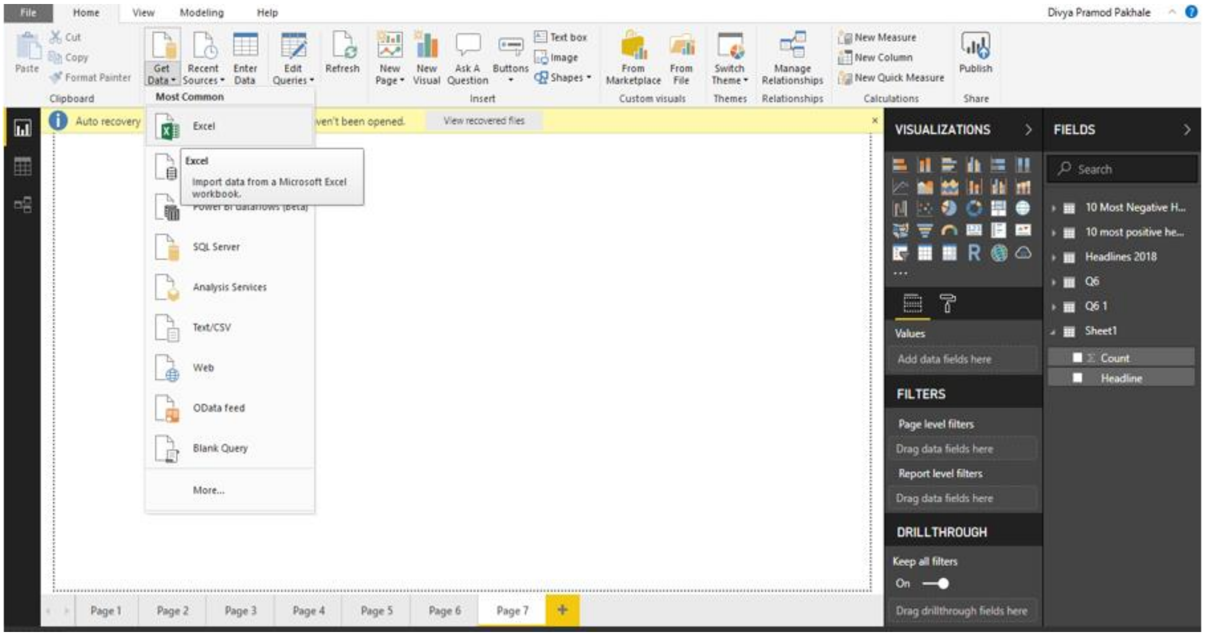
For Example, in my case it was: c drive -> users -> siddhiudani

VISUALIZATION OF DATA

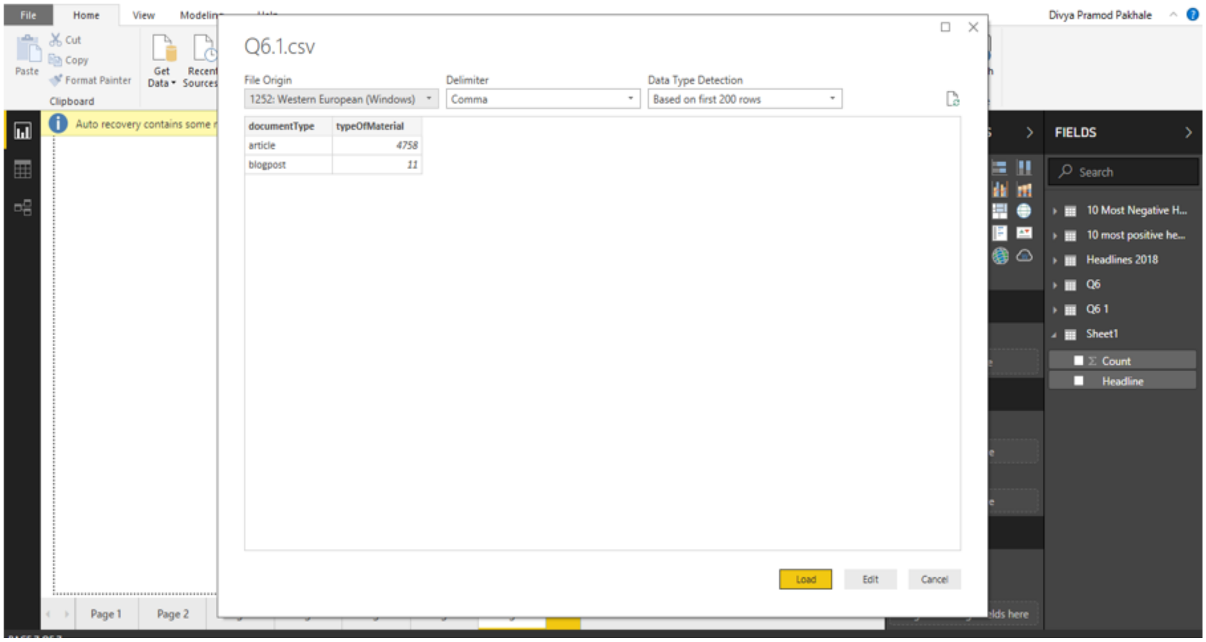
Visualizing Data: (In order to visualize the data, we have used tableau, power BI as well as Excel 3D Maps)

Query 1: Count of document type by type of material for the year 2017 and 2018.

Open the power bi tool. Click on Get data -> Excel -> Select your file (Q6.1)



Click on Load. Once the data is loaded, go to one sheet



Q6.1.csv

File Origin: 1252: Western European (Windows)

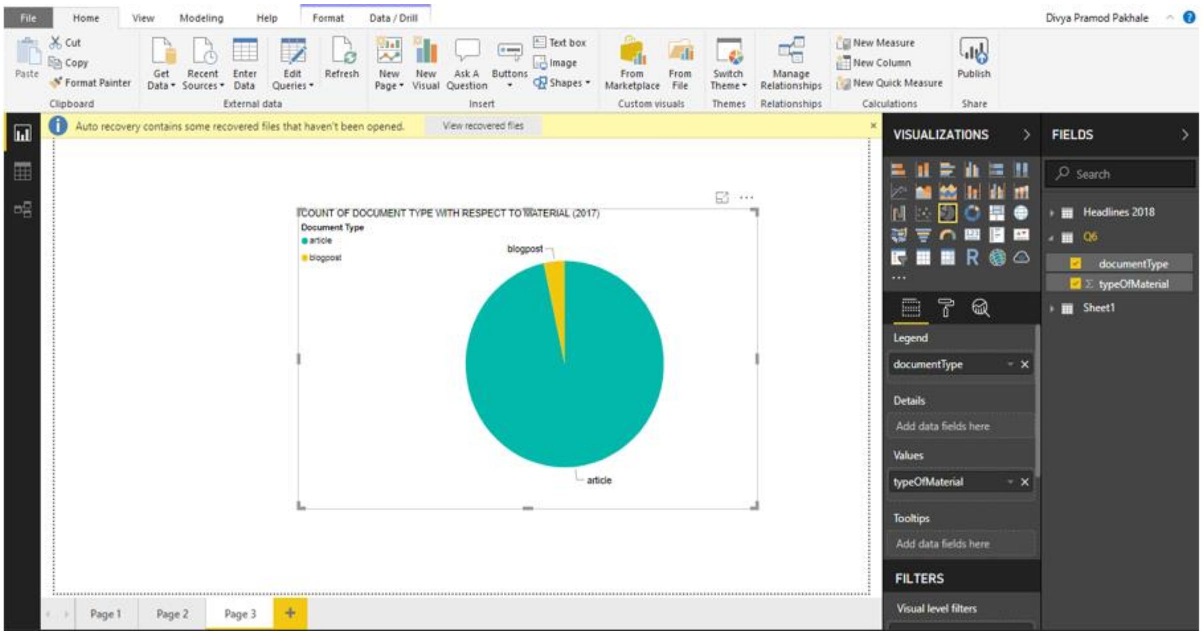
Delimiter: Comma

Data Type Detection: Based on first 200 rows

documentType	typeOfMaterial
article	4758
blogpost	11

For the Year 2017

Drag document type and type of material as shown in the picture below and select pie chart visualization from the visualization field

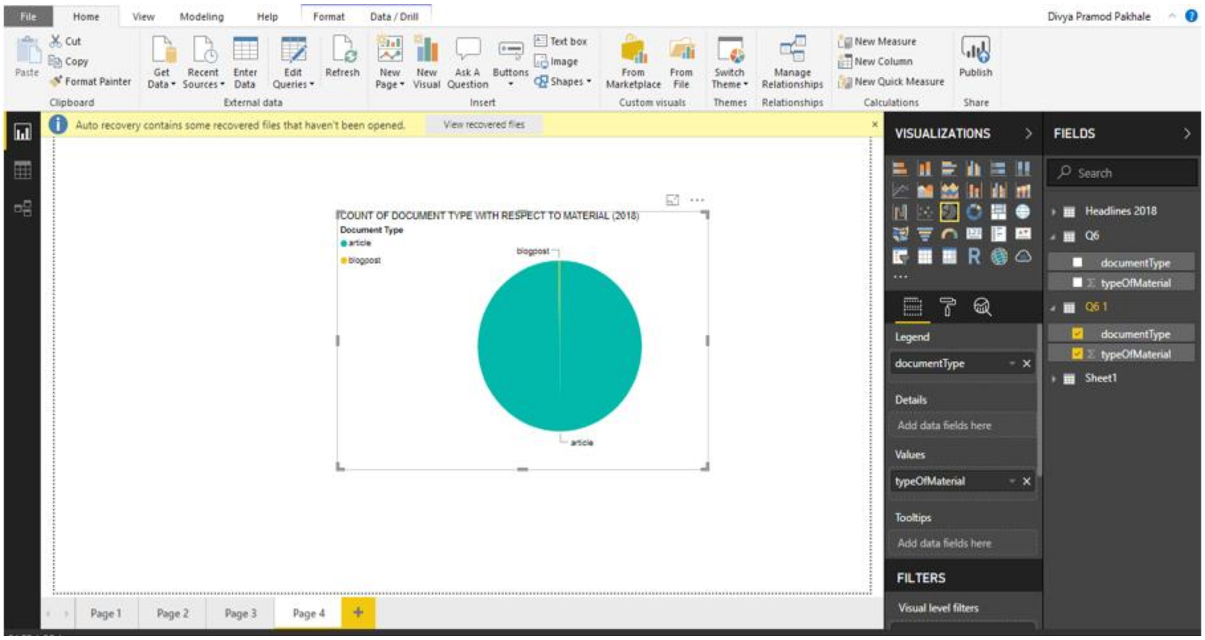


Legend: documentType

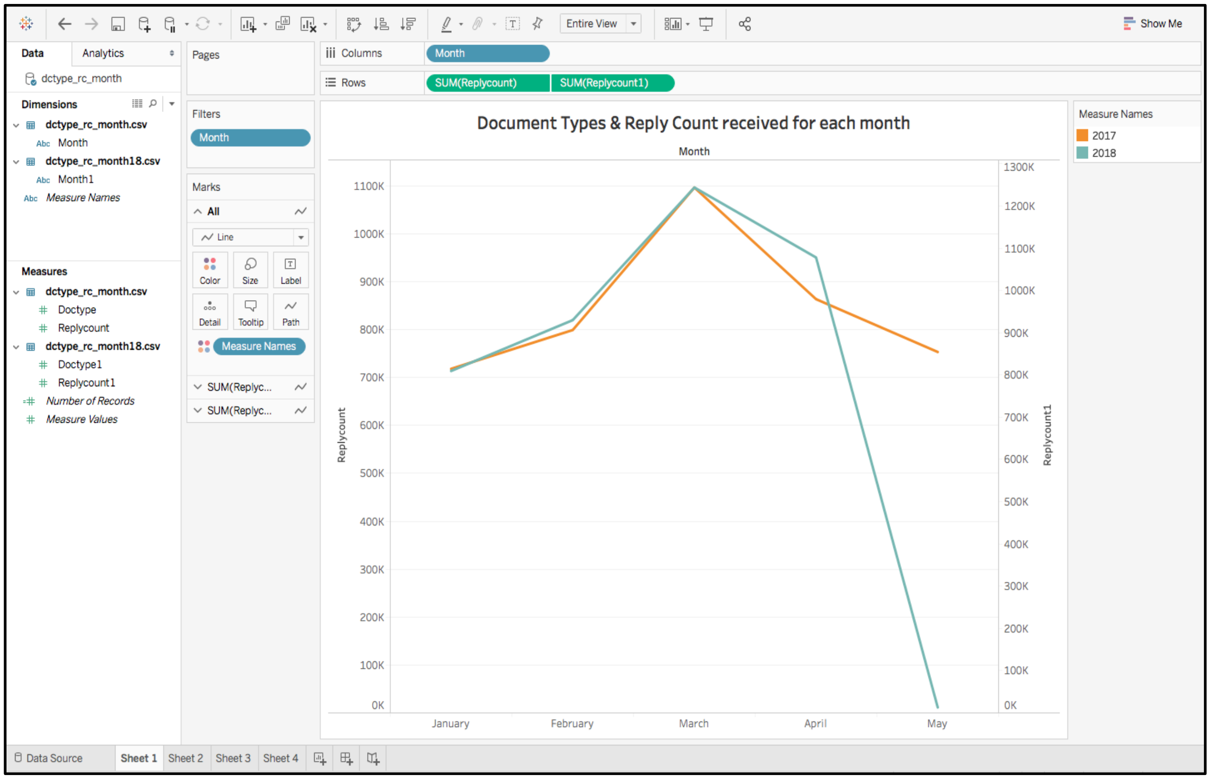
Values: typeOfMaterial

For the year 2018

Drag document type and type of material as shown in the picture below and select pie chart visualization from the visualization field



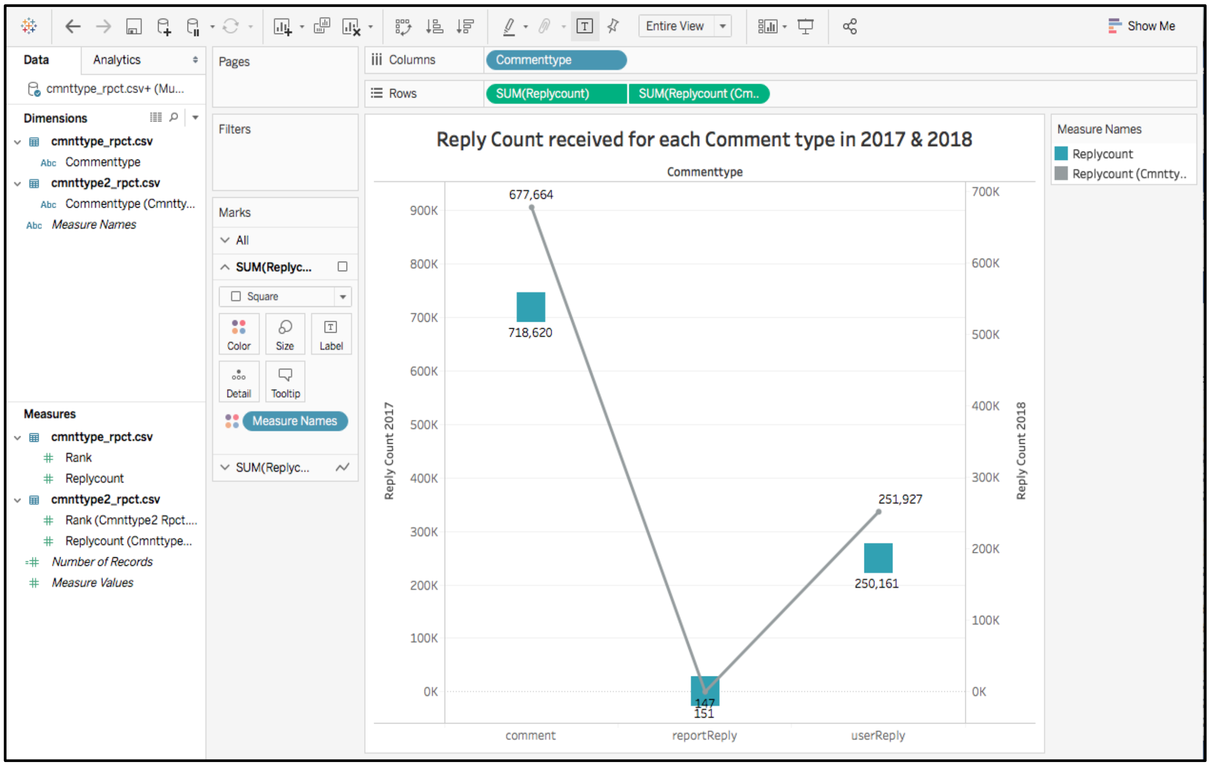
Query 2: What is the reply count for the document type month wise for Year 2017 & 2018?



Above visual shows Line Chart. Open Tableau Tool -> Upload the output file -> Drag month from dimensions to rows and reply count from dimensions to rows.

X Axis represents Month from January to May. Y Axis represents reply cunt for 2017 (Left) and 2018 (Right). Orange color represents year 2017 and blue color shows trend in year 2018

Query 3: What is the reply count for each comment type.

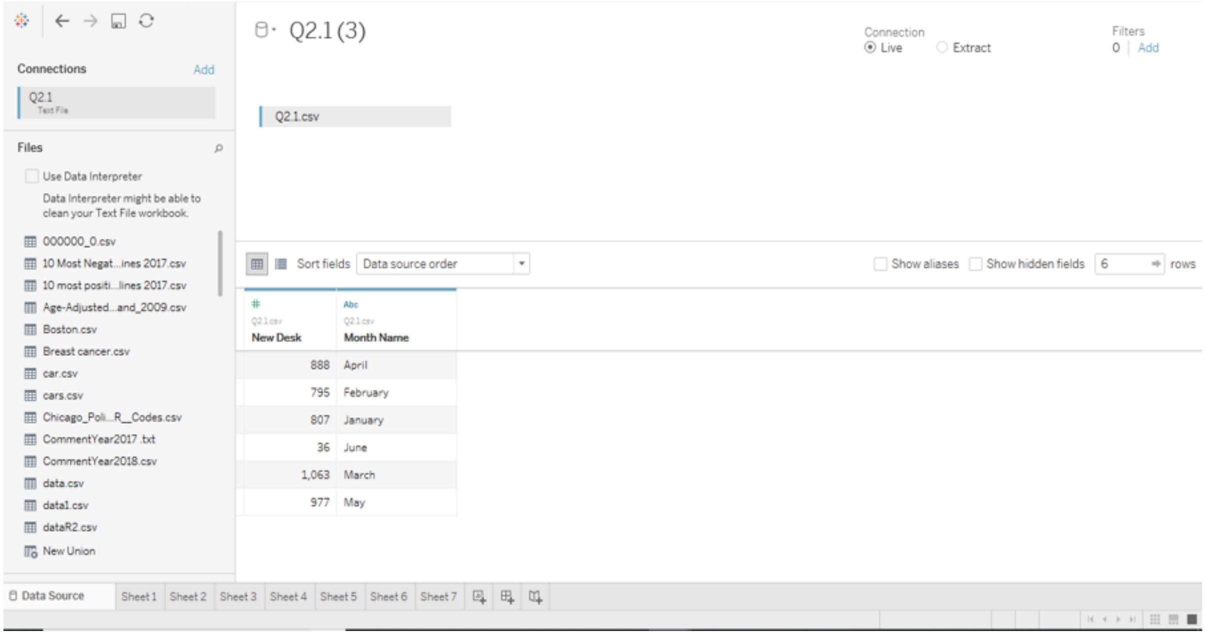


Above visual shows comparative analysis using dual axis. Open Tableau Tool -> Upload the output file -> Drag comment type from dimensions to rows and reply count for 2017 and 2018 from dimensions to rows.

X Axis represents comment type. Y Axis represents reply count for 2017 (Left) and 2018 (Right). Blue color represents year 2017 and grey color shows trend in year 2018.

Query 4: What is the count of new desk month wise.

Open the output file from “q2.1” folder in Tableau. After loading the data in Tableau, we will click of sheet 1 as shown in the picture below to create our visualization.



Q2.1 (3)

Connection ● Live ○ Extract

Filters 0 | Add

Connections Add

Q2.1
Text File

Files

□ Use Data Interpreter

Data Interpreter might be able to clean your Text File workbook.

- 000000_0.csv
- 10 Most Negat...ines 2017.csv
- 10 most positi...lines 2017.csv
- Age-Adjusted...and_2009.csv
- Boston.csv
- Breast cancer.csv
- car.csv
- cars.csv
- Chicago_Poli...R__Codes.csv
- CommentYear2017 .txt
- CommentYear2018.csv
- data.csv
- data1.csv
- dataR2.csv
- New Union

Q2.1.csv

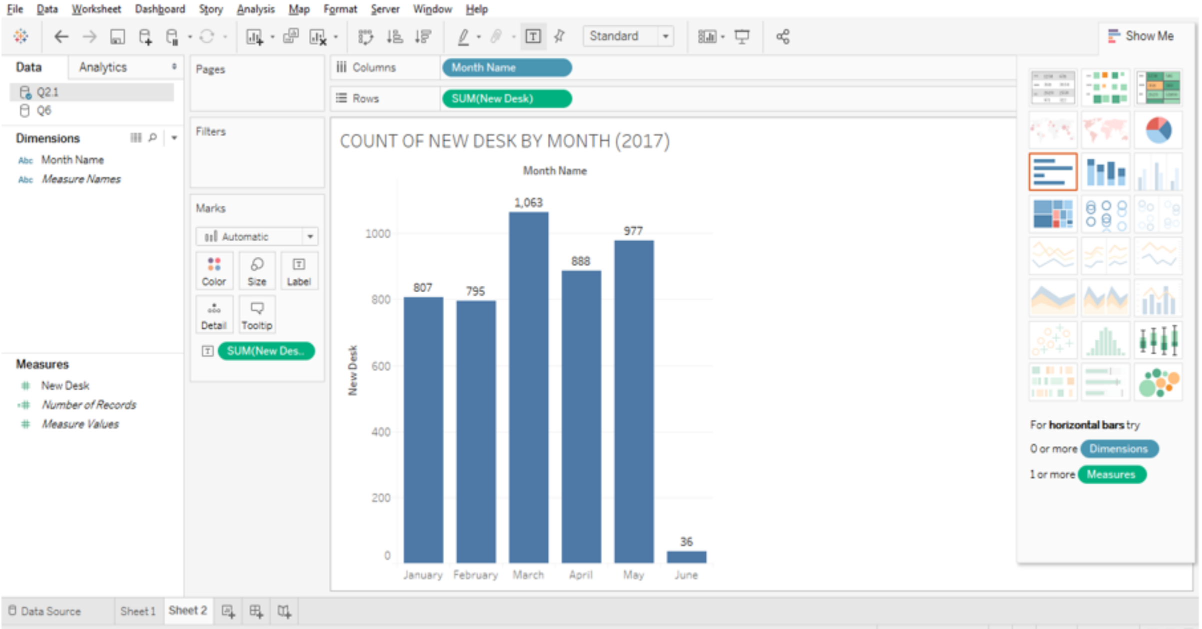
Sort fields Data source order

□ Show aliases □ Show hidden fields 6 rows

New Desk	Month Name
888	April
795	February
807	January
36	June
1,063	March
977	May

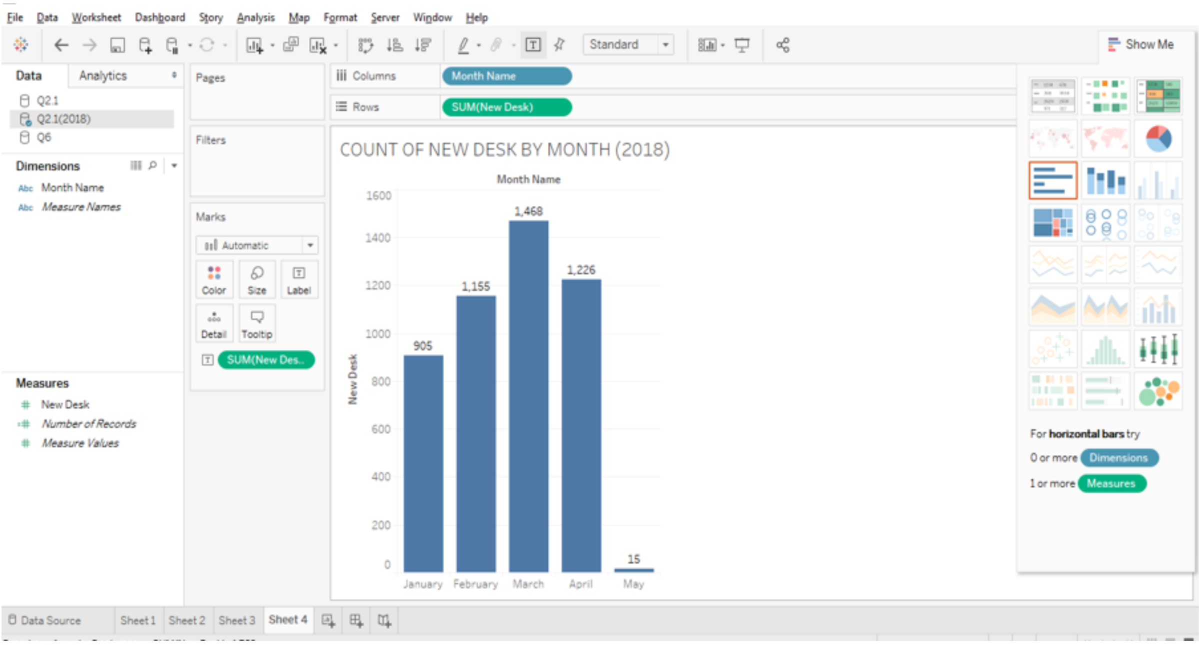
After clicking on sheet 1 worksheet, we drag the month name from dimensions to the rows field and number of desks filed from measures to the columns field.

For the year 2017



After clicking on sheet 2 worksheet, we drag the month name from dimensions to the rows field and number of desks filed from measures to the columns field.

For the year 2018



Columns: Month Name

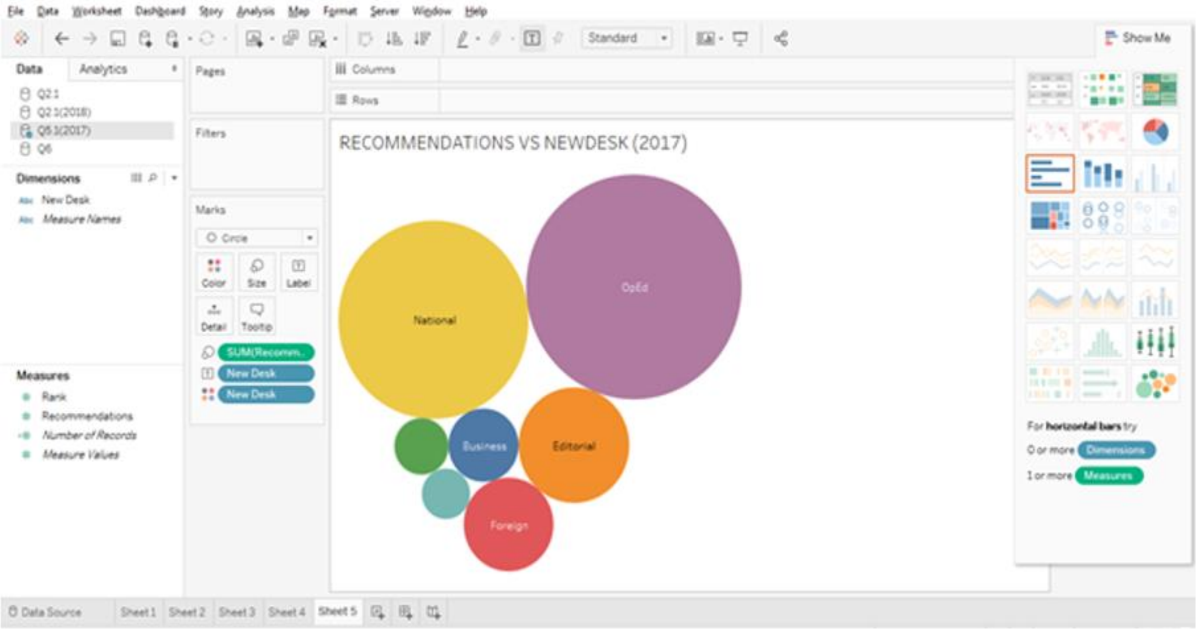
Rows: SUM(New Desk)

Query 5: What is the count of new desk based on recommendations.

Open the output file from “q5.1(2017)” folder in Tableau. After loading the data in Tableau, as shown in the picture below to create our visualization.

Drag New Desk from the dimensions field into rows and drag recommendations from measures into text in the marks field and then clicked on packed bubbles from the show me tab to get the visualization.

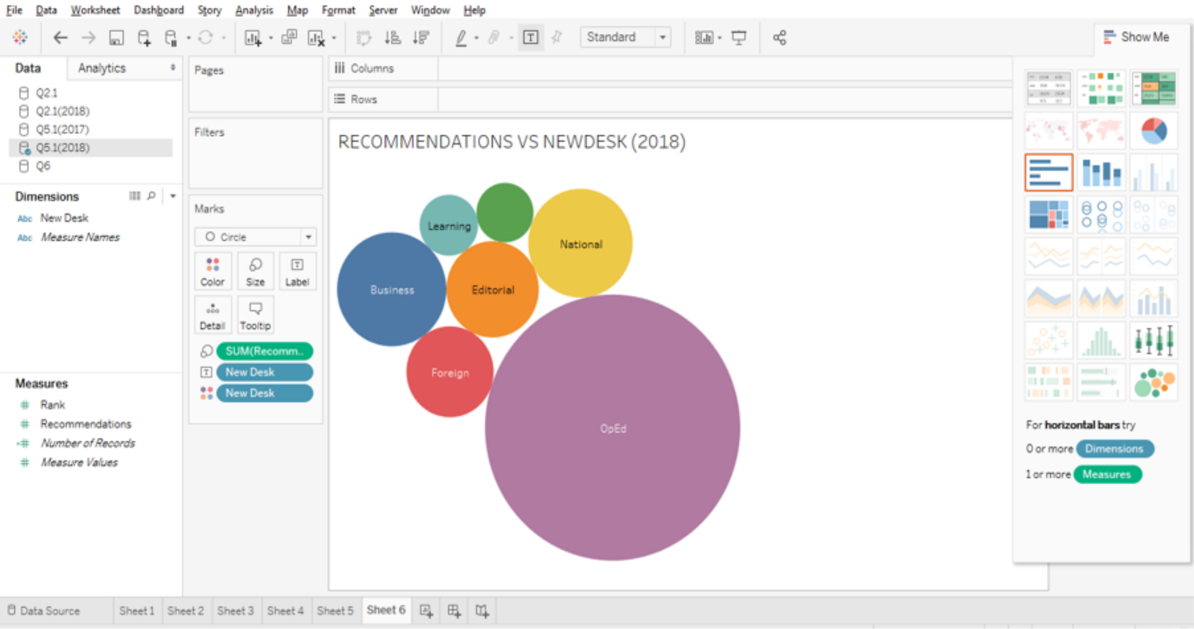
For the year 2017



For the year 2018

Open the output file from “q5.1(2018)” folder in Tableau. After loading the data in Tableau, as shown in the picture below to create our visualization.

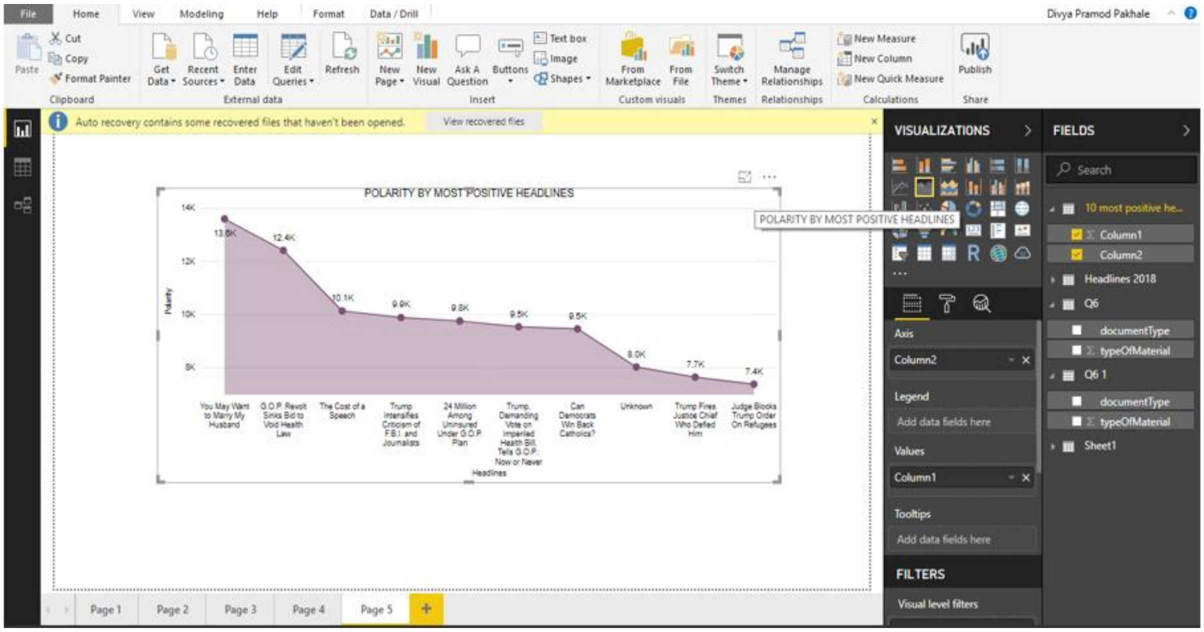
Drag New Desk from the dimensions field into rows and drag recommendations from measures into text in the marks field and then clicked on packed bubbles from the show me tab to get the visualization.



Query 6: What is the degree of polarity by most positive headlines.

Open the power bi tool. Click on Get data -> Excel -> Select your file (Ten most positive headlines)

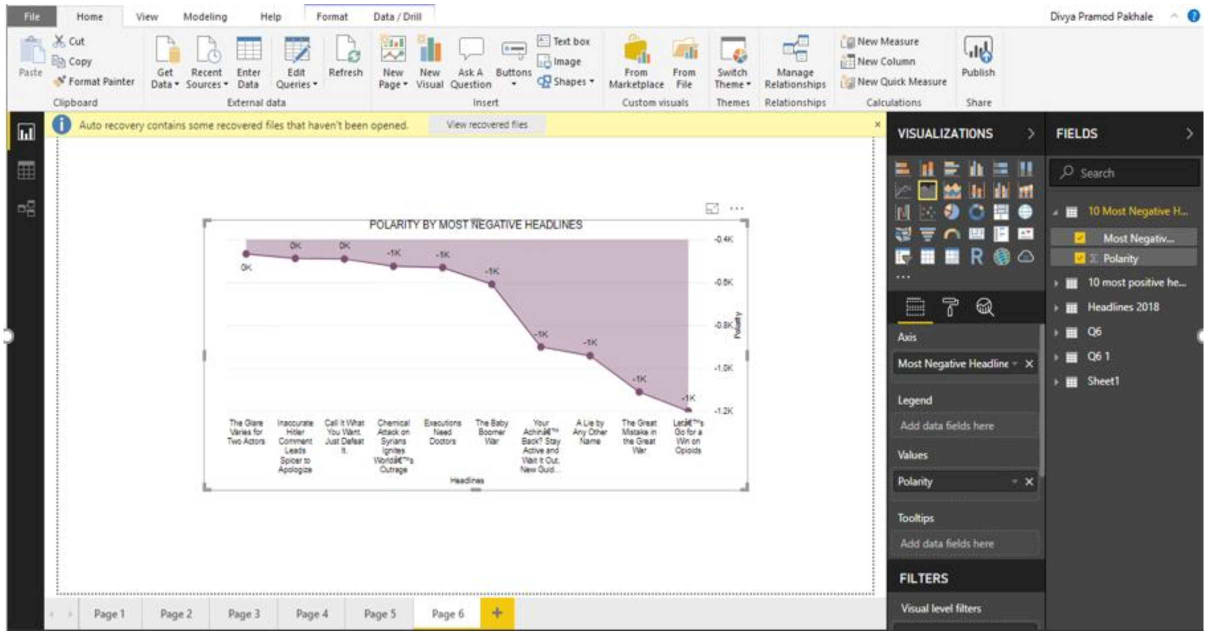
Drag column 1- Degree of Polarity and column 2 - 10 Most Positive Headlines based on the public comments as shown in the picture below and select area chart visualization from the visualization field.


File
Home
View
Modeling
Help
Format
Data / Drill
Divya Pramod Pakhale
Auto recovery contains some recovered files that haven't been opened.
View recovered files
POLARITY BY MOST POSITIVE HEADLINES
14K
12K
10K
8K
Polarity
13.6K
12.4K
10.1K
9.9K
9.8K
9.5K
9.5K
8.0K
7.7K
7.4K
You May Want to Marry My Husband
G.O.P. Revolt Sinks Bid to Void Health Law
The Cost of a Speech
Trump Intensifies Criticism of F.B.I. and Journalists
24 Million Among Uninsured Under G.O.P. Plan
Trump, Demanding Vote on Imperiled Health Bill, Tells G.O.P.: Now or Never
Can Democrats Win Back Catholics?
Unknown
Trump Fires Justice Chief Who Defied Him
Judge Blocks Trump Order On Refugees
Headlines
VISUALIZATIONS
FIELDS
Axis
Column2
Legend
Values
Column1
Tooltips
FILTERS
Visual level filters
10 most positive he...
Column1
Column2
Headlines 2018
Q6
documentType
typeOfMaterial
Q6 1
Sheet1
Page 1
Page 2
Page 3
Page 4
Page 5

Query 7: What is the degree of polarity by most negative headlines.

Open the power bi tool. Click on Get data -> Excel -> Select your file (Ten most negative headlines)

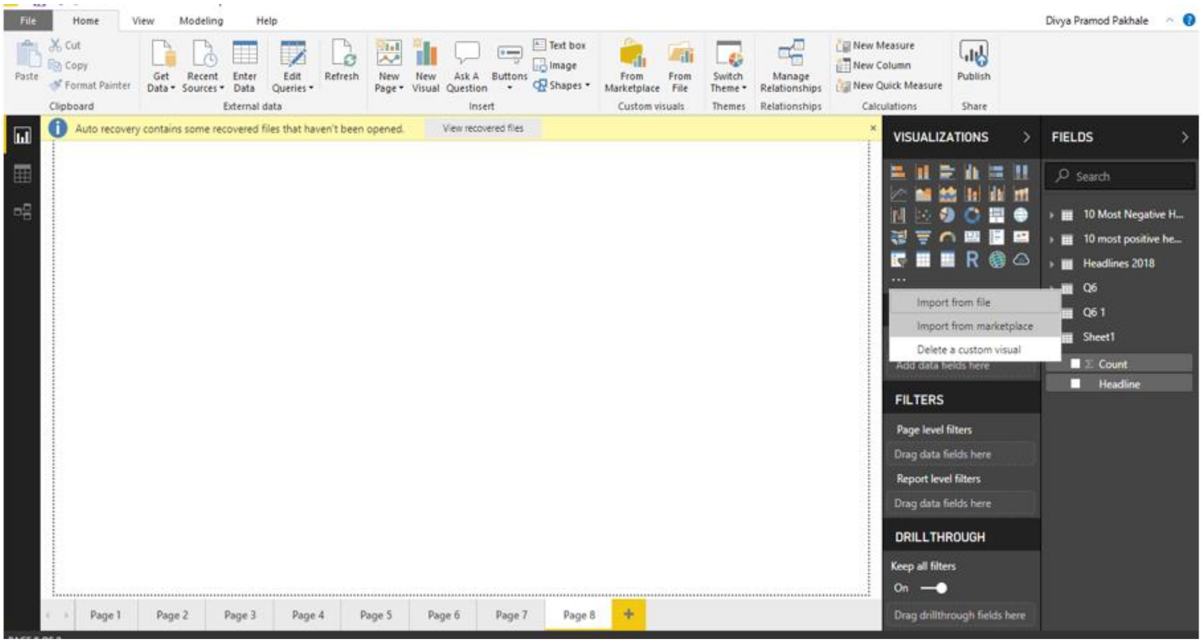
Drag most negative headlines and degree of polarity columns based on the public comments as shown in the picture below and select area chart visualization from the visualization field.



Query 8: What is the most common words in the headlines for article year 2017.

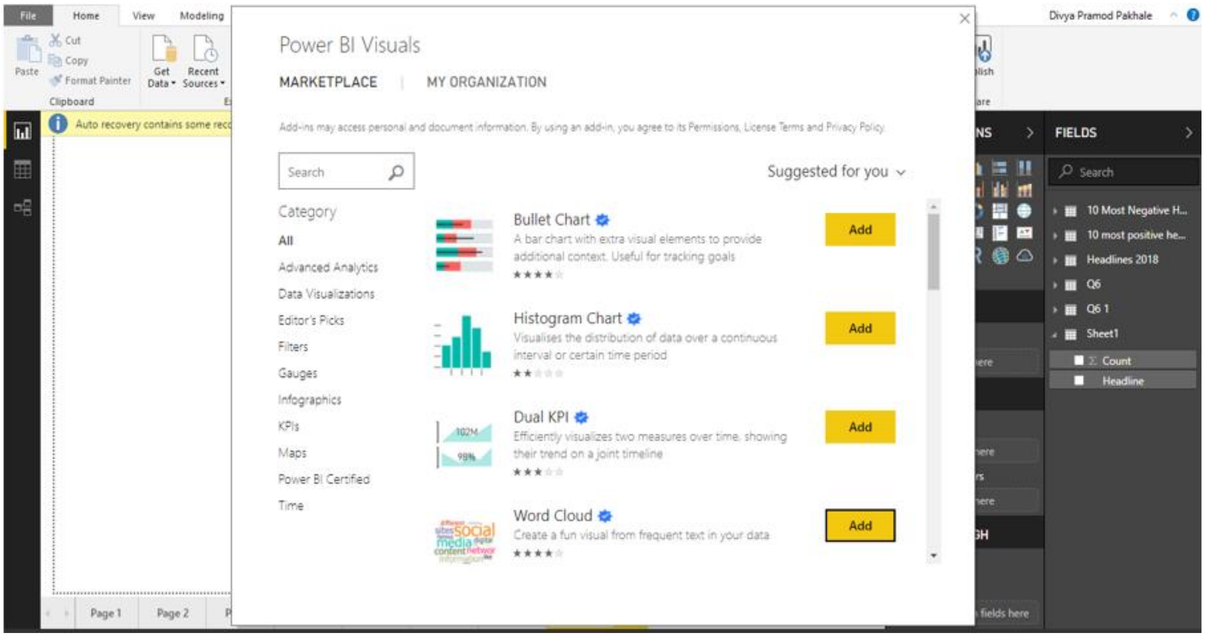
Open the power bi tool. Click on Get data -> Excel -> Select your file (Sheet1)

Click on … from the visualization field and select Import from marketplace.



Auto recovery contains some recovered files that haven't been opened. View recovered files

Click on ADD button for word cloud



Add-ins may access personal and document information. By using an add-in, you agree to its Permissions, License Terms and Privacy Policy.

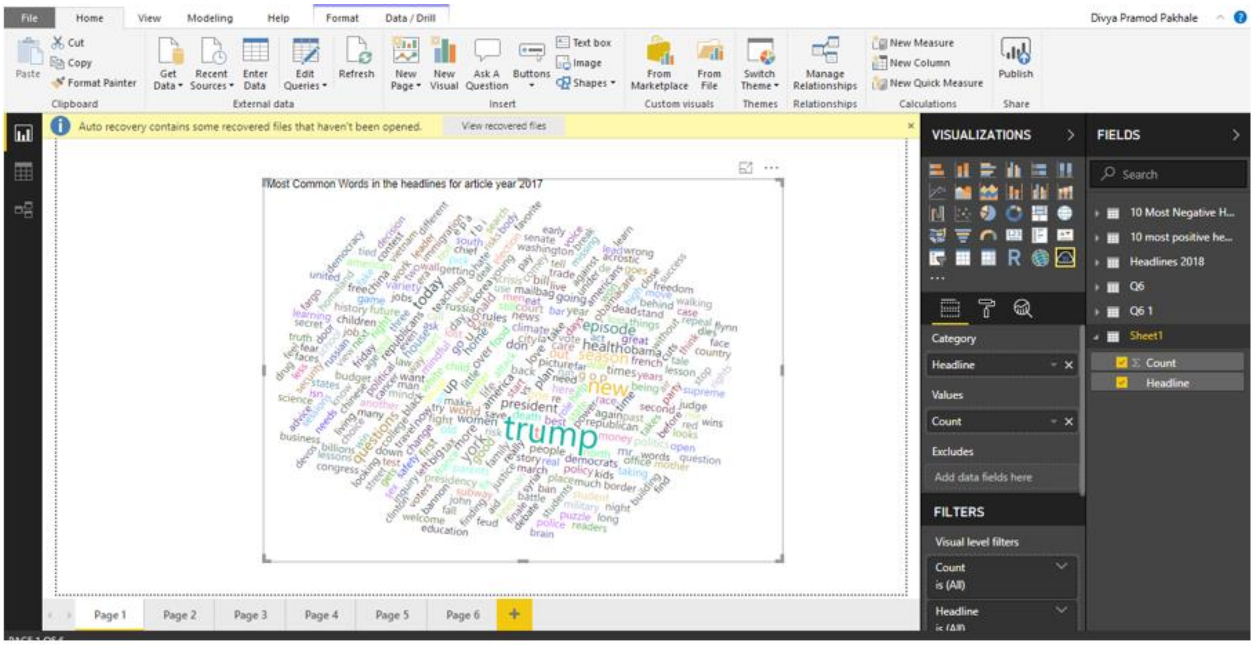
Bullet Chart: A bar chart with extra visual elements to provide additional context. Useful for tracking goals

Histogram Chart: Visualises the distribution of data over a continuous interval or certain time period

Dual KPI: Efficiently visualizes two measures over time, showing their trend on a joint timeline

Word Cloud: Create a fun visual from frequent text in your data

Drag Count and headlines column and make the changes accordingly



Auto recovery contains some recovered files that haven't been opened.

Category: Headline

Values: Count

Excludes: Add data fields here

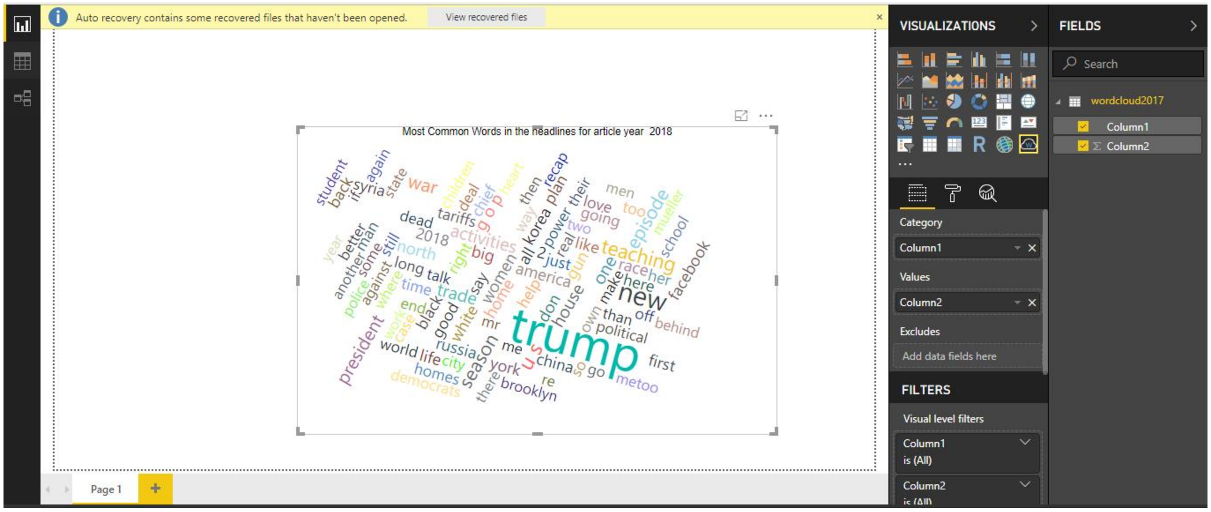
FILTERS

Visual level filters

Count is (All)

Headline is (All)

Query 9: What are the most common words in the headlines for article year 2018.



Auto recovery contains some recovered files that haven't been opened.

Category: Column1

Values: Column2

Excludes: Add data fields here

FILTERS

Visual level filters

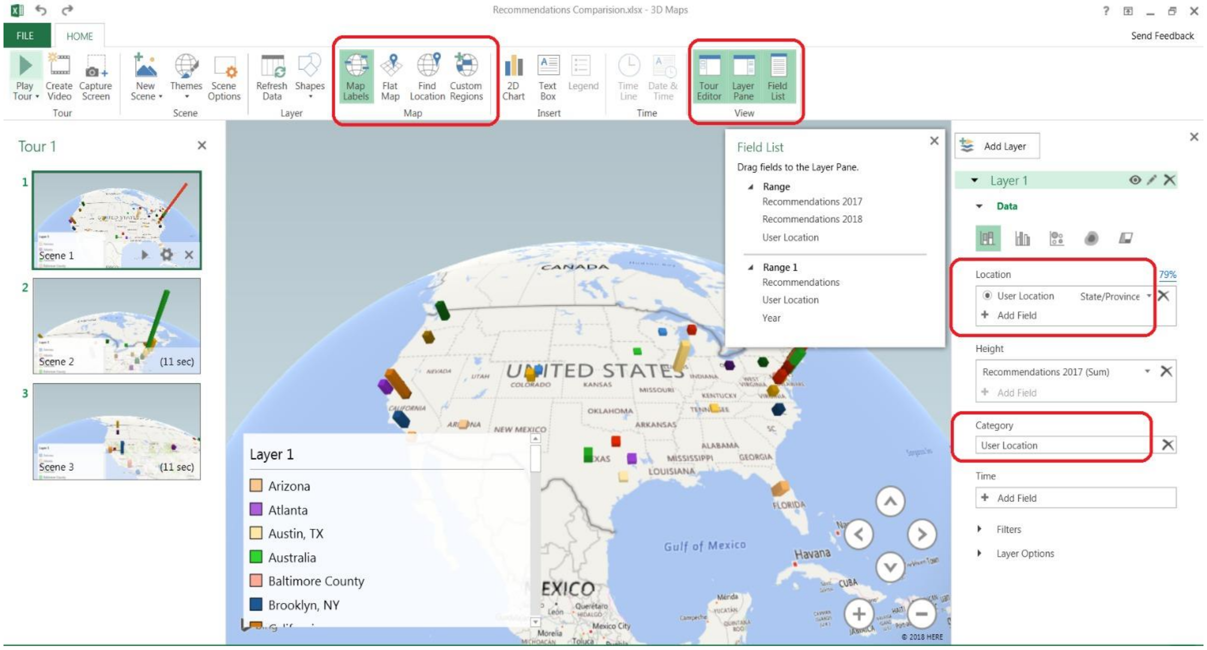
Column1 is (All)

Column2 is (All)

Query 10: What are the recommendations by the user's location for the year 2017.

To create the 3D-Map, follow these steps:

1. Load the output file for the respective query (which had been extracted from Hive) into Microsoft Excel.
2. Select the data, including the column headers in the table format.
3. Click Insert | 3D Maps | Open 3D Maps.
4. Drag fields (column header names) to the Layer panel as shown in the screenshot.
5. Reveal the 3D-Map.



Field List

Drag fields to the Layer Pane.

- Range
  - Recommendations 2017
  - Recommendations 2018
  - User Location
- Range 1
  - Recommendations
  - User Location
  - Year

Layer 1

Data

Location: User Location — State/Province (79%)

Height: Recommendations 2017 (Sum)

Category: User Location

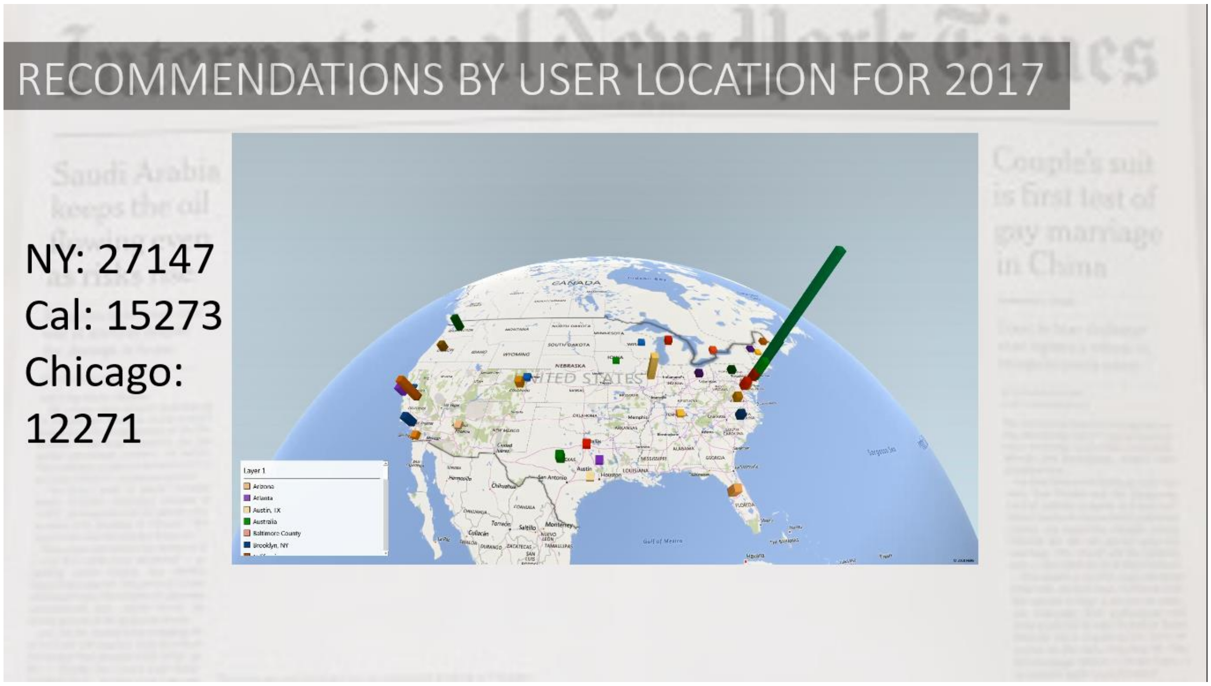
Time: Add Field

Filters

Layer Options

Layer 1

- Arizona
- Atlanta
- Austin, TX
- Australia
- Baltimore County
- Brooklyn, NY

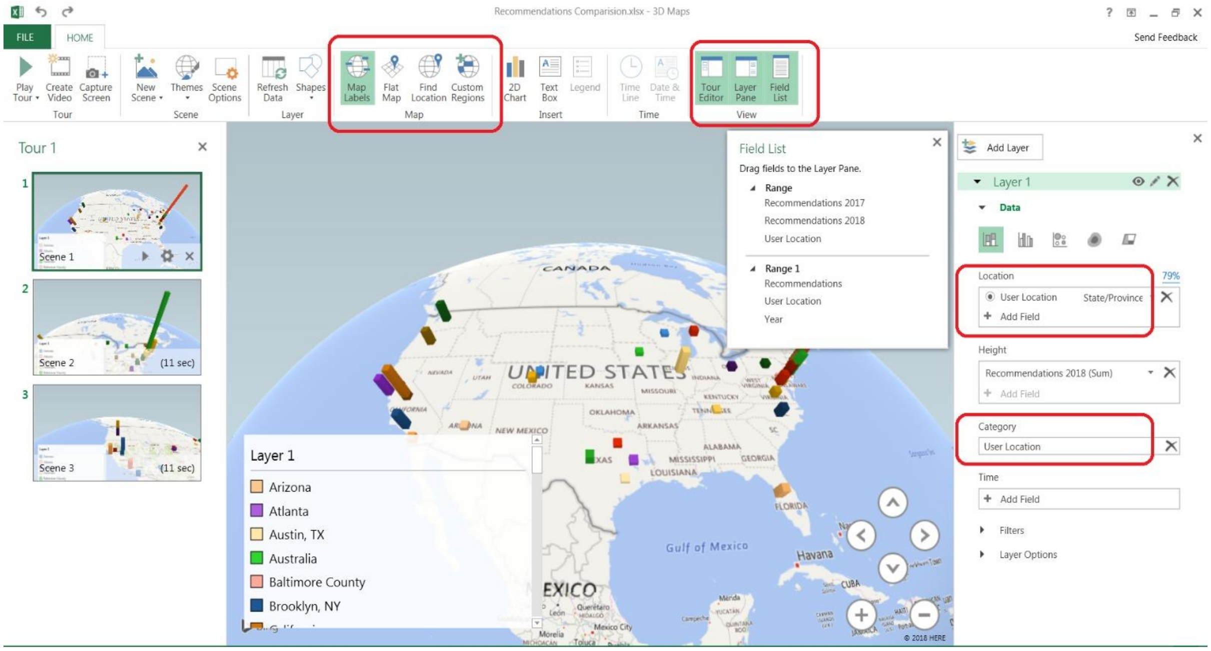


RECOMMENDATIONS BY USER LOCATION FOR 2017

NY: 27147
Cal: 15273
Chicago:
12271

Query 11: What are the recommendations by the user's location for the year 2018.

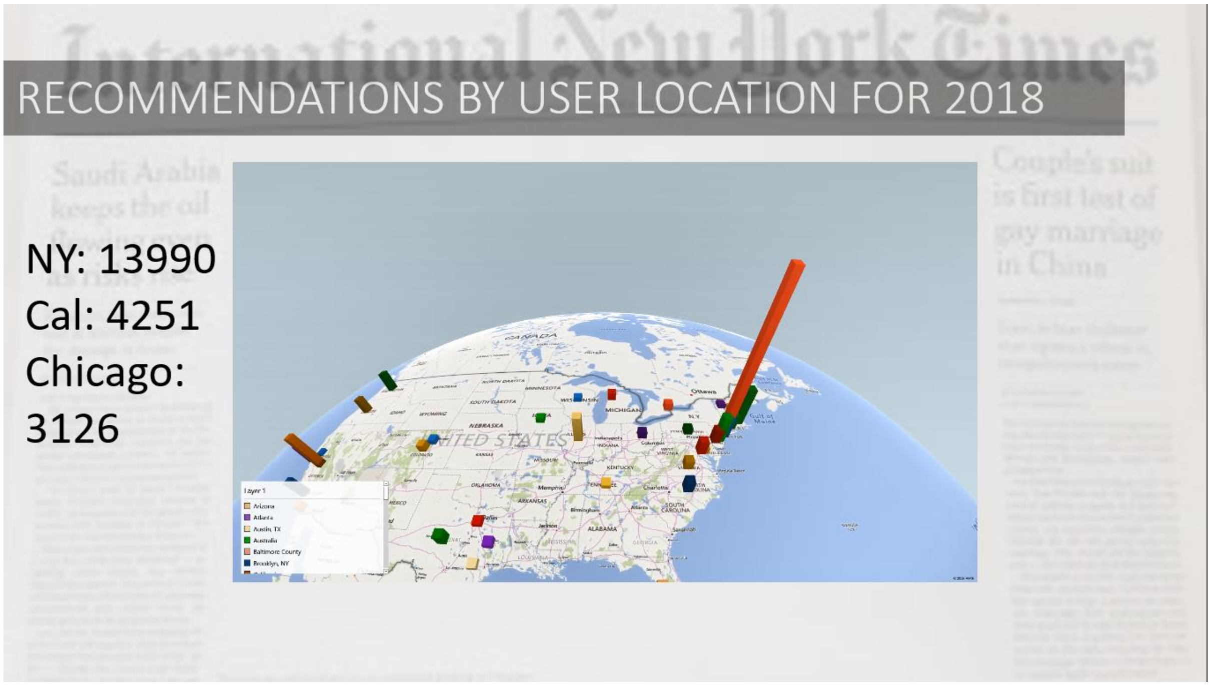
(Follow the same steps as mentioned above)



Recommendations Comparision.xlsx - 3D Maps

Field List
Drag fields to the Layer Pane.
Range
Recommendations 2017
Recommendations 2018
User Location
Range 1
Recommendations
User Location
Year

Layer 1
Data
Location: User Location — State/Province
Height: Recommendations 2018 (Sum)
Category: User Location
Time: Add Field



RECOMMENDATIONS BY USER LOCATION FOR 2018

NY: 13990

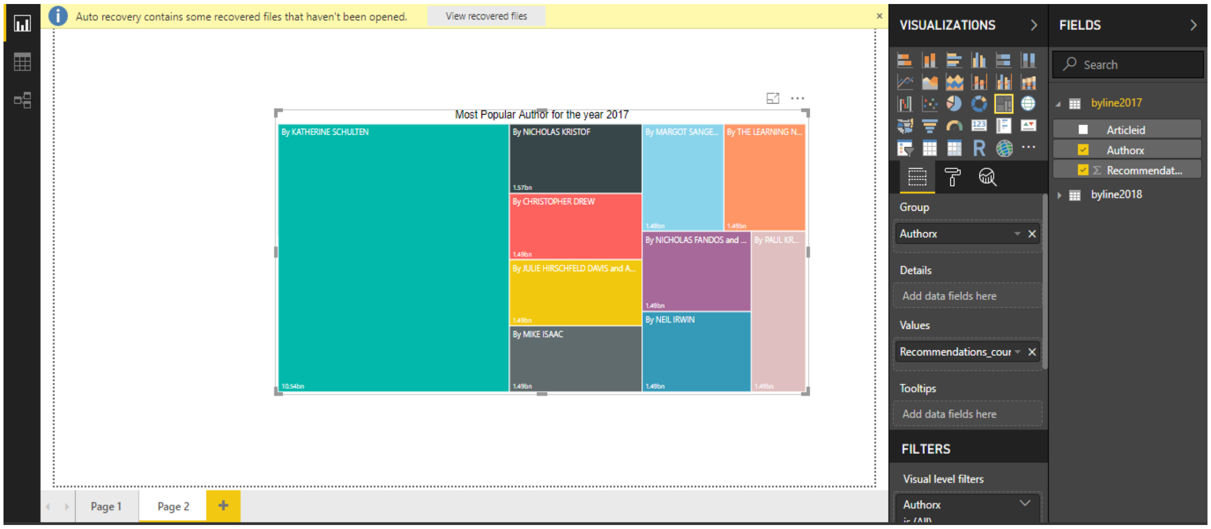
Cal: 4251

Chicago: 3126

Query 12: Most Popular Author(byline) with respect to recommendations of public for the year 2017?

Open the power bi tool. Click on Get data -> Excel -> Select your file (byline2017.csv)

Drag Author and Recommendations column as shown in the picture below and select tree map chart visualization from the visualization field.



Auto recovery contains some recovered files that haven't been opened.

Group: Authorx

Details: Add data fields here

Values: Recommendations_cour

Tooltips: Add data fields here

FILTERS

Visual level filters

Authorx is (All)

FIELDS

byline2017

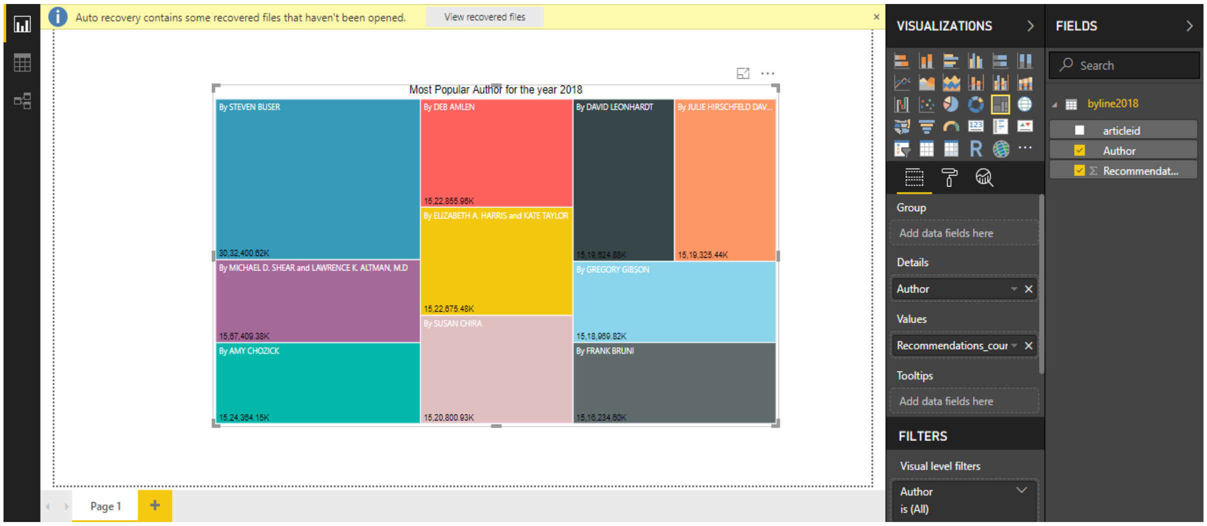
- ☐ Articleid
- ☑ Authorx
- ☑ Σ Recommendat...

byline2018

Query 13: Most Popular Author(byline) with respect to recommendations of public for the year 2018?

Open the power bi tool. Click on Get data -> Excel -> Select your file (byline2018.csv)

Drag Author and Recommendations column as shown in the picture below and select tree map chart visualization from the visualization field.



Auto recovery contains some recovered files that haven't been opened.

Group: Add data fields here

Details: Author

Values: Recommendations_cour

Tooltips: Add data fields here

FILTERS

Visual level filters

Author is (All)

REFERENCES AND GITHUB LINK

1. https://github.com/tanvigawade/Project5200_Group3

2. https://towardsdatascience.com/predicting-popularity-of-the-new-york-times-comments-part-1-d32f26261f6f

3. https://www.kaggle.com/aashita/word-clouds-of-various-shapes

4. https://www.kaggle.com/aashita/exploratory-data-analysis-of-comments-on-nyt/notebook

This is the end of the lab