1) SQL Queries(Acquire the top 200,000 posts by viewcount)

1) SELECT TOP 50000 * from posts ORDER BY viewCount DESC;
2) SELECT * from posts ORDER BY viewCount DESC OFFSET 100000 ROWS FETCH NEXT 50000 ROWS ONLY;
3) SELECT * from posts ORDER BY viewCount DESC OFFSET 50000 ROWS FETCH NEXT 50000 ROWS ONLY;
4) SELECT * from posts ORDER BY viewCount DESC OFFSET 150000 ROWS FETCH NEXT 50000 ROWS ONLY;

2) HADOOP Commands

1) ls #List all the uploaded files
2) cat First.csv > mainfile.csv #Concat all the csv's into one
3) cat Second.csv >> mainfile.csv #Concat all the csv's into one
4) cat Third.csv >> mainfile.csv #Concat all the csv's into one
5) cat Fourth.csv >> mainfile.csv #Concat all the csv's into one
6) ls -lah mainfile.csv #To Know the size of the file
7) sed ':a;N;$!ba;s/\n/ /g' mainfile.csv > mainfile1.csv #It will replace all \n to \\n so it escape all returns to new line
8) hadoop fs -put mainfile1.csv /Assignment1 #Put csv in Hadoop
9) head -c 3000 mainfile1. #Print Head of the file

```
g2sagar771994@cluster-77fc-m:~$ ls -lah mainfile.csv
-rw-r--r-- 1 g2sagar771994 g2sagar771994 253M Nov  4 19:56 mainfile.csv
```

```
g2sagar771994@cluster-77fc-m:~$ sed ':a;N;$!ba;s/\n/ /g' mainfile.csv > mainfile1.csv
```

3) PIG Commands(Using Pig or MapReduce, extract, transform and load the data as applicable)

1) pig #To Enter Pig
2) cd ../.. #To enter current directory
3) Assignment1data = LOAD '/Assignment1' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES_MULTILINE', 'NOCHANGE', 'SKIP_INPUT_HEADER') as (id :int,posttypeid :int,acceptedanswerid :int,parentid :int,creationdate :chararray,deletiondate :chararray,score :int,viewcount :int,body :chararray,owneruserid :int,ownerdisplayname :chararray,lasteditoruserid :int,lasteditordisplayname :chararray,lasteditdate :chararray,lastactivitydate :chararray,title :chararray,tags :chararray,answercount :int,commentcount :int,favoritecount :int,closeddate :chararray,communityowneddate : chararray); #Load Data in an Assignment1data by giving required schema
4) cleandata = FOREACH Assignment1data GENERATE id,score,viewcount,REPLACE(REPLACE(body,'\n',''),',','') AS body,owneruserid,ownerdisplayname,title,tags; #Clean Data using replace function

```
grunt> Assignment1data = LOAD '/Assignment1' USING org.apache.pig.piggybank.storage.CSVExcelStor
age(',', 'YES_MULTILINE', 'NOCHANGE', 'SKIP_INPUT_HEADER') as (id :int,posttypeid :int,accepteda
nswerid :int,parentid :int,creationdate :chararray,deletiondate :chararray,score :int,viewcount
:int,body :chararray,owneruserid :int,ownerdisplayname :chararray,lasteditoruserid :int,lastedit
ordisplayname :chararray,lasteditdate :chararray,lastactivitydate :chararray,title :chararray,ta
gs :chararray,answercount :int,commentcount :int,favoritecount :int,closeddate :chararray,commun
ityowneddate : chararray);
```

```
grunt> cleandata = FOREACH Assignment1data  GENERATE  id,score,viewcount,REPLACE(REPLACE(body,'\
n',''),',','') AS body,owneruserid,ownerdisplayname,title,tags;
```

5) cleaned = FOREACH cleandata GENERATE FLATTEN((id,score,viewcount,body,owneruserid,ownerdisplayname,title,tags)); #Flatten Dataset
6) limitcleaned = LIMIT cleaned 10; #Limiting Data to dump
7) dump limitcleaned #Dump the limiting file
8) STORE cleaned INTO '/OutputPig' USING PigStorage (','); #Store clean data in directory to use it in hive

```
grunt> cleaned = FOREACH cleandata GENERATE FLATTEN((id,score,viewcount,body,owneruserid,ownerdi
splayname,title,tags));
```

9) fs -ls /OutputPig #list all the files
10) fs -rm /OutputPig/_SUCCESS; #Remove the success
11) fs -tail /OutputPig/part-m-00000 #Print tail data of the file
12) fs -tail /OutputPig/part-m-00001 #Print tail data of the file

4) Hive(Hive Queries)

1) Create database HiveData
2) CREATE TABLE Assignment(id int, score BIGINT, viewcount BIGINT, body string, owneruserid string, ownerdisplayname string, title string, tags string)ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'; #Create Table to store data
3) LOAD DATA INPATH '/OutputPig' INTO TABLE Assignment; #Load data into created table from Pig Directory
4) Select count(*) from Assignment #Get count of all the data in the table

```
hive> CREATE TABLE Assignment(id int, score BIGINT, viewcount BIGINT, body string, owneruserid s
tring, ownerdisplayname string, title string, tags string)ROW FORMAT DELIMITED FIELDS TERMINATED
 BY ',' LINES TERMINATED BY '\n';
OK
```

5) SELECT id,score,title FROM Assignment ORDER BY score DESC LIMIT 10; (The top 10 posts by score) #Top 10 post by score
6) SELECT owneruserid AS USER_NAME,SUM(SCORE) AS TOTAL_SCORE FROM Assignment WHERE owneruserid != '' GROUP BY owneruserid ORDER BY TOTAL_SCORE DESC LIMIT 10; (The top 10 users by post score) #Top 10 user by post score
7) SELECT COUNT(DISTINCT(owneruserid)) FROM Assignment WHERE lower(body) like '%hadoop%' or lower(tags) like '%hadoop%' or lower(title) like '%hadoop%'; (The number of distinct users, who used the word “Hadoop” in one of their posts) #Number of Distinct users who used word Hadoop in their post

```
hive> SELECT id,score,title FROM Assignment ORDER BY score DESC LIMIT 10;
Query ID = g2sagar771994_20191104203419_9c768936-5f1f-4aef-8f7a-685482f5077e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1572864186099_0024)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      5          5        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 21.06 s
--------------------------------------------------------------------------------
OK
405783  1625    Why does man print "gimme gimme gimme" at 00:30?
4126    1236    What is the exact difference between a 'terminal'
185764  1066    How do I get the size of a directory on the command line?
26047   922     How to correctly add a path to PATH?
34196   806     Why was '~' chosen to represent the home directory?
159114  771
112023  751     How can I replace a string in a file(s)?
12107   728     How to unfreeze after accidentally pressing Ctrl-S in a terminal?
106480  676     How to copy files from one machine to another using ssh
18154   642     What is the purpose of the lost+found folder in Linux and Unix?
Time taken: 22.253 seconds, Fetched: 10 row(s)
```

```
--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      5          5        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 25.34 s
--------------------------------------------------------------------------------
OK
885     8851
674     5023
22565   4622
7453    4223
688     4218
29      3372
22222   3276
10287   3022
6960    2961
5614    2757
Time taken: 26.637 seconds, Fetched: 10 row(s)
```

```
hive> SELECT COUNT(DISTINCT(owneruserid)) FROM Assignment WHERE lower(body) like '%hadoop%' or l
ower(tags) like '%hadoop%' or lower(title) like '%hadoop%';
Query ID = g2sagar771994_20191104203845_a394d44a-411e-4660-ac08-736a7d986843
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1572864186099_0024)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      5          5        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 27.27 s
--------------------------------------------------------------------------------
OK
126
Time taken: 28.423 seconds, Fetched: 1 row(s)
```

5) TFIDF(Using Mapreduce calculate the per-user TF-IDF)

1) create table Usersdataposts as SELECT owneruserid AS USER_NAME,SUM(SCORE) AS TOTAL_SCORE FROM Assignment WHERE owneruserid != '' GROUP BY owneruserid ORDER BY TOTAL_SCORE DESC LIMIT 10; #Creating a new table with all the required columns
2) select USER_NAME,TOTAL_SCORE from Usersdataposts; #Print selected Data from Usersdataposts

```
hive> create table Usersdataposts as SELECT owneruserid AS USER_NAME,SUM(SCORE) AS TOTAL_SCORE F
ROM Assignment WHERE owneruserid != '' GROUP BY owneruserid ORDER BY TOTAL_SCORE DESC LIMIT 10;

Query ID = g2sagar771994_20191104222140_da9f4d81-9013-41a5-b2e7-72da7e618033
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1572864186099_0044)
```

3) select owneruserid, body from Assignment where owneruserid in (select USER_NAME from Usersdataposts); #Print data from both the table
4) insert overwrite local directory '/home/g2sagar771994/tfidfNew' row format delimited fields terminated by ',' select owneruserid, body from Assignment where owneruserid in (select USER_NAME from Usersdataposts); #Create a new directory with all the data from the previous query with delimiter as comma

```
hive> insert overwrite local directory '/home/g2sagar771994/Tfidf' row format delimited fields t
erminated by ',' select owneruserid, body from Assignment where owneruserid in (select USER_NAME
 from Users_data_posts);
Query ID = g2sagar771994_20191104213417_29ad075a-f7fd-47e2-95a1-c4704f80a7ae
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1572864186099_0030)
```

5) hadoop fs -put /home/g2sagar771994/tfidfNew /input_data
6) hadoop fs -ls /input_data
7) hadoop fs -ls /input_data/tfidfNew #Check if data is loaded

```
g2sagar771994@cluster-77fc-m:~/tfidfNew$ hadoop fs -ls /input_data
Found 1 items
drwxr-xr-x   - g2sagar771994 hadoop          0 2019-11-04 22:29 /input_data/tfidfNew
g2sagar771994@cluster-77fc-m:~/tfidfNew$ hadoop fs -ls /input_data/tfidfNew
Found 5 items
-rw-r--r--   2 g2sagar771994 hadoop     929693 2019-11-04 22:29 /input_data/tfidfNew/000000_0
-rw-r--r--   2 g2sagar771994 hadoop     591667 2019-11-04 22:29 /input_data/tfidfNew/000001_0
-rw-r--r--   2 g2sagar771994 hadoop    2808114 2019-11-04 22:29 /input_data/tfidfNew/000002_0
-rw-r--r--   2 g2sagar771994 hadoop    1381098 2019-11-04 22:29 /input_data/tfidfNew/000003_0
-rw-r--r--   2 g2sagar771994 hadoop     474126 2019-11-04 22:29 /input_data/tfidfNew/000004_0
```

8) hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/g2sagar771994/MapperPhaseOne.py /home/g2sagar771994/ReducerPhaseOne.py -mapper "python MapperPhaseOne.py" -reducer "python ReducerPhaseOne.py" -input /input_data/tfidfNew -output /output1 #Loading the jar, then giving the path of mapper and reducer and then running it with input files and output location [1]
9) hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/g2sagar771994/MapperPhaseTwo.py /home/g2sagar771994/ReducerPhaseTwo.py -mapper "python MapperPhaseTwo.py" -reducer "python ReducerPhaseTwo.py" -input /output1/part-00000 /output1/part-00001 -output /output2 #Loading the jar, then giving the path of mapper and reducer and then running it with input files and output location [1]
10) hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/g2sagar771994/MapperPhaseThree.py /home/g2sagar771994/ReducerPhaseThree.py -mapper "python MapperPhaseThree.py" -reducer "python ReducerPhaseThree.py" -input /output2/part-00000 /output2/part-00001 -output /output3 #Loading the jar, then giving the path of mapper and reducer and then running it with input files and output location [1]

```
19/11/04 22:34:51 INFO streaming.StreamJob: Output directory: /output1
g2sagar771994@cluster-77fc-m:~/tfidfNew$
```

```
19/11/04 22:38:33 INFO streaming.StreamJob: Output directory: /output2
```

```
19/11/04 22:42:14 INFO streaming.StreamJob: Output directory: /output3
```

11) hadoop fs -getmerge /output3/part-00000 /output3/part-00001 /home/g2sagar771994/NewData_TFIDF.csv #Merging the generated files
12) sed -e 's/\s/,/g' NewData_TFIDF.csv > NewData_TFIDF1.csv #Replacing all the spaces with comma and storing it in a new file

```
g2sagar771994@cluster-77fc-m:~$ hadoop fs -getmerge /output3/part-00000 /output3/part-00001 /hom
e/g2sagar771994/NewData_TFIDF.csv
```

```
g2sagar771994@cluster-77fc-m:~$ sed -e 's/\s/,/g' NewData_TFIDF.csv > NewData_TFIDF1.csv
```

13) create external table if not exists TFIDF_Table(Term string,Id string,tfidf float) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','; #Create a new table
14) LOAD DATA LOCAL INPATH 'NewData_TFIDF1.csv' OVERWRITE INTO TABLE TFIDF_Table; #Load the data of the csv in the new table

```
hive> create external table if not exists TFIDF_Table(Term string,Id string,tfidf float) ROW FOR
MAT DELIMITED FIELDS TERMINATED BY ',';
OK
```

```
hive> LOAD DATA LOCAL INPATH 'NewData_TFIDF1.csv' OVERWRITE INTO TABLE TFIDF_Table;
Loading data to table default.tfidf_table
OK
```

15) CREATE TABLE final_tfidf AS (select term,regexp_replace(id,'[^0-9]','') as id,tfidf from TFIDF_Table); #Removing all the unwanted things from data and selecting only numerals from 0-9 with regular expression
16) Create table final_main as (select * from final_tfidf where id in (select USER_NAME from Usersdataposts)); #Create a new table, select all the columns from table final_tfidf with id's of all the USER_NAME in Usersdataposts
17) SELECT * FROM (SELECT id,term,tfidf, ROW_NUMBER() OVER (PARTITION BY id ORDER BY tfidf DESC) AS RANK FROM final_main WHERE id != '') t WHERE RANK <= 10 ; #Selection columns, partitioning and ordering it from the last table and filtering the id and restricting the data to top 10

```
--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 7.80 s
--------------------------------------------------------------------------------
OK
10287   sshknownhosts   0.27272728      1
10287   errorreporting  0.25714287      2
10287   EALL    0.25714287      3
10287   ENOTICE 0.25714287      4
10287   omeka   0.18620689      5
10287   hrefhttpimg830imageshackusimg8304526screenshotwithshadowpng     0.16363636      6
10287   httpimg830imageshackusimg8304526screenshotwithshadowpngap       0.16363636      7
10287   utilise 0.16363636      8
10287   precodeUnless   0.16363636      9
10287   Pagea   0.16363636      10
22222   codeiconscode   0.39130434      1
22222   codethemescodep 0.39130434      2
22222   codegnome2code  0.39130434      3
7453    octalli 0.3272727       7
7453    decimalli       0.3272727       8
7453    783M    0.31034482      9
7453    2667000 0.2982456       10
885     Packet  0.75    1
885     wordlist        0.75    2
885     ownersoffilesinvar      0.7297297       3
885     musl    0.6923077       4
885     countertxt      0.6506024       5
885     OCR     0.64285713      6
885     0ad     0.5943396       7
885     schmijos        0.5625  8
885     Cabeginningofline       0.5294118       9
885     kbdCtrlkbdkbdAkbdp      0.5294118       10
Time taken: 12.597 seconds, Fetched: 100 row(s)
hive>
```

REFERENCES:-

1) https://github.com/SatishUC15/TFIDF-HadoopMapReduce

Changes made in the python Code:

2) Added stopwords in the mapper one python file
3) Used stack exchange data set as input data set for the mapper reducer operation
4) Imported punctuation library in mapper one file to use some string functions required in mapper reducer operation.