**#To Display all the uploaded files in GCP:**

ls

**#To upload files on Hadoop:**

hadoop fs -ls QueryResults\* /

**#In Pig:**

REGISTER /usr/lib/pig/piggybank.jar

DEFINE CSVLoader org.apache.pig.piggybank.storage.CSVLoader();

pwd

cd ../..

**#Loading files in Pig**

datastack1 = LOAD 'QueryResults1.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE', 'NOCHANGE','SKIP\_INPUT\_HEADER') as (Id:chararray, PostTypeId:chararray, AcceptedAnswerId:chararray, ParentId:chararray, CreationDate:chararray, DeletionDate:chararray, Score:chararray, ViewCount:chararray, Body:chararray, OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray, LastEditorDisplayName:chararray, LastEditDate:chararray, LastActivityDate:chararray, Title:chararray, Tags:chararray, AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray, ClosedDate:chararray, CommunityOwnedDate:chararray);

datastack2 = LOAD 'QueryResults2.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE', 'NOCHANGE','SKIP\_INPUT\_HEADER') as (Id:chararray, PostTypeId:chararray, AcceptedAnswerId:chararray, ParentId:chararray, CreationDate:chararray, DeletionDate:chararray, Score:chararray, ViewCount:chararray, Body:chararray, OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray, LastEditorDisplayName:chararray, LastEditDate:chararray, LastActivityDate:chararray, Title:chararray, Tags:chararray, AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray, ClosedDate:chararray, CommunityOwnedDate:chararray);

datastack3 = LOAD 'QueryResults3.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE', 'NOCHANGE','SKIP\_INPUT\_HEADER') as (Id:chararray, PostTypeId:chararray, AcceptedAnswerId:chararray, ParentId:chararray, CreationDate:chararray, DeletionDate:chararray, Score:chararray, ViewCount:chararray, Body:chararray, OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray, LastEditorDisplayName:chararray, LastEditDate:chararray, LastActivityDate:chararray, Title:chararray, Tags:chararray, AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray, ClosedDate:chararray, CommunityOwnedDate:chararray);

datastack4 = LOAD 'QueryResults4.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE', 'NOCHANGE','SKIP\_INPUT\_HEADER') as (Id:chararray, PostTypeId:chararray, AcceptedAnswerId:chararray, ParentId:chararray, CreationDate:chararray, DeletionDate:chararray, Score:chararray, ViewCount:chararray, Body:chararray, OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray, LastEditorDisplayName:chararray, LastEditDate:chararray, LastActivityDate:chararray, Title:chararray, Tags:chararray, AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray, ClosedDate:chararray, CommunityOwnedDate:chararray);

datastack5 = LOAD 'QueryResults5.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE', 'NOCHANGE','SKIP\_INPUT\_HEADER') as (Id:chararray, PostTypeId:chararray, AcceptedAnswerId:chararray, ParentId:chararray, CreationDate:chararray, DeletionDate:chararray, Score:chararray, ViewCount:chararray, Body:chararray, OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray, LastEditorDisplayName:chararray, LastEditDate:chararray, LastActivityDate:chararray, Title:chararray, Tags:chararray, AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray, ClosedDate:chararray, CommunityOwnedDate:chararray);

**#Merging all files in one file**

combined\_data = UNION datastack1, datastack2, datastack3, datastack4,datastack5 ;

**#Cleaning of Data**

combined\_data\_1 = FOREACH combined\_data GENERATE Id, Score, ViewCount, Body,OwnerUserId, OwnerDisplayName, Title, Tags;

combined\_data\_2 = FILTER combined\_data\_1 by ((OwnerUserId != '') AND (OwnerDisplayName != ''));

combined\_data\_3 = FOREACH combined\_data\_2 GENERATE REPLACE(REPLACE(REPLACE(REPLACE(Id,'\\n',''),'\\r',''),'\\r\\n',''),'<br>','') as Id,REPLACE(REPLACE(REPLACE(REPLACE(Score,'\\n',''),'\\r',''),'\\r\\n',''),'<br>','') as Score,ViewCount,REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(Body,'\'',''),'\\"',''),'\\.',''),'\\..,',''),',',''),'\\.,',''),'\\n','') as Body,OwnerUserId, OwnerDisplayName, Title, Tags;

**#Storing of Data in Hadoop**

STORE combined\_data\_3 INTO 'result1' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',','YES\_MULTILINE','NOCHANGE');

**#Merging of all output parts into one csv file**

hadoop fs -getmerge hdfs://cluster-9caf-m/user/tejal\_nijai2/result1/part-m-00000 hdfs://cluster-9caf-m/user/tejal\_nijai2/result1/part-m-00001 hdfs://cluster-9caf-m/user/tejal\_nijai2/result1/part-m-00002 hdfs://cluster-9caf-m/user/tejal\_nijai2/result1/part-m-00003 hdfs://cluster-9caf-m/user/tejal\_nijai2/result1/part-m-00004 /home/tejal\_nijai2/Query\_Data.csv

**#Creating table in Hive**

hive> create external table if not exists Stack\_Exch\_data (Id int, Score int, ViewCount int,Body String, OwnerUserId int, OwnerDisplayName string, Title string, Tags string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',';

hive> load data local inpath 'Query\_Data.csv' overwrite into table Stack\_Exch\_data;

**#Q.3(I).The top 10 posts by score**

hive> select Body, Score from Stack\_Exch\_data order by Score desc limit 10;

**#Q.3(II).The top 10 users by post score**

hive> create table post\_score\_users as select ownerUserId as a, SUM(Score) as b from Stack\_Exch\_data group by ownerUserId;

hive> select \* from post\_score\_users order by b desc limit 10;

**#Q.3(III).The number of distinct users, who used the word “Hadoop” in one of their posts**

hive> select COUNT(DISTINCT OwnerUserId) from Stack\_Exch\_data where lower(Body) like '%hadoop%';

**#Q.4 Solution Approach:**

**#Creating Table to get Top users by their posts score which will be joined to Main Stack\_Exch\_data table to get the posts of all those users**

hive>create table Users\_data\_posts as select a as OwnerId,b as Scr from post\_score\_users f order by Scr desc limit 10;

hive>select OwnerId,Scr from Users\_data\_posts;

hive>select OwnerUserId,Body from Stack\_Exch\_data where OwnerUserId in (select OwnerId from Users\_data\_posts);

**#Creating csv from the above query which will be used as base for calculating TFIDF of the users**

hive>INSERT OVERWRITE LOCAL DIRECTORY '/home/tejal\_nijai2/Datafortfidf'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

select OwnerUserId,Body from Stack\_Exch\_data where OwnerUserId in (select OwnerId from Users\_data\_posts);

**#To check the generated file in GCP**

cd Datafortfidf/

**#Command to remove comma from the file so that file can be used as input for MapReduce program**

sed 's/,/ /g' 000000\_0 > Filetocsv

**#After Uploading all python executables files of mapper and reducer, following command executed to change its privileges**

chmod +x MapperPhase\* ReducerPhase\*

**#A directory was created on hadoop to keep the input data for executable python scripts**

hadoop fs -mkdir /input\_data

hadoop fs -put Datafortfidf/Filetocsv /input\_data

**#Commands to run python scripts**

hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/tejal\_nijai2/MapperPhaseOne.py /home/tejal\_nijai2/ReducerPhaseOne.py -mapper "python MapperPhaseOne.py" -reducer "python ReducerPhaseOne.py" -input hdfs://cluster-9caf-m/input\_data/Filetocsv -output hdfs://cluster-9caf-m/output4

hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/tejal\_nijai2/MapperPhaseTwo.py /home/tejal\_nijai2/ReducerPhaseTwo.py -mapper "python MapperPhaseTwo.py" -reducer "python ReducerPhaseTwo.py" -input hdfs://cluster-9caf-m/output4/part-00000 hdfs://cluster-9caf-m/output4/part-00001 hdfs://cluster-9caf-m/output4/part-00002 hdfs://cluster-9caf-m/output4/part-00003 hdfs://cluster-9caf-m/output4/part-00004 -output hdfs://cluster-9caf-m/output5

hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/tejal\_nijai2/MapperPhaseThree.py /home/tejal\_nijai2/ReducerPhaseThree.py -mapper "python MapperPhaseThree.py" -reducer "python ReducerPhaseThree.py" -input hdfs://cluster-9caf-m/output5/part-00000 hdfs://cluster-9caf-m/output5/part-00001 hdfs://cluster-9caf-m/output5/part-00002 hdfs://cluster-9caf-m/output5/part-00003 hdfs://cluster-9caf-m/output5/part-00004 -output hdfs://cluster-9caf-m/output6

**#Final output of the MapReduce program is merged into a file**

hadoop fs -getmerge hdfs://cluster-9caf-m/output6/part-00000 hdfs://cluster-9caf-m/output6/part-00001 hdfs://cluster-9caf-m/output6/part-00002 hdfs://cluster-9caf-m/output6/part-00003 hdfs://cluster-9caf-m/output6/part-00004 /home/tejal\_nijai2/Tfidf\_output\_data.csv

**#Command to convert a file to Csv file**

sed -e 's/\s/,/g' Tfidf\_output\_data.csv > Tfidf\_final\_output\_data.csv

**#Creating a table in hive of a csv which has all data about TFIDF per user(Users from Q.3(II)) and further to display the top terms of top 10 users referring to Q.3(II)**

create external table if not exists TFIDF\_Final\_Data (Term String,Id int,tfidf float)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',';

load data local inpath 'Tfidf\_final\_output\_data.csv' overwrite into table TFIDF\_Final\_Data;

SELECT \*

FROM (

SELECT ROW\_NUMBER()

OVER(PARTITION BY Id

ORDER BY tfidf DESC) AS TfidfRank, \*

FROM TFIDF\_Final\_Data) n

WHERE TfidfRank IN (1,2,3,4,5,6,7,8,9,10);

**Reference:**

<https://github.com/SatishUC15/TFIDF-HadoopMapReduce>

**MapperPhaseOne.Py**

#!/usr/bin/env python

from string import punctuation

import sys

# TF-IDF computation: Phase One

# Mapper output: <<word, document\_name> 1>

stopwords= ['a','able','about','across','after','all','almost','also','am','among','an','and','any','are','as','at','be','because','been','but','by',

'can','cannot','could','dear','did','do','does','either','else','ever','every','for','from','get','got','had','has','have','he','her','hers',

'him','his','how','however','i','if','in','into','is','it','its','just','least','let','like','likely','may','me','might','most','must','my',

'neither','no','nor','not','of','off','often','on','only','or','other','our','own','rather','said','say','says','she','should','since','so',

'some','than','that','the','their','them','then','there','these','they','this','tis','to','too','twas','us','wants','was','we','were','what',

'when','where','which','while','who','whom','why','will','with','would','yet','you','your'];

for line in sys.stdin:

line = line.translate(None, punctuation).strip('\t')

line\_contents = line.split(" ")

doc\_name = line\_contents[0]

line\_contents.remove(doc\_name)

for content in line\_contents:

content = content.lower()

content = content.rstrip()

if content not in stopwords:

key = content + "," + doc\_name

print '%s\t%s' % (key, 1)