CA675 - CLOUD TECHNOLOGIES

ASSIGNEMENT1 - DATA ANALYSIS

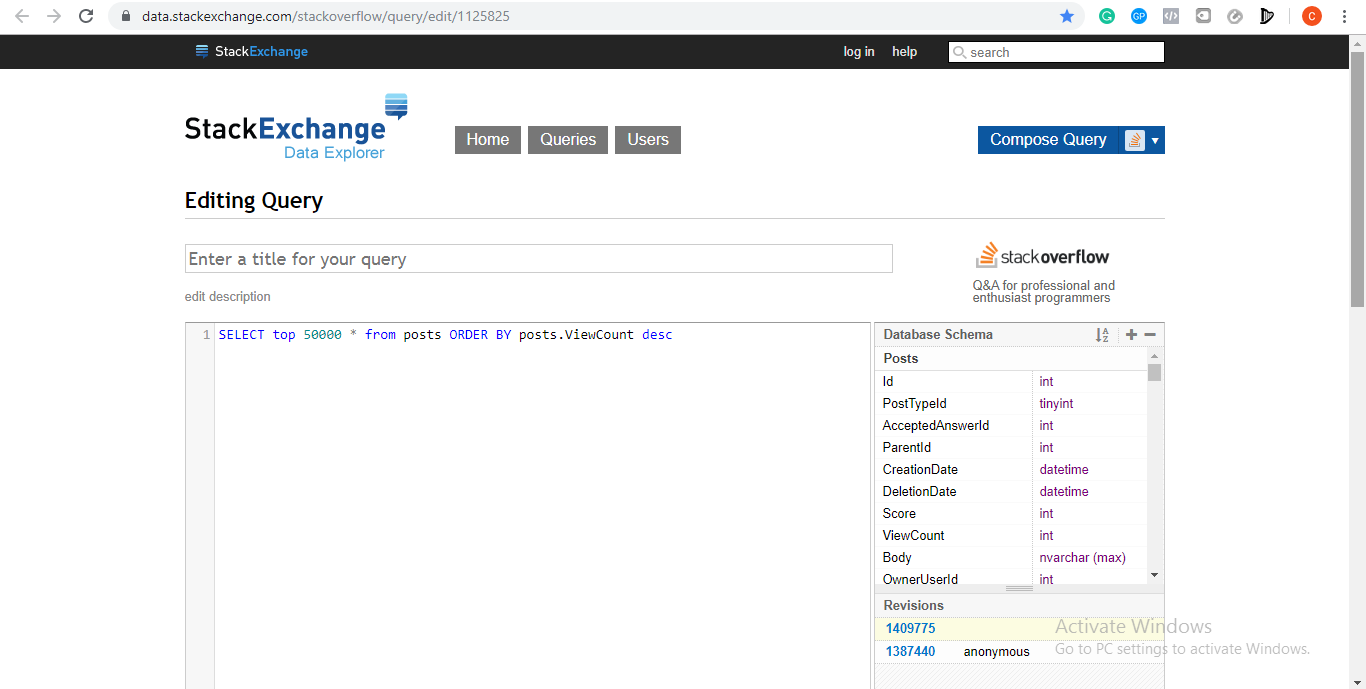
1. **Get Data from Stack Exchange**

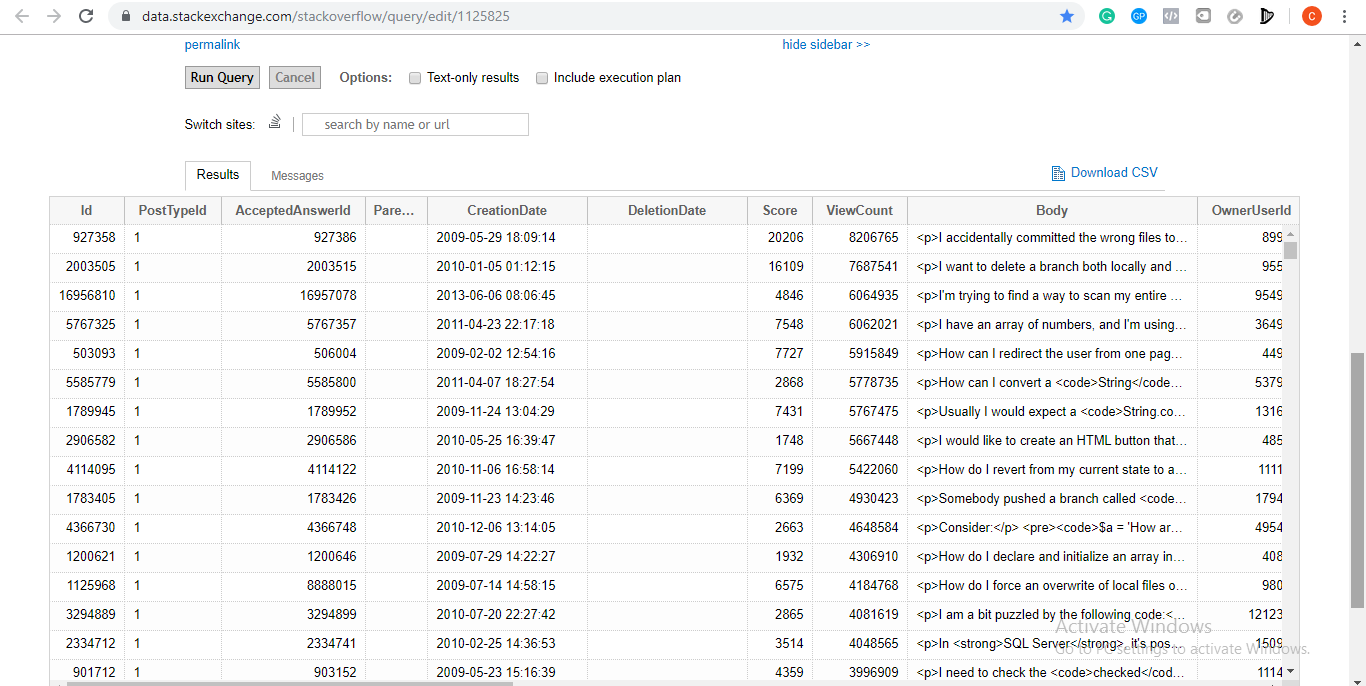
Run Query on StackExchange:

[https://data.stackexchange.com/stackoverflow/query/edit/112 5825](https://data.stackexchange.com/stackoverflow/query/edit/1125825)

Query to access top 50000 posts at a time:

* SELECT top 50000 \* from posts ORDER BY posts.ViewCount desc





Query to access next top posts and so on:

* SELECT top 50000 \* from posts where posts.ViewCount<96538 ORDER BY posts.ViewCount desc
* SELECT top 50000 \* from posts where posts.ViewCount<56799 ORDER BY posts.ViewCount desc
* SELECT top 50000 \* from posts where posts.ViewCount<40672 ORDER BY posts.ViewCount desc

Running above queries on StackExchange will retrieve 200000 posts. Download the data in csv(comma separated value) QueryResult(1).csv, QueryResult(2).csv, QueryResult(3).csv, QueryResult(4).csv.

Merged the above four data files into single file ‘Merged.csv’

Command prompt: **copy \*.csv Merged.csv**

1. **Load them with Pig:**

* Data(‘Merged.csv’) is uploaded on Google Cloud Platform(GCP). This was done by creating a cluster and on cloud shell ‘upload file’ option selected.
* Replace new line and format data file by:

**Sed ‘:a;N;$!ba;s/\n//g’Merged.csv>Cleanmerged.csv**

* **Create directory in Hadoop using:**

hadoop fs -mkdir TempStorage

* **Copy Cleanmerged.csv into new directory /TempStorage**

hadoop fs -put Cleanmerged.csv /TempStorage

* Enter -pig on the cloud shell to perform ETL (Extract, Transform, Load) operation.
* **Load the data into new variable in pig using:**

Pigdata = LOAD '/TempStorage’ 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 , nsweCount : int , CommentCount : int , FavoriteCount : int , ClosedDate : chararray , CommunityOwnedDate : chararray);

* **Remove newline characters from the data and store the data in a variable ‘CleanData’**

CleanData = FOREACH PigData GENERATE id, score, viewcount, REPLACE (REPLACE(body,'\n',''),',','') AS body,OwnerUseId, OwnerDisplayName, Title, Tags;

* **Clean data by removing unwanted brackets by:**

cleaneddata1 = FOREACH cleaneddata GENERATE FLATTEN((Id, Score, ViewCount, Body, OwnerUserId, OwnerDisplayName, Title, Tags));

* **Store cleaned data into a new file: FinalData**

STORE cleaneddata1 INTO 'hdfs://cluster-646b-m/FinalData' USING PigStorage (',');

* **Get FinalData into HDFS by:**

hadoop fs -get /FinalData

FinalData

part-m-00000 part-m-00001

* FinalData has two files: part-m-00000 and part-m-00001. Load these two files into different csv files.
* mv FinalData/part-m-00000 finaldata1.csv
* mv FinalData/part-m-00001 finaldata2.csv
* **From Hive create new Database:**

Create database stackexchangedatabase;

* Use the above database:

Use stackexchangedatabase;

* **Create table to store the cleaned data:**

CREATE TABLE data\_store(Id int, Score bigint, ViewCount bigint, Body string, OwnerUserId string, OwnerDisplayName string, Title string, Tags string) ROW FORMAT DELIMITED FIELS TERMINTED BY ‘,’ LINES TERMINATED BY ‘\n’;

* **Load data into new table:**

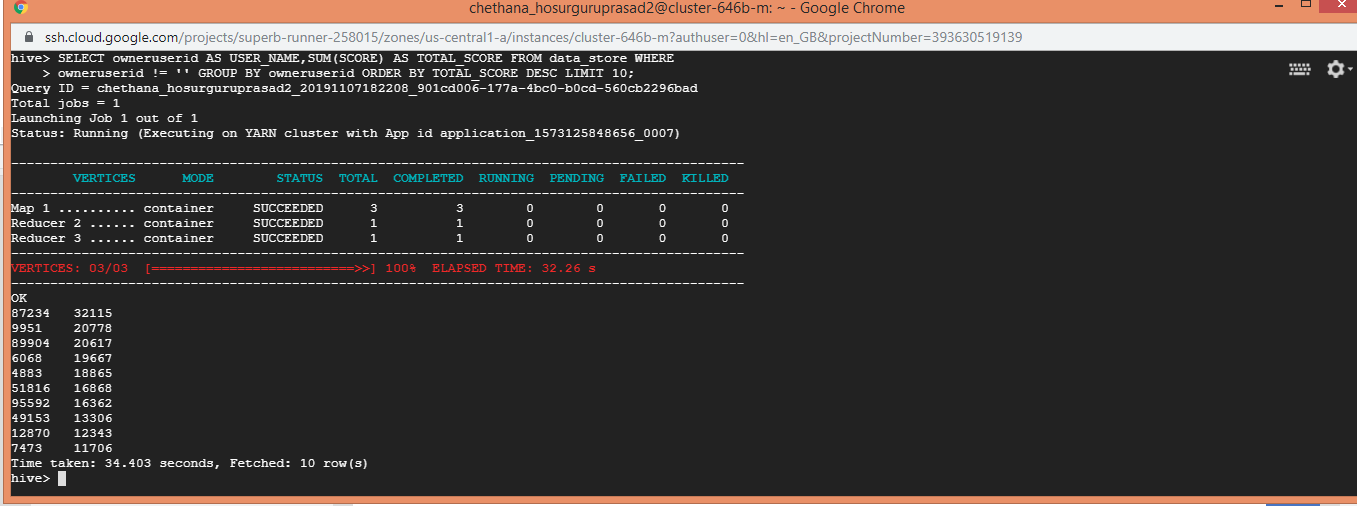
LOAD DATA LOCAL INPATH ‘/home/chethana\_hosurguruprasad2/finaldata1.csv’ INTO TABLE data\_store;

LOAD DATA LOCAL INPATH ‘/home/chethana\_hosurguruprasad2/finaldata2.csv’ INTO TABLE data\_store;

1. **Query Data:**

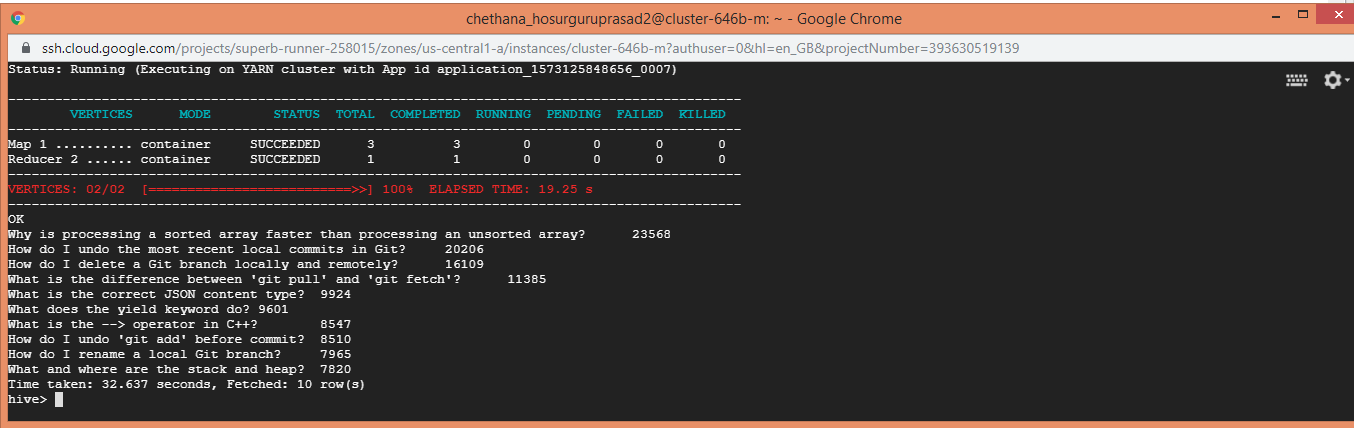
* **Top 10 Users by post score:**

SELECT OwnerUserId AS User\_Name, SUM(Score), AS Total\_Score FROM data\_store WHERE OwnerUserId!=”GROUP BY OwnereUserId ODER BY Total\_Score DESC LIMIT 10;

****

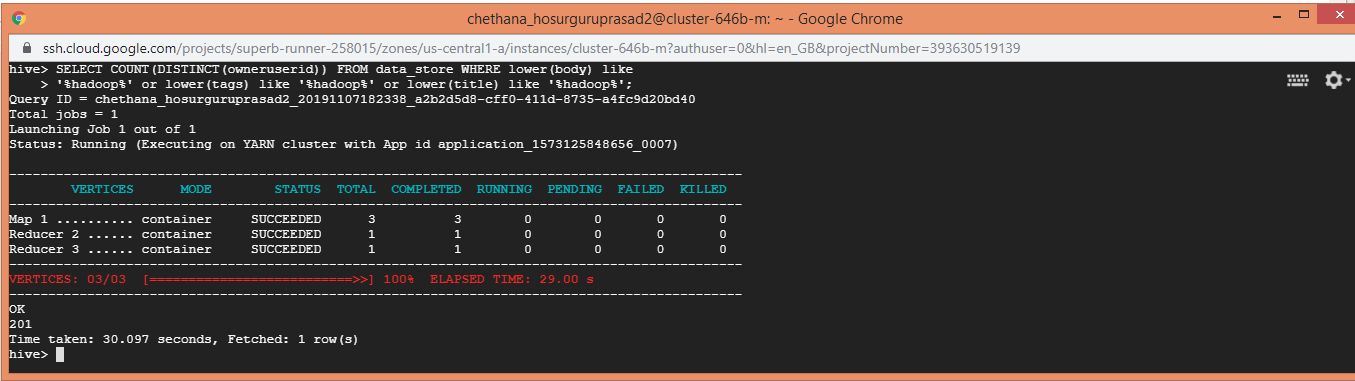
* **Top 10 post by score:**

SELECT Title, Score FROM data\_store ORDER BY score DESC LIMIT 10;

****

* **Number of distinct users, who used the word Hadoop in one of their posts:**

SELECT COUNT (DISTINCT(OwnerUserId)) FROM data\_store WHERE lower(body) like ‘%hadoop%’ or lower(tags) like ‘%hadoop%’ or lower(title) like ‘%hadoop%’;

****

1. **Calculate TF-IDF with Mapreduce: Using Mapreduce calculate per user TF-IDF**

* Upload mapper and reducer files to the console(map1.py, map2.py, map3.py, reduc1.py, reduve2.py, reduce3.py).
* **Get top 10 posts by top 10 users on Hive:**

CREATE TABLE Postdata AS SELECT owneruserid AS USER\_NAME,SUM (SCORE) AS TOTAL\_SCORE FROM data\_store WHERE owneruserid != '' GROUP BY owneruserid ORDER BY TOTAL\_SCORE DESC LIMIT 10;

* **Get all post from top 10 users:**

SELECT owneruserid, body FROM data\_store WHERE owneruserid in (SELECT USER\_NAME FROM Postdata);

* **Store data into file:**

Insert overwrite local directory ‘/home/chethana\_hosurguruprasad2/tfidf’ row format delimited fields terminated by ‘,’ select OwnerUserId, body from data\_store where OwnerUserId in (SELECT User\_Name from Postdata);

* **Create new directory in Hadoop and place the new file in the new directory:**

hadoop fs -mkdir /inputdatahive;

hadoop fs -put tfidf /inputdatahive

* **Run Mapper1 and Reducer1 on the data and store it into new file:**

Hadoop jar/usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/chethana\_hosurguruprasad2/map1.py /home/chethana\_hosurguruprasad2/reduce1.py -mapper”python map1.py” -reducer “python reduce1.py” -input /input\_data/tfidf -output/FINALOUTPUT1

* **Run Mapper2 and Reducer2 on the data and store it into new file:**

Hadoop jar/usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/chethana\_hosurguruprasad2/map2.py /home/chethana\_hosurguruprasad2/reduce2.py -mapper”python map2.py” -reducer “python reduce2.py” -input /FINALOUTPUT1/part-00000/FINALOUTPUT1/part-00001 -output/FINALOUTPUT2

* **Run Mapper3 and Reducer3 on the data and store it into new file:**

Hadoop jar/usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file /home/chethana\_hosurguruprasad2/map3.py /home/chethana\_hosurguruprasad2/reduce3.py -mapper”python map3.py” -reducer “python reduce3.py” -input /FINALOUTPUT2/part-00000/FINALOUTPUT2/part-00001 -output/FINALOUTPUT3

* **Merge all parts of final output into a single file:**

hadoop fs -getmerge /FINALOUTPUT3/part-00000 /FINALOUTPUT3/part-00001/FINAL3/part-00002/FINAL3/part-00003/F INALOUTPUT3/part-00004/home/chethana\_hosurguruprasad2/Tfidf\_d ata.csv

* **Clean and store data into FINALTFIDF1.csv**

**sed -e’s/\s/,/g’ Tfidf\_data.csv>Tfidf\_data2.csv**

* **On Hive, create new table.**

CREATE external TABLE if not exists TFIDF\_FINAL\_DATA(Term string, Id string, tfidf float)ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘,’;

* After executing mapper and reducer functions, **load generated data into a new table:**

LOAD DATA LOCAL INPATH ‘FINALTFDFI1.csv’ OVERWRITE INTO TABLE TFIDF\_FINAL\_DATA;

* **Remove Unwanted attributes and create a table:**

FINALTF AS (SELECT Term,regexp\_replace(ID,'[^0-9]','') as Id,tfidf from TFIDF\_FINAL\_DATA);

* **Generate output and print TF-IDF of top 10 posts used by each of the top 10 users:**

SELECT \* FROM(SELECT Id, Term, tfidf, ROW\_NUMBER() OVER (PARTITION BY Id ORDER BY tfidf DESC) AS RANK FROM FINAL\_TFIDF) WHERE RANK<=10;

