1. Preparing the data to calculate TF-IDF.

**Required data:** A two column table with the **OwnerUserId** and the **Posts.** Only posts of the top 10 users (Calculated in 3.2 of the assignment) needs to be considered for this.

**Considerations:** To get all the words of used by the user, I concatenated the columns **Title, Body** and **Tags.**

**Reference:** [TF-IDF Term Weighting · Hivemall User Manual (apache.org)](https://hivemall.incubator.apache.org/userguide/ft_engineering/tfidf.html)

**Code used:**

\*\*\*\*\*\*\*\*TERMINAL\*\*\*\*\*\*\*\*\*

hive -e 'SELECT OwnerUserId, SUM(Score) AS TotalScore FROM part3\_posts GROUP BY OwnerUserId ORDER BY TotalScore DESC LIMIT 10;' | sed 's/[\t]/,/g' > /home/siddarth\_patil2/query\_output.csv

hadoop fs -put /home/siddarth\_patil2/query\_output.csv /copied

\*\*\*\*\*\*\*\*HIVE\*\*\*\*\*\*\*\*\*

USE DATABASE part\_three;

create table query\_table (OwnerUserId int, TotalScore int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

LOAD DATA INPATH '/copied/query\_output.csv' INTO TABLE query\_table;

\*\*\*\*\*\*\*\*TERMINAL\*\*\*\*\*\*\*\*\*

hive -e 'SELECT part3\_posts.\* FROM part3\_posts JOIN query\_table ON (part3\_posts.OwnerUserId=query\_table.OwnerUserId);' | sed 's/[\t]/,/g' > /home/siddarth\_patil2/top\_users\_posts.csv

hadoop fs -put /home/siddarth\_patil2/top\_users\_posts.csv /copied

\*\*\*\*\*\*\*\*HIVE\*\*\*\*\*\*\*\*\*

create table topuser\_posts (Title string, Body string, Tags string, Score int, Id int, ViewCount int, OwnerUserId int, OwnerDisplayName string)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

LOAD DATA INPATH '/copied/top\_users\_posts.csv' INTO TABLE topuser\_posts;

\*\*\*\*\*\*\*\*PIG\*\*\*\*\*\*\*\*\*

hadoop fs -put /home/siddarth\_patil2/top\_users\_posts.csv /copied

last = LOAD 'hdfs://cluster-cb25-m/copied/top\_users\_posts.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE') AS (Title:chararray, Body:chararray, Tags:chararray, Score:int, Id:int, ViewCount:int, OwnerUserId:int, OwnerDisplayName:chararray );

post\_concat = foreach last Generate (CONCAT (Title, ' ', Body, ' ', Tags), OwnerUserId);

STORE post\_concat INTO 'hdfs://cluster-cb25-m/pig\_concat' USING PigStorage(',');

hadoop fs -mkdir /pig\_concat\_backup

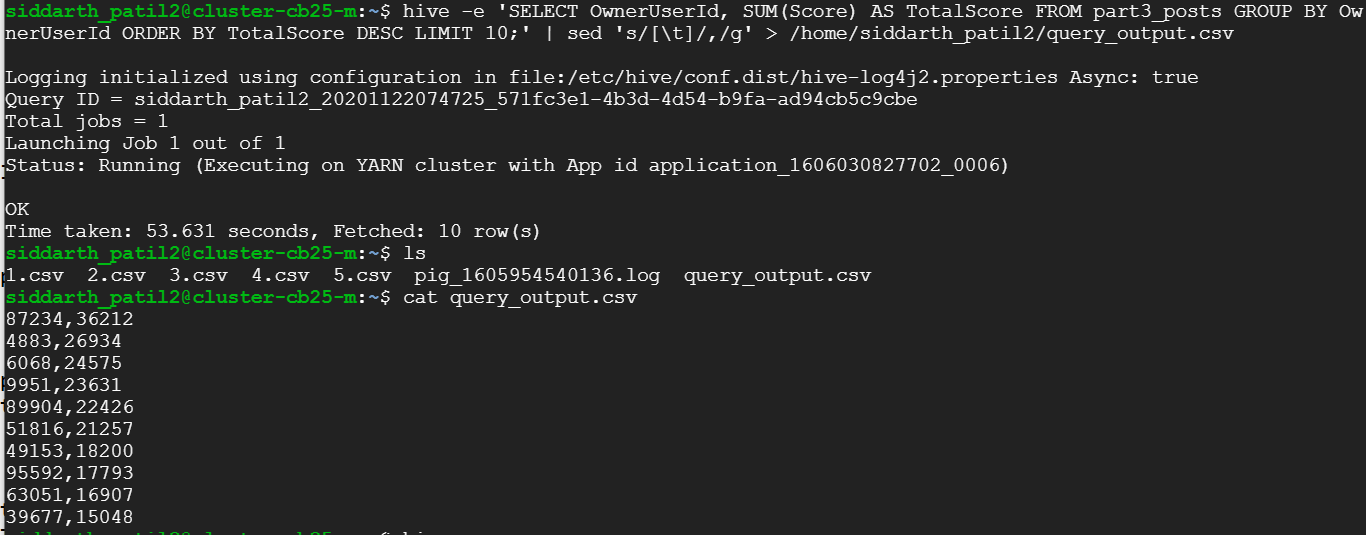
hadoop fs -cp /pig\_con/part-m-00000 /pig\_concat\_back/

\*\*\*\*\*\*\*\*HIVE\*\*\*\*\*\*\*\*\*

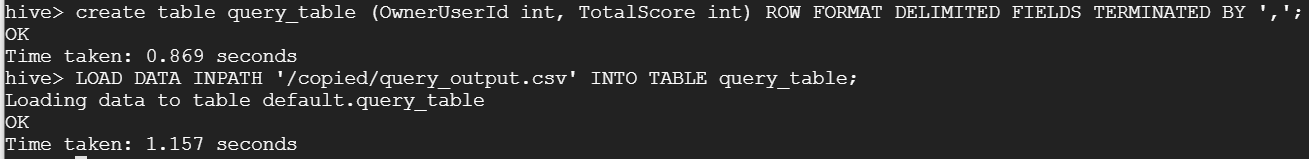
create table final\_table (Posts string, OwnerUserId int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

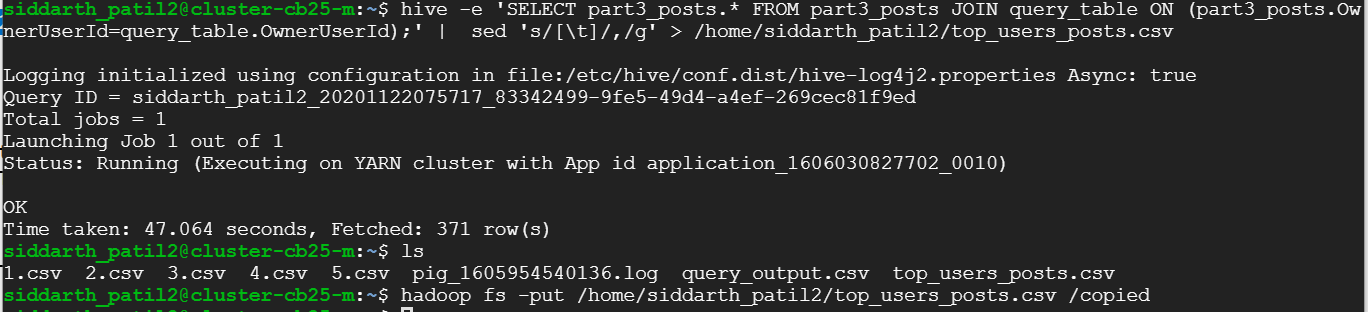
LOAD DATA LOCAL INPATH '/home/siddarth\_patil2/part-m-00000' INTO TABLE final\_table;

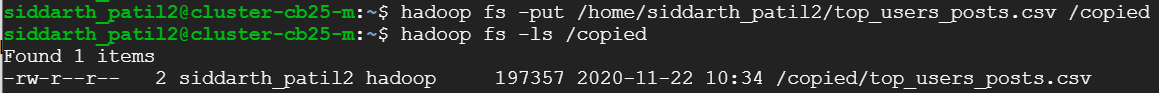
**Screenshots:**

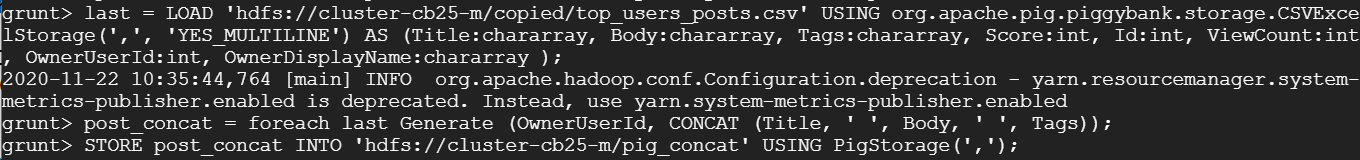


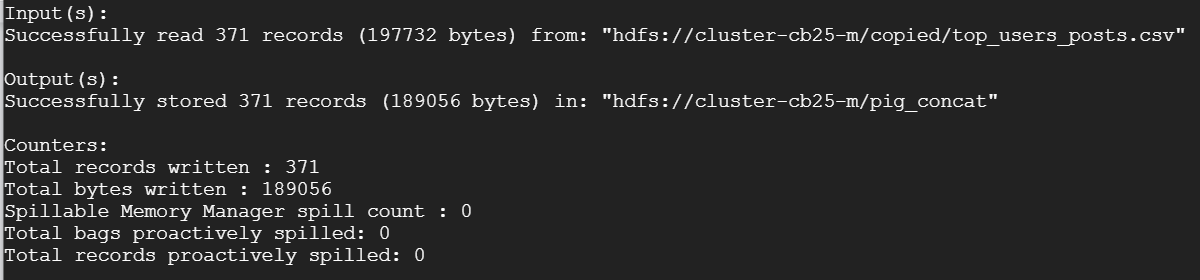
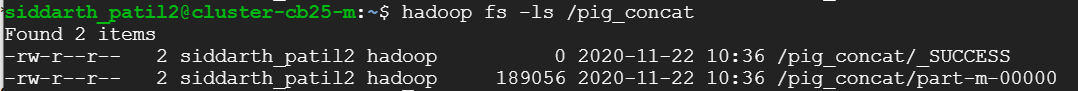


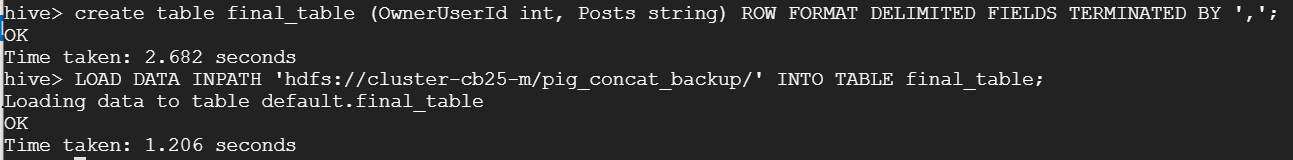








1. Adding the required libraries. First downloading the files from the source given below, then uploading it using the GUI and later making use of these lib after adding them to **tmp**

**Source:** <https://github.com/myui/hivemall/releases>

**Code:**

\*\*\*\*\*\*\*\*TERMINAL\*\*\*\*\*\*\*\*\*

cp /home/siddarth\_patil2/hivemall-core-0.4.2-rc.2-with-dependencies.jar /tmp/

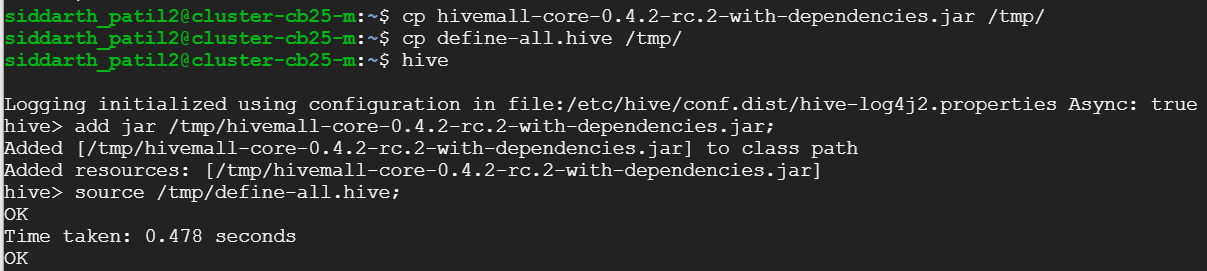
cp /home/siddarth\_patil2/define-all.hive /tmp/

\*\*\*\*\*\*\*\*HIVE\*\*\*\*\*\*\*\*\*

add jar /tmp/hivemall-core-0.4.2-rc.2-with-dependencies.jar;

source /tmp/define-all.hive;

**Screenshot:**



1. Final step is to calculate the TF-IDF in hive.

**Source:** <https://github.com/myui/hivemall/releases>

**Code**: \*\*\*\*\*\*\*\*HIVE\*\*\*\*\*\*\*\*\*

create temporary macro max2(x INT, y INT) if(x>y,x,y);

create temporary macro tfidf(tf FLOAT, df\_t INT, n\_docs INT) tf \* (log(10, CAST(n\_docs as FLOAT)/max2(1,df\_t)) + 1.0);

create or replace view exploded as select ownerUserId, word from final\_table LATERAL VIEW explode(tokenize(Posts, True)) t as word where not is\_stopword(word);

create or replace view term\_frequency as select ownerUserid, word, freq from (select ownerUserId, tf(word) as word2freq from exploded group by ownerUserId) t LATERAL VIEW explode(word2freq) t2 as word, freq;

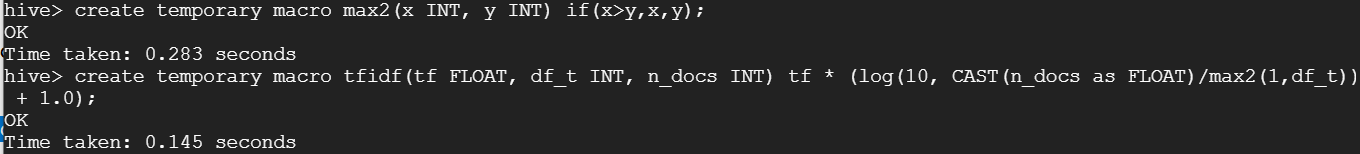
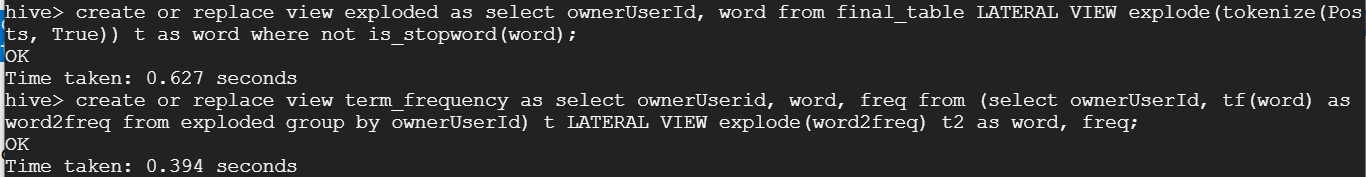
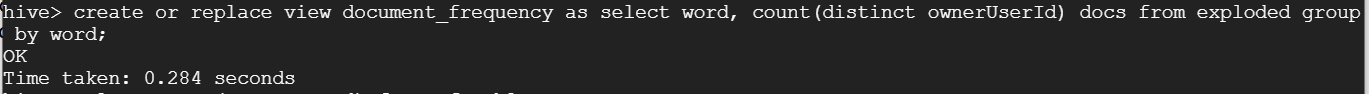
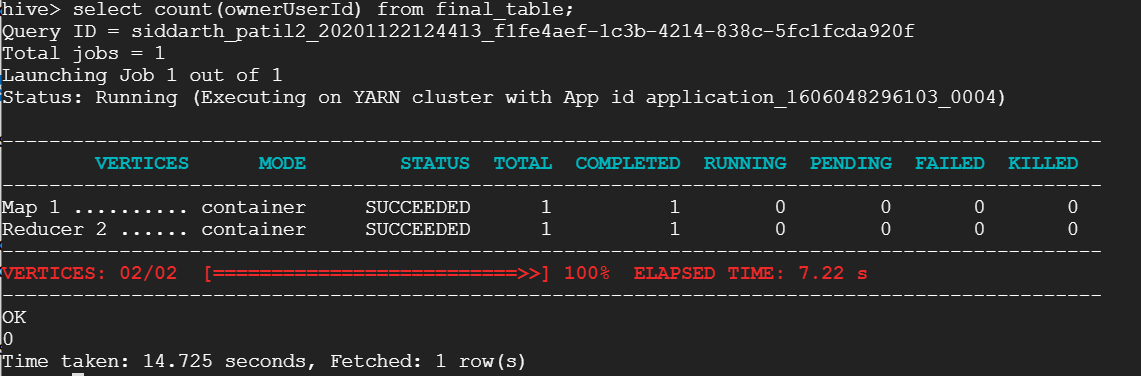
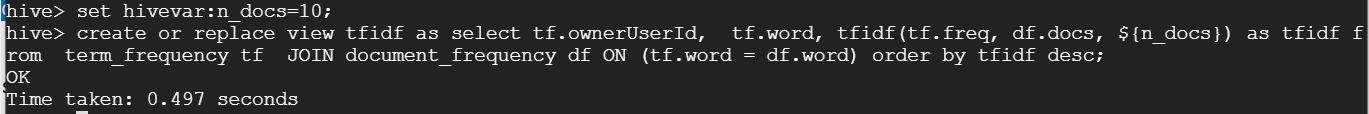
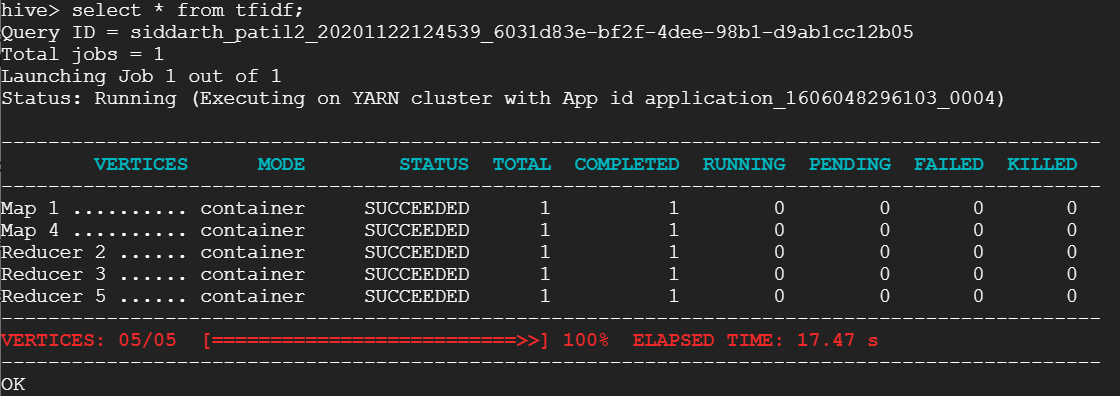
create or replace view document\_frequency as select word, count(distinct ownerUserId) docs from exploded group by word;

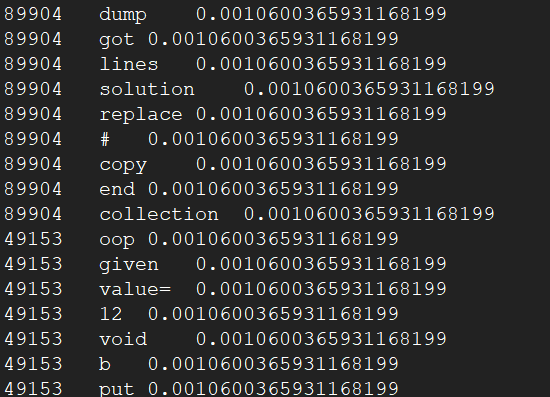
select count(ownerUserId) from final\_table;

set hivevar:n\_docs=10;

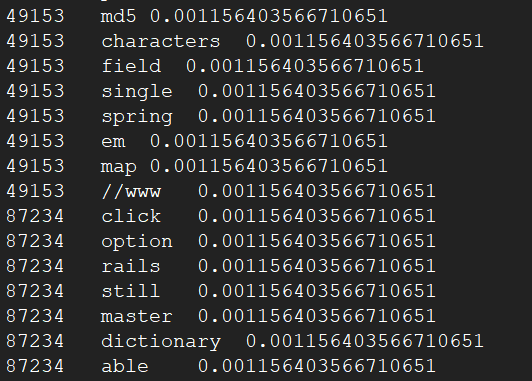
create or replace view tfidf as select tf.ownerUserId, tf.word, tfidf(tf.freq, df.docs, ${n\_docs}) as tfidf from term\_frequency tf JOIN document\_frequency df ON (tf.word = df.word) order by tfidf desc;

select \* from tfidf;

**Screenshots:**      

****

**…**

****