**Cloud Assignment 1**

**1.GET DATA FROM STACK EXCHANGE:**

Performed the following queries to get the 4 CSV files with 50,000 records in each CSV file.

Queries:-

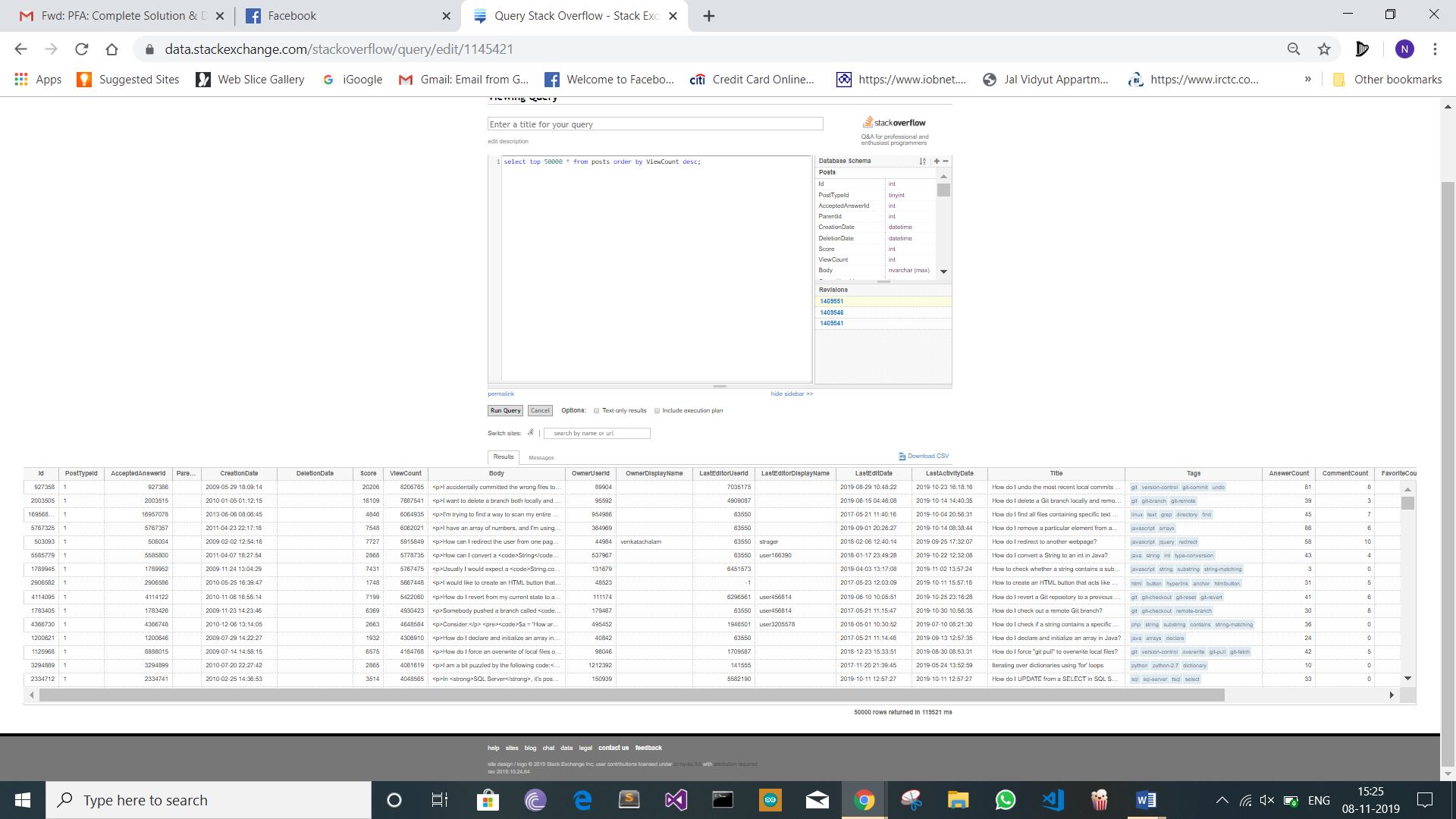
select top 50000 \* from posts where posts.ViewCount > 28534 order by posts.ViewCount DESC

select top 50000 \* FROM posts WHERE posts.ViewCount > 86500 ORDER BY posts.ViewCount DESC

select top 50000 \* from posts order by ViewCount desc;

select top 50000 \* from posts where ViewCount <= 96840 *(last ViewCount of 50000th data)* and id not in ('5117251') order by ViewCount desc

Output:- (More Outputs for the total of 200,000 records are on drive link given at last page that is in Screenshots Folder Under SCREENSHOTS>STACKEXCHANGESQL)



**2.Load Them With PIG:**

After uploading the CSV files on the Google Cloud Platform ,the 4 CSV Files were Loaded and Cleaned through PIG i.e. ETL was PERFORMED. The File was then stored in the CSV Location.

Queries:-

Step 1:-Loading CSV Files Via PIG

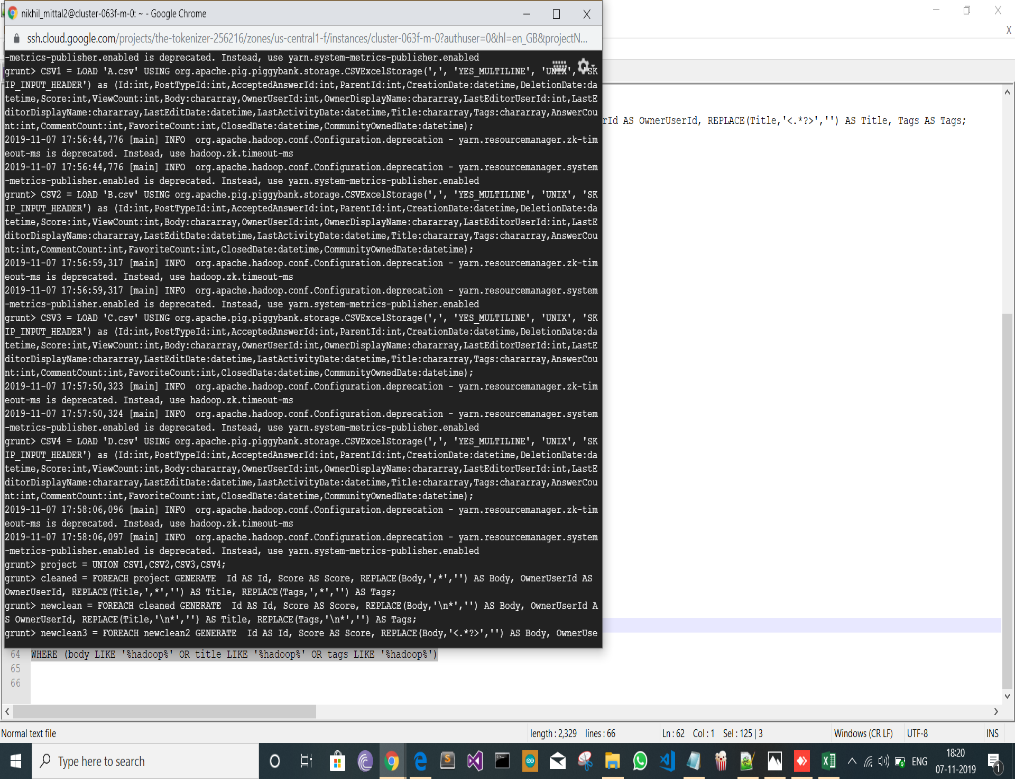
CSV1 = LOAD 'A.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE', 'UNIX', 'SKIP\_INPUT\_HEADER') as (Id:int,PostTypeId:int,AcceptedAnswerId:int,ParentId:int,CreationDate:datetime,DeletionDate:datetime,Score:int,ViewCount:int,Body:chararray,OwnerUserId:int,OwnerDisplayName:chararray,LastEditorUserId:int,LastEditorDisplayName:chararray,LastEditDate:datetime,LastActivityDate:datetime,Title:chararray,Tags:chararray,AnswerCount:int,CommentCount:int,FavoriteCount:int,ClosedDate:datetime,CommunityOwnedDate:datetime);

CSV2 = LOAD 'B.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE', 'UNIX', 'SKIP\_INPUT\_HEADER') as (Id:int,PostTypeId:int,AcceptedAnswerId:int,ParentId:int,CreationDate:datetime,DeletionDate:datetime,Score:int,ViewCount:int,Body:chararray,OwnerUserId:int,OwnerDisplayName:chararray,LastEditorUserId:int,LastEditorDisplayName:chararray,LastEditDate:datetime,LastActivityDate:datetime,Title:chararray,Tags:chararray,AnswerCount:int,CommentCount:int,FavoriteCount:int,ClosedDate:datetime,CommunityOwnedDate:datetime);

CSV3 = LOAD 'C.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE', 'UNIX', 'SKIP\_INPUT\_HEADER') as (Id:int,PostTypeId:int,AcceptedAnswerId:int,ParentId:int,CreationDate:datetime,DeletionDate:datetime,Score:int,ViewCount:int,Body:chararray,OwnerUserId:int,OwnerDisplayName:chararray,LastEditorUserId:int,LastEditorDisplayName:chararray,LastEditDate:datetime,LastActivityDate:datetime,Title:chararray,Tags:chararray,AnswerCount:int,CommentCount:int,FavoriteCount:int,ClosedDate:datetime,CommunityOwnedDate:datetime);

CSV4 = LOAD 'D.csv' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'YES\_MULTILINE', 'UNIX', 'SKIP\_INPUT\_HEADER') as (Id:int,PostTypeId:int,AcceptedAnswerId:int,ParentId:int,CreationDate:datetime,DeletionDate:datetime,Score:int,ViewCount:int,Body:chararray,OwnerUserId:int,OwnerDisplayName:chararray,LastEditorUserId:int,LastEditorDisplayName:chararray,LastEditDate:datetime,LastActivityDate:datetime,Title:chararray,Tags:chararray,AnswerCount:int,CommentCount:int,FavoriteCount:int,ClosedDate:datetime,CommunityOwnedDate:datetime);

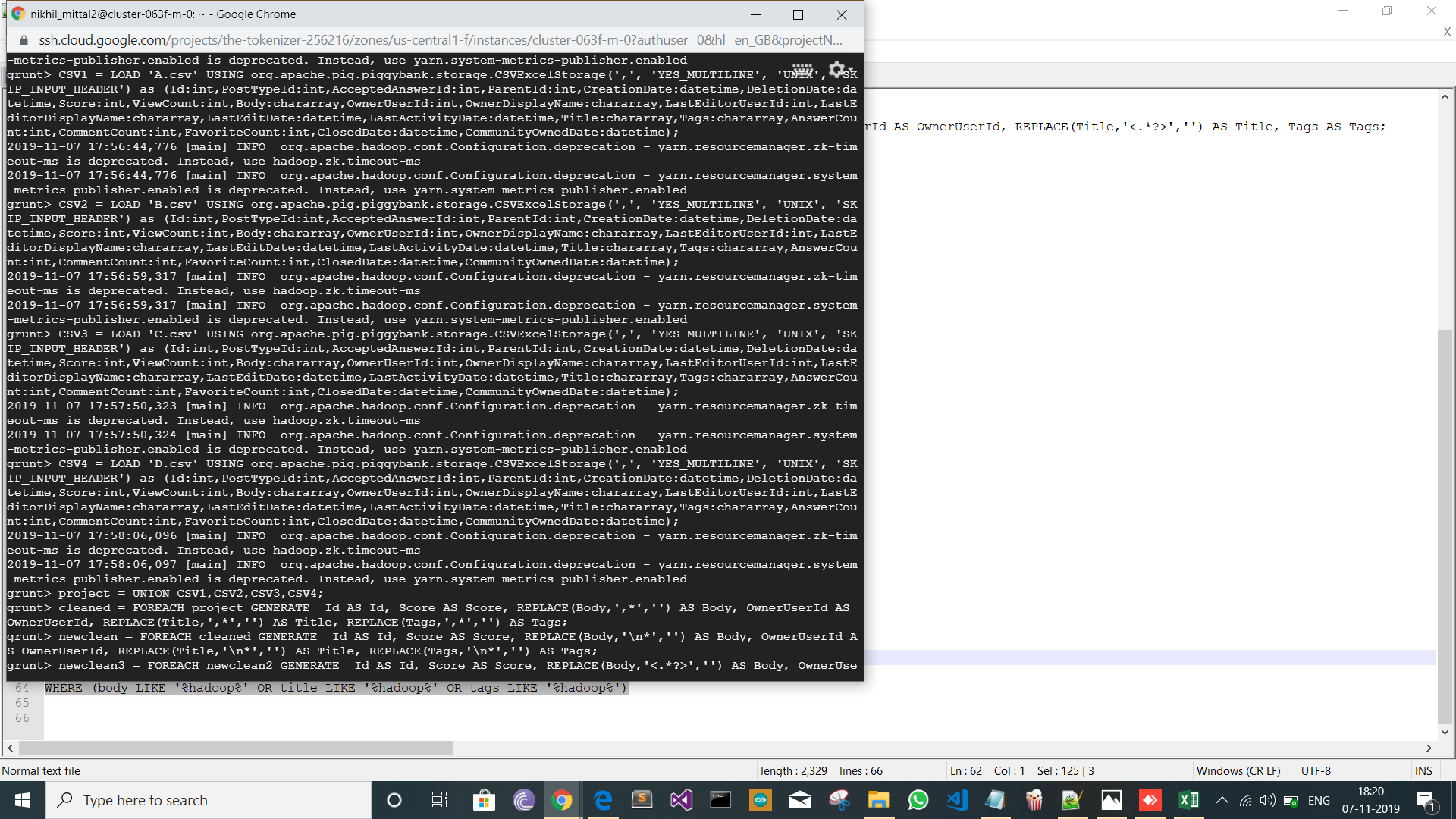
Output:-



Step 2:- Merging the files into One

project = UNION CSV1,CSV2,CSV3,CSV4;

OUTPUT:-



Step 3:- Cleaning the Merged File

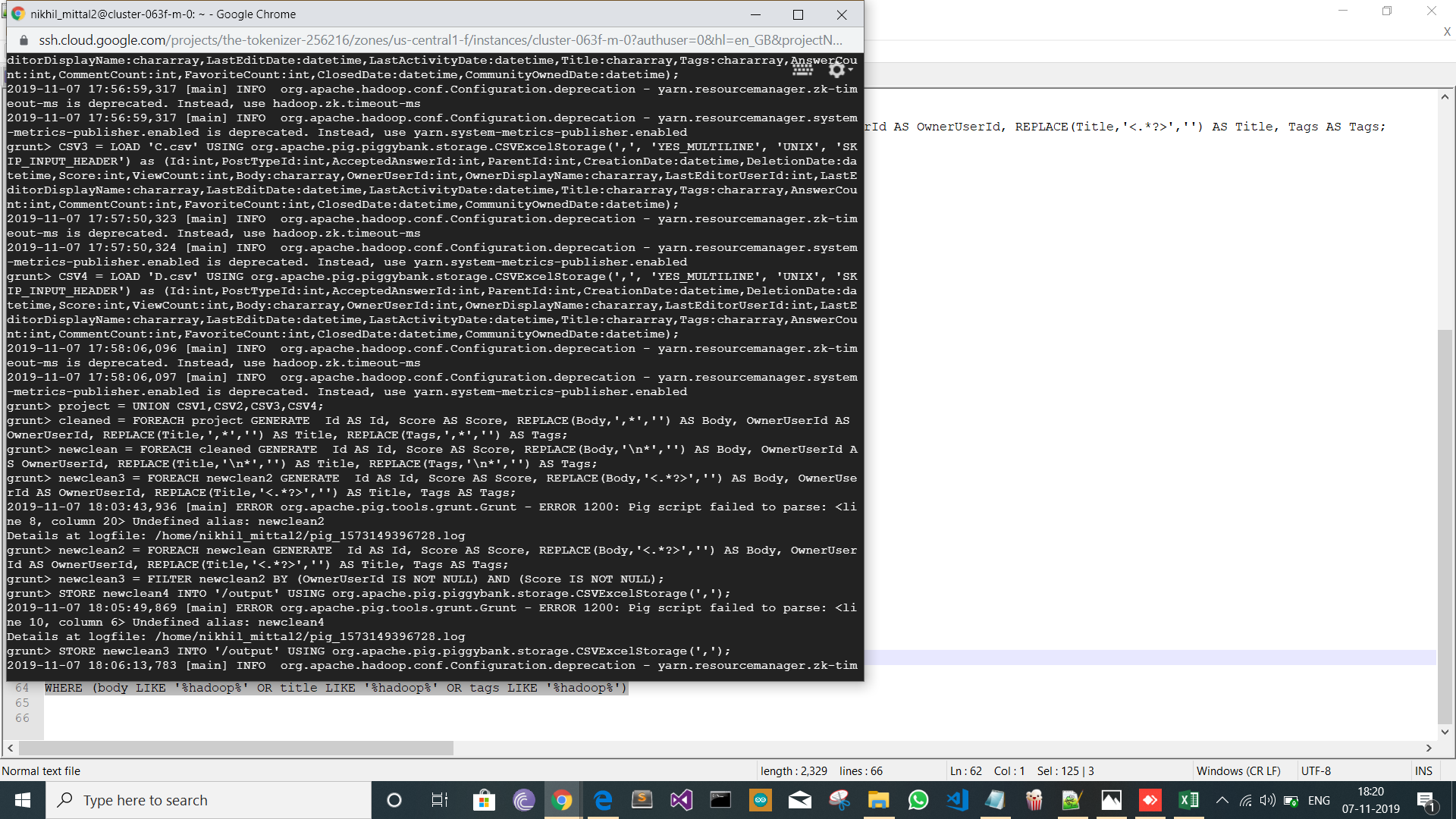
cleaned = FOREACH project GENERATE Id AS Id, Score AS Score, REPLACE(Body,',\*','') AS Body, OwnerUserId AS OwnerUserId, REPLACE(Title,',\*','') AS Title, REPLACE(Tags,',\*','') AS Tags;

newclean = FOREACH cleaned GENERATE Id AS Id, Score AS Score, REPLACE(Body,'\n\*','') AS Body, OwnerUserId AS OwnerUserId, REPLACE(Title,'\n\*','') AS Title, REPLACE(Tags,'\n\*','') AS Tags;

newclean2 = FOREACH newclean GENERATE Id AS Id, Score AS Score, REPLACE(Body,'<.\*?>','') AS Body, OwnerUserId AS OwnerUserId, REPLACE(Title,'<.\*?>','') AS Title, Tags AS Tags;

newclean3 = FILTER newclean2 BY (OwnerUserId IS NOT NULL) AND (Score IS NOT NULL);

Output:-

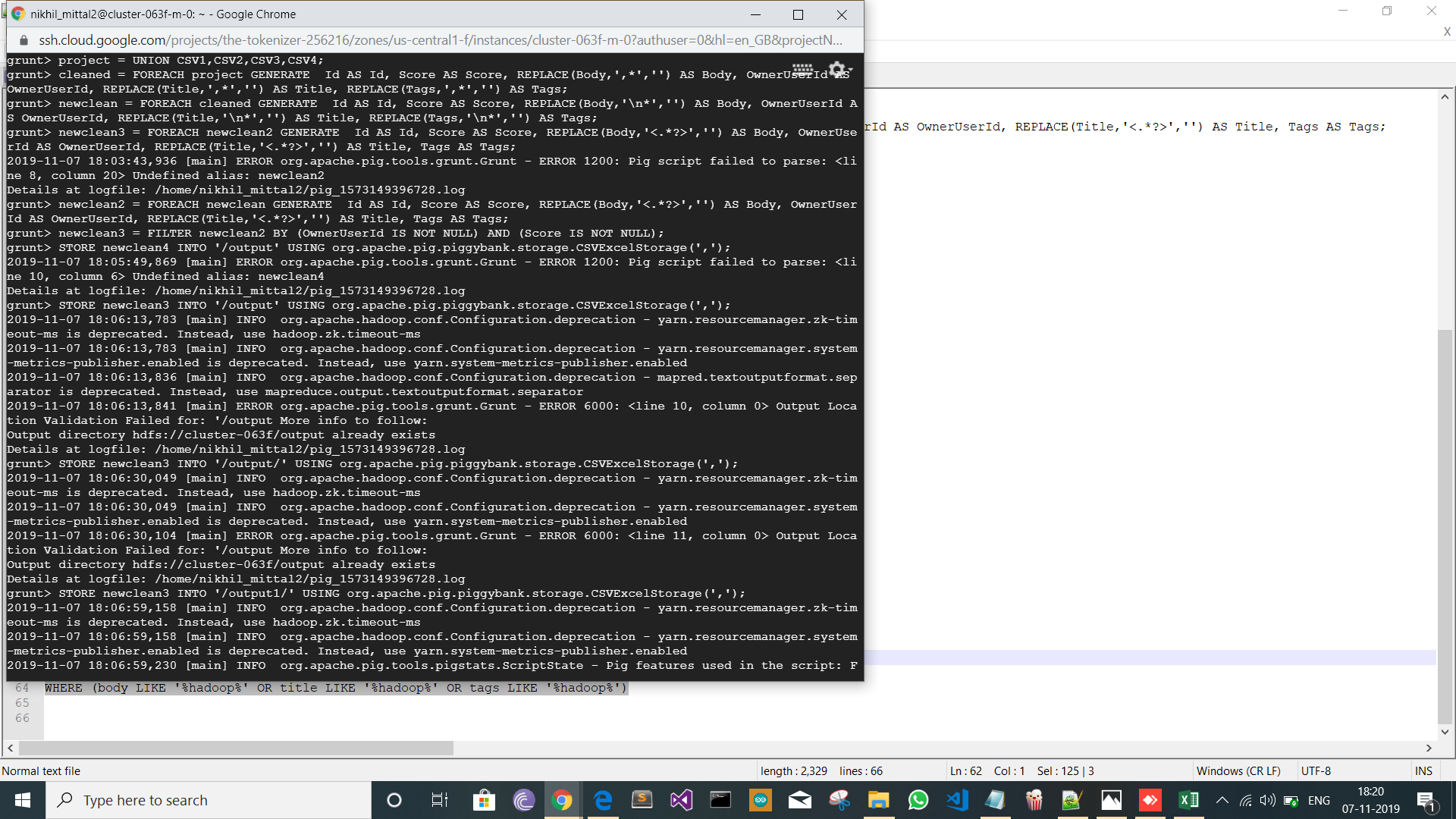


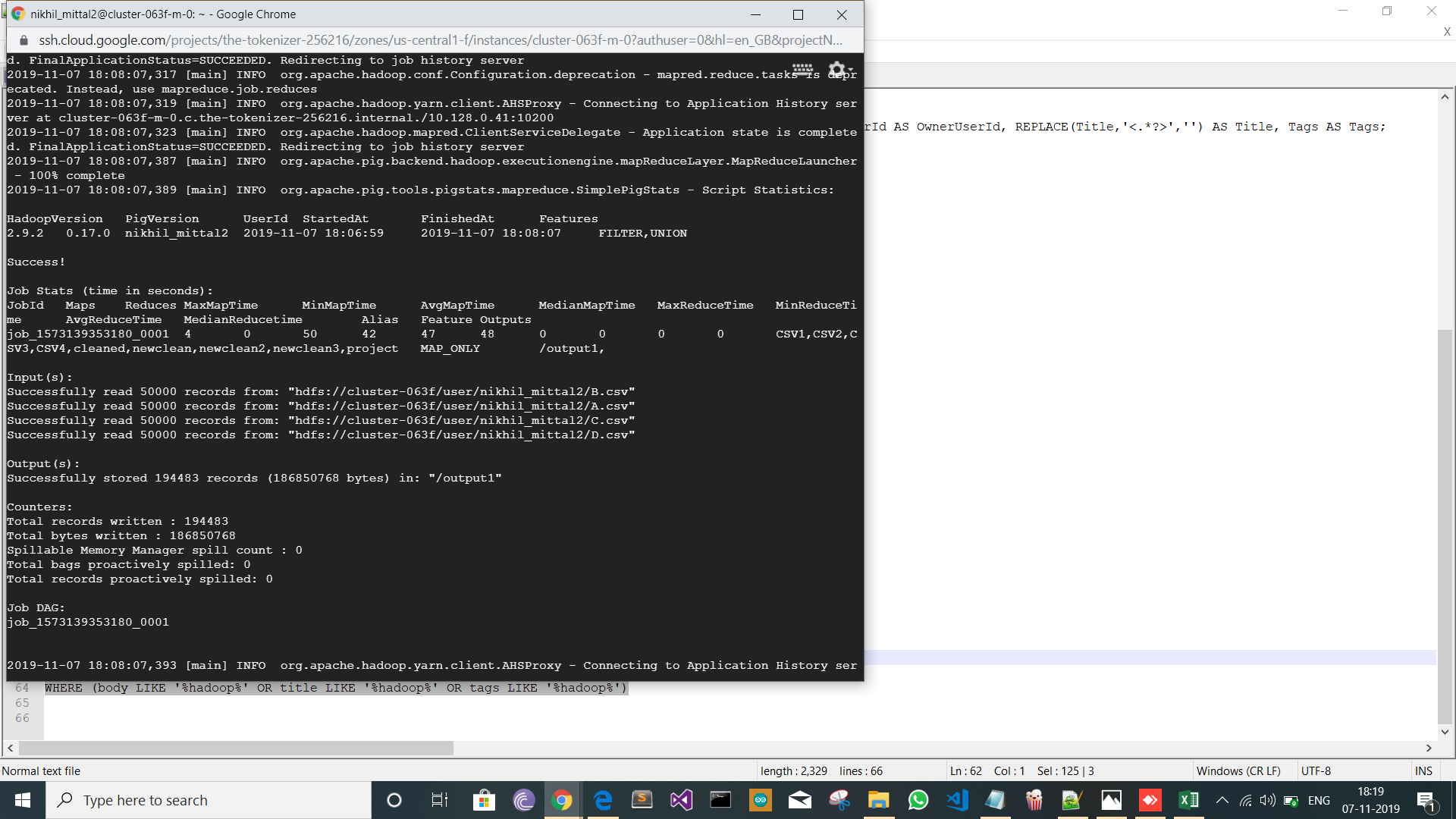
Step 4:- Storing the Cleaned File in the HDFS Location

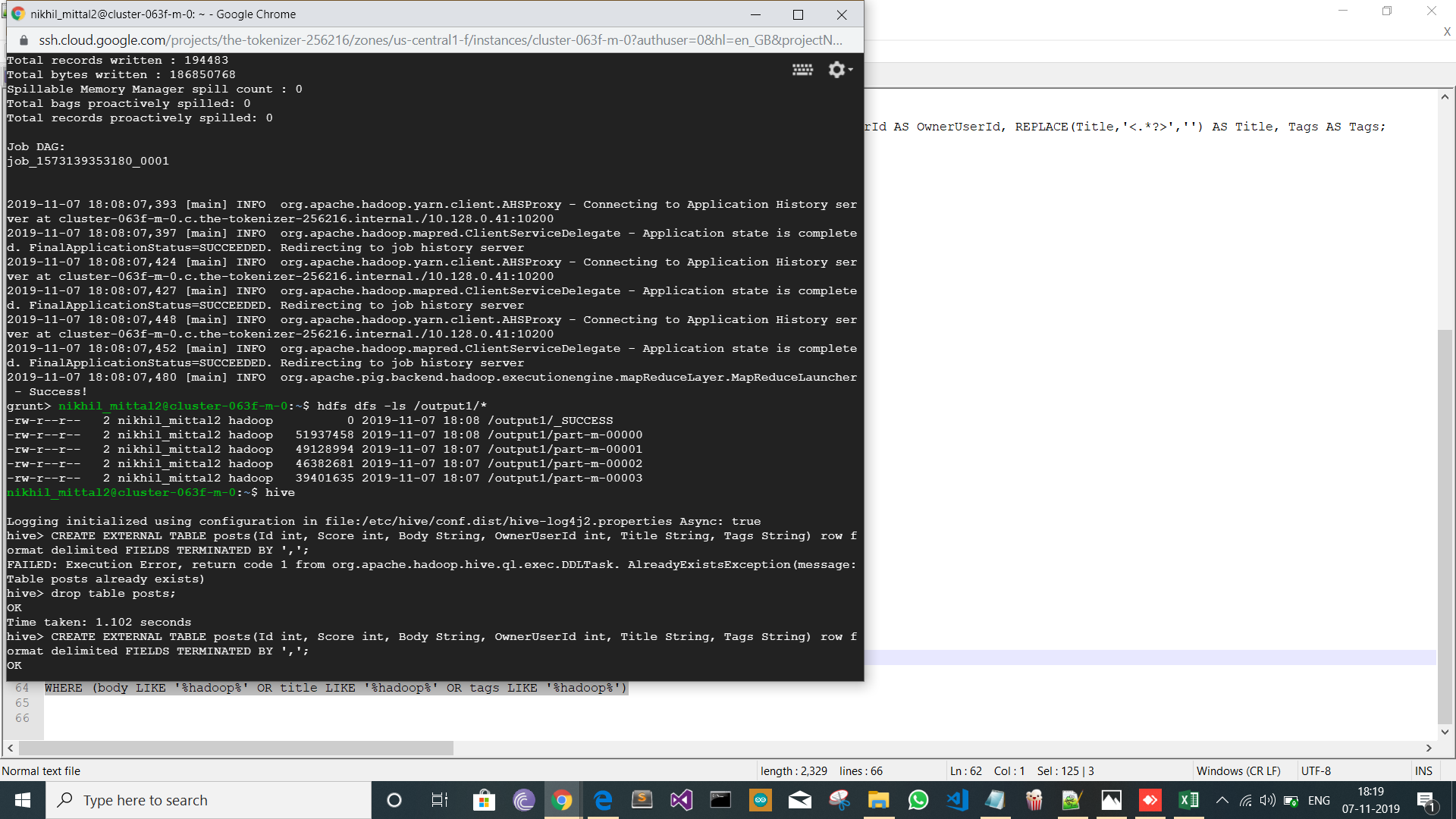
STORE newclean3 INTO '/output1' USING org.apache.pig.piggybank.storage.CSVExcelStorage(',');

Output:-

This is the Output After performing ETL and cleaning through PIG.







**3.Query Them with HIVE:**

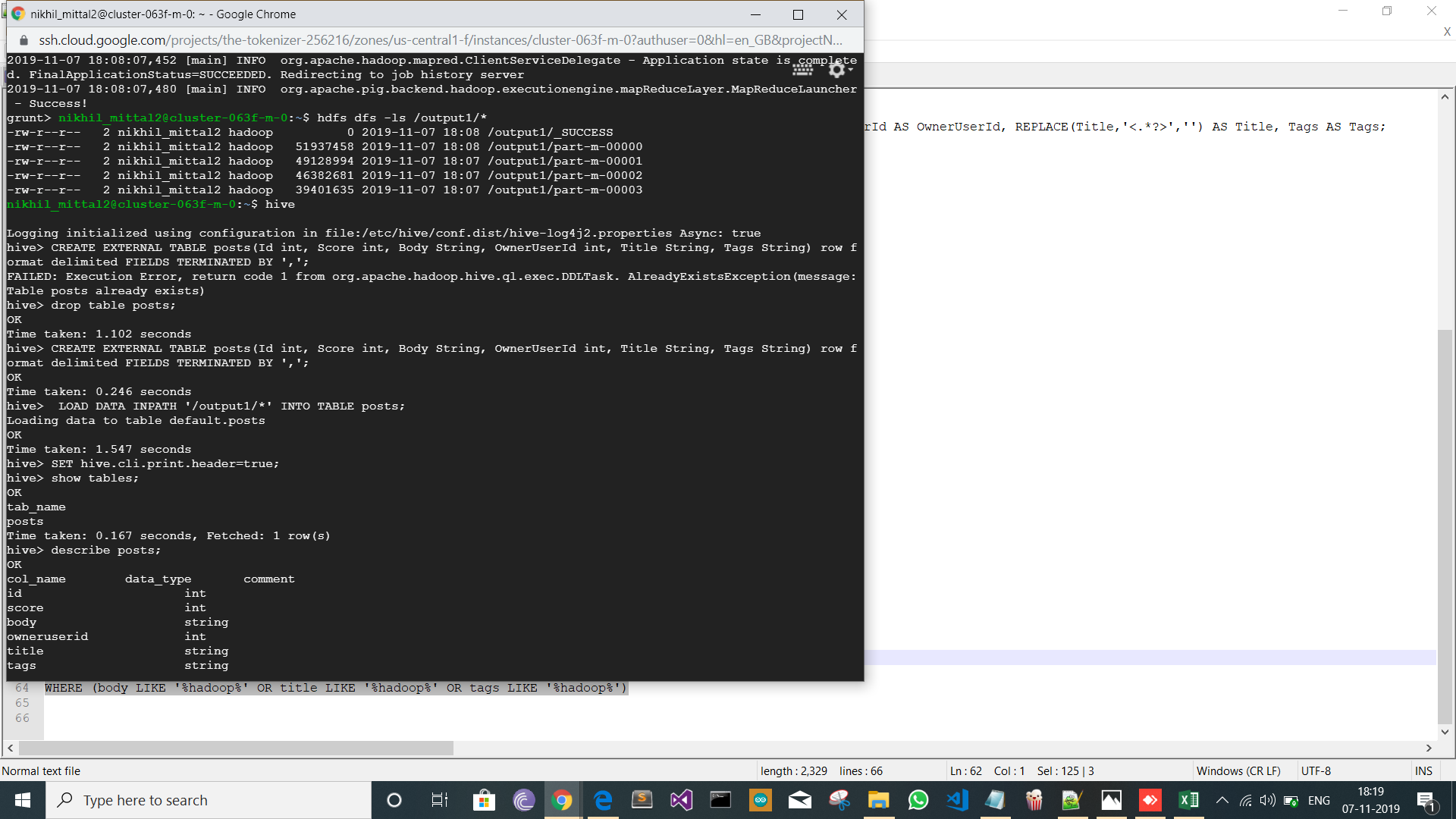
Queries:

create table posts(Id int, Score int, Body String, OwnerUserId Int, Title String, Tags String)

row format delimited

FIELDS TERMINATED BY ','

LOAD DATA INPATH '/output1/\*' INTO TABLE posts;



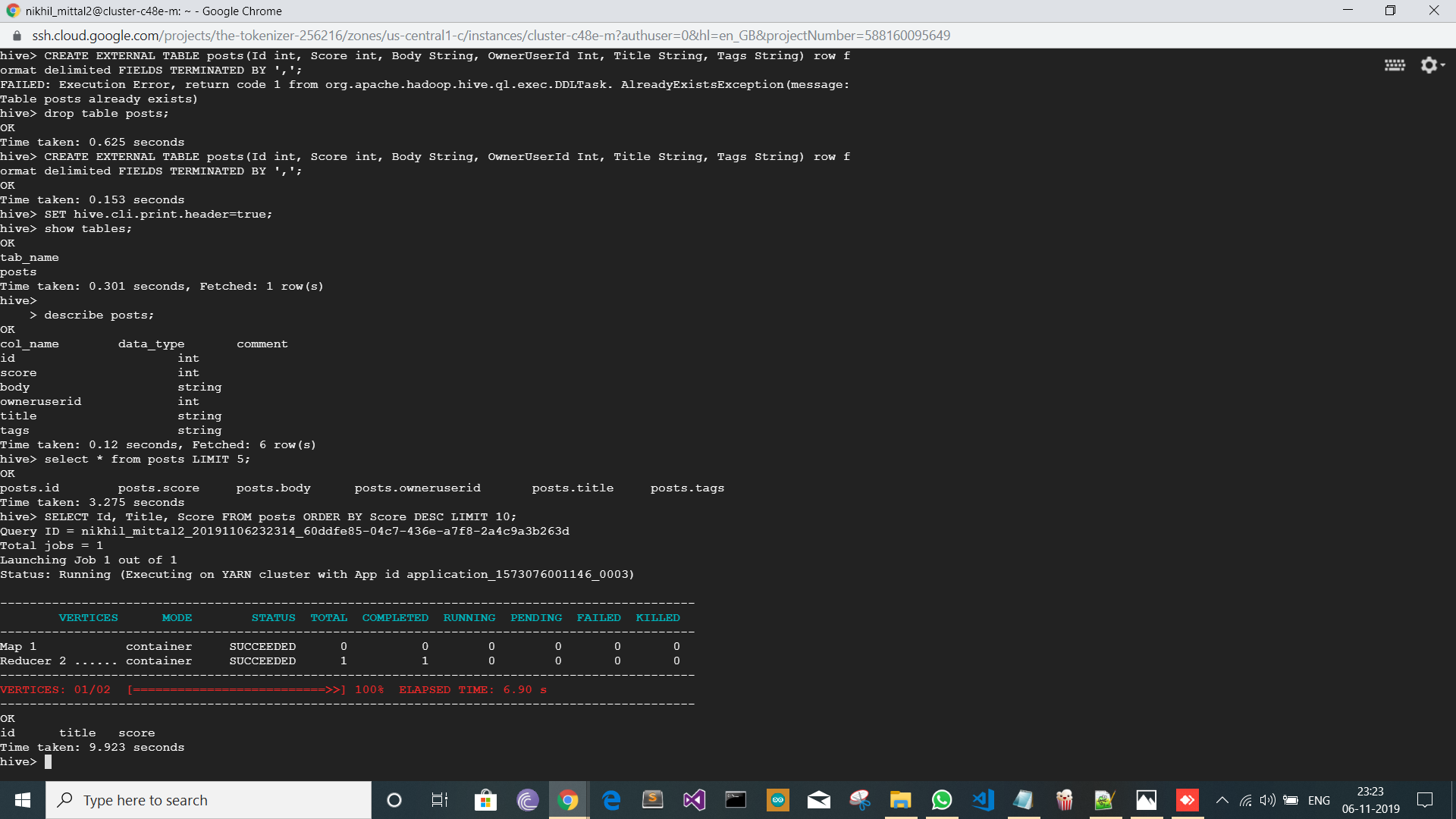
Output:-

a)The Top ten posts by score

Queries:

SELECT Id, Title, Score FROM posts ORDER BY Score DESC LIMIT 10;

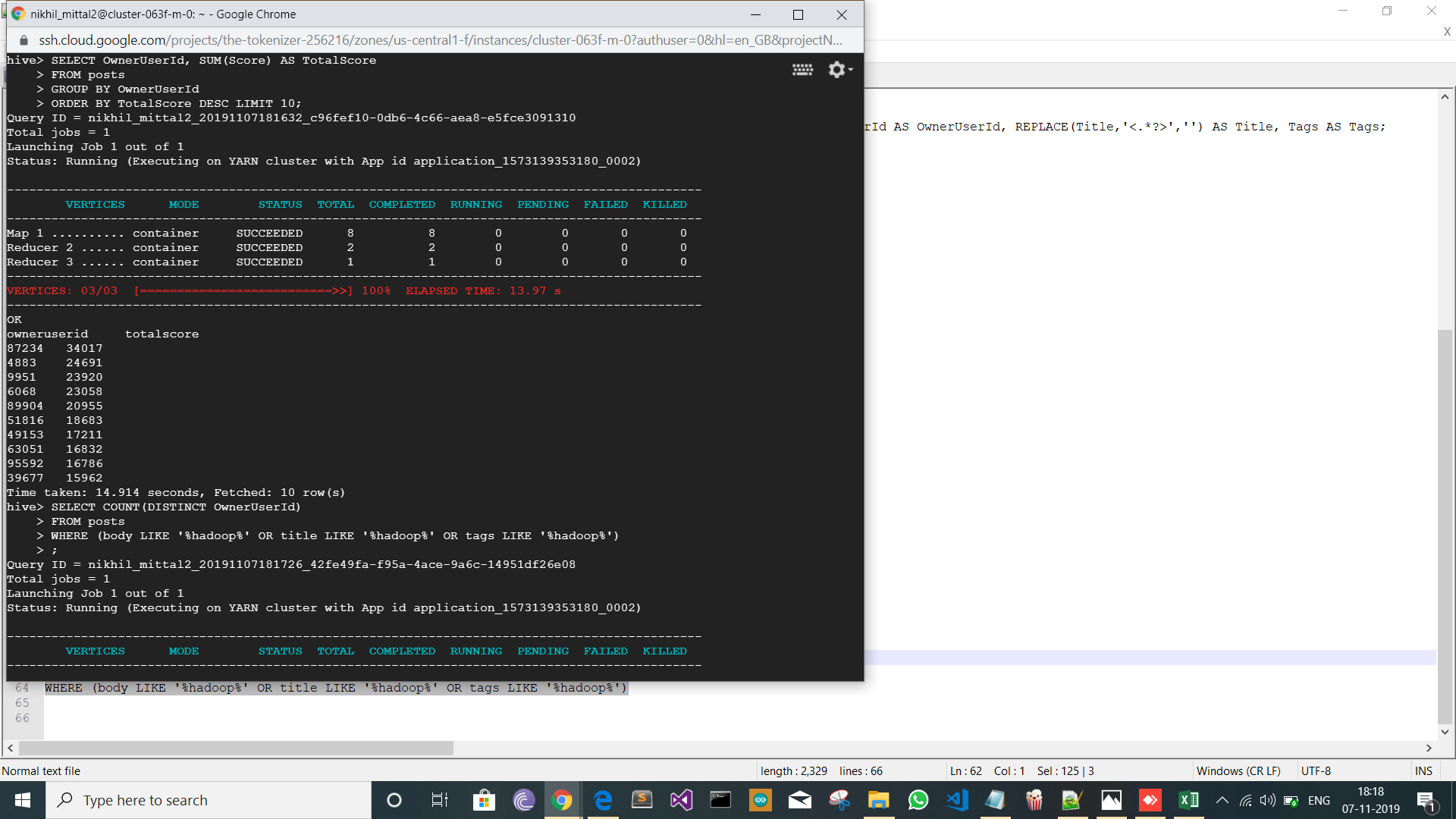
Output:-



b)The Top ten users by post score

Queries:- SELECT OwnerUserId, SUM(Score) AS TotalScore FROM posts GROUP BY OwnerUserId ORDER BY TotalScore DESC LIMIT 10;

Output:-

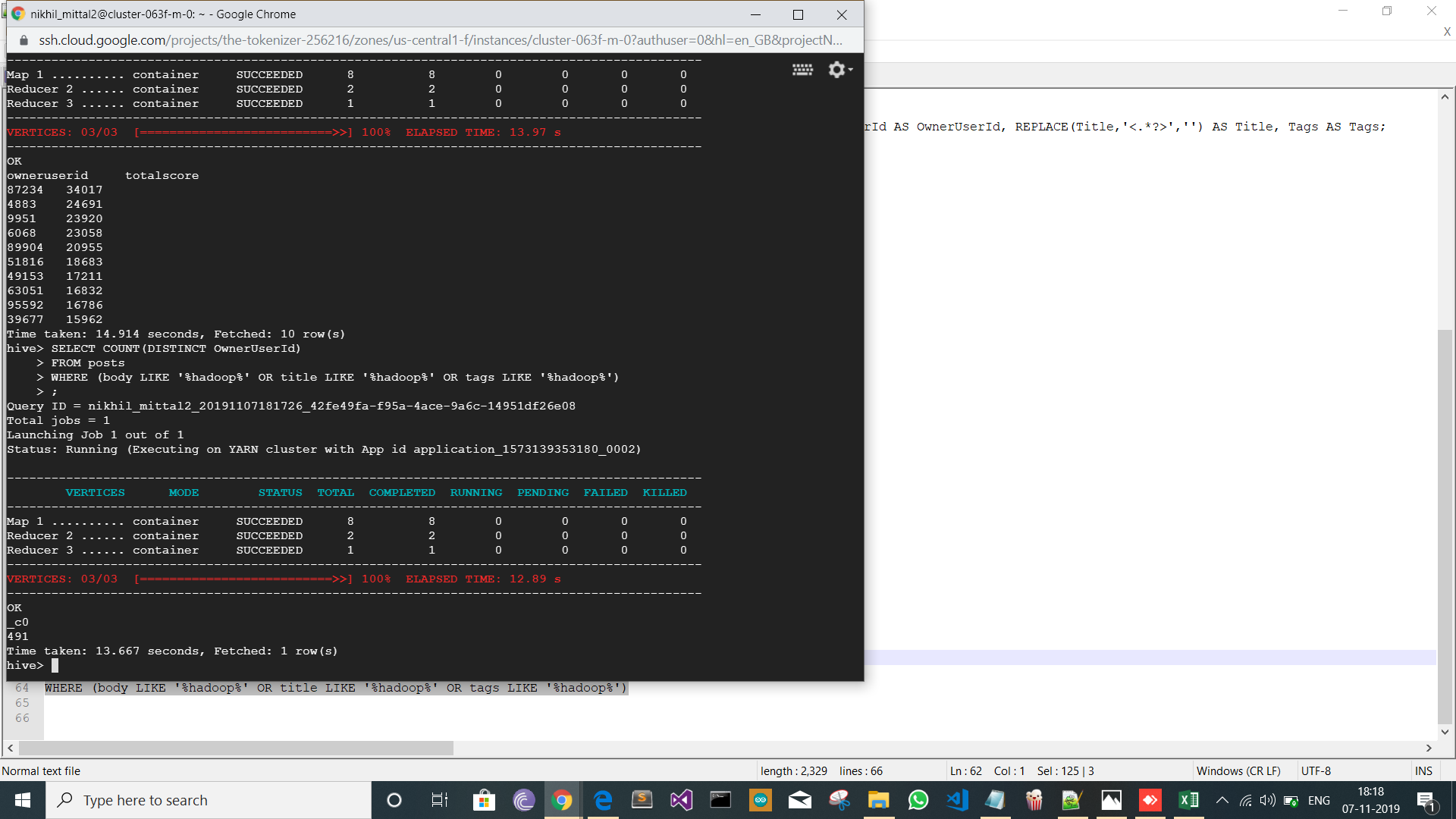


c)The number of distinct users ,who used the word ‘Hadoop’ in one their posts.

Queries:-

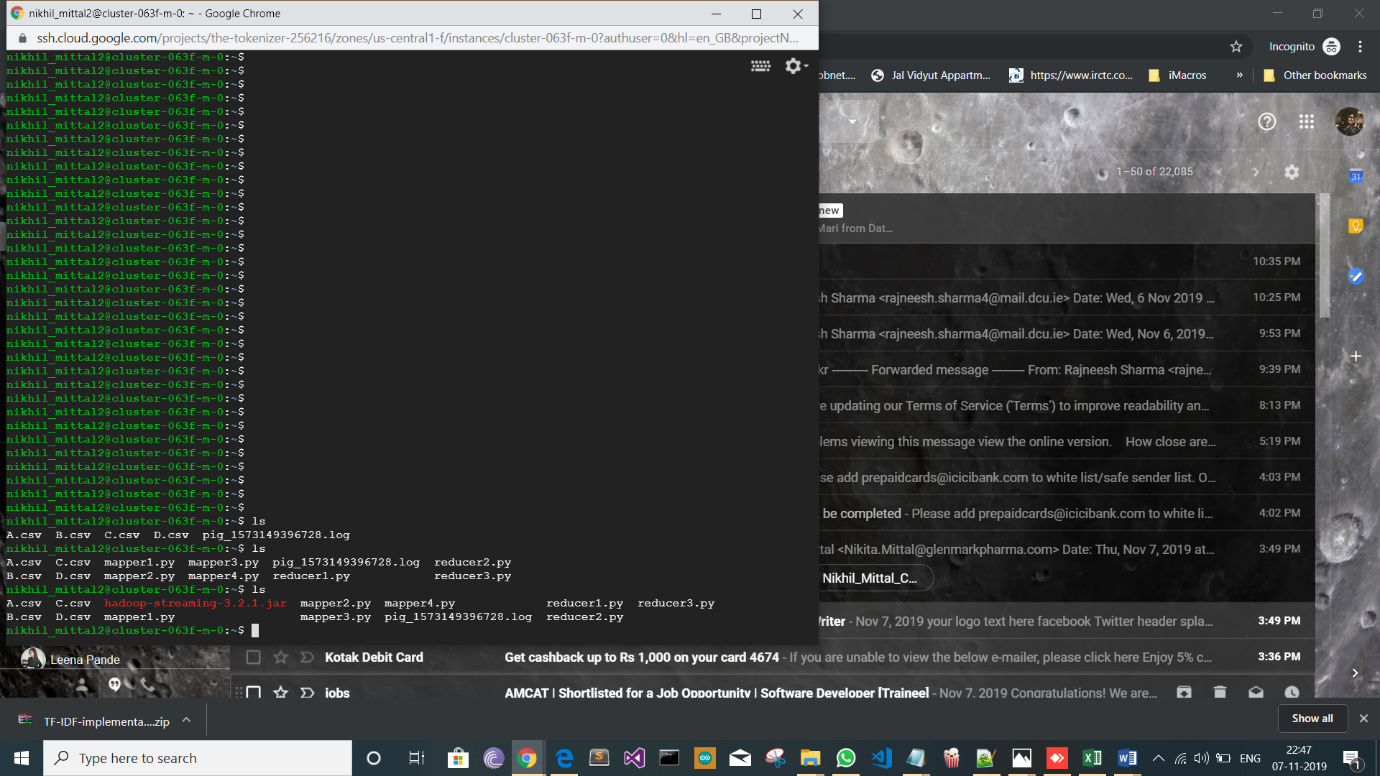
SELECT COUNT(DISTINCT OwnerUserId) FROM posts WHERE (body LIKE '%hadoop%' OR title LIKE '%hadoop%' OR tags LIKE '%hadoop%');

Output:-



**4)Calculate TF-IDF with MapReduce.**

Firstly, four mapper files were loaded, and 3 reducer files were loaded. Then a Hadoop jar file is loaded into the Hadoop cluster.



Hive Query performed

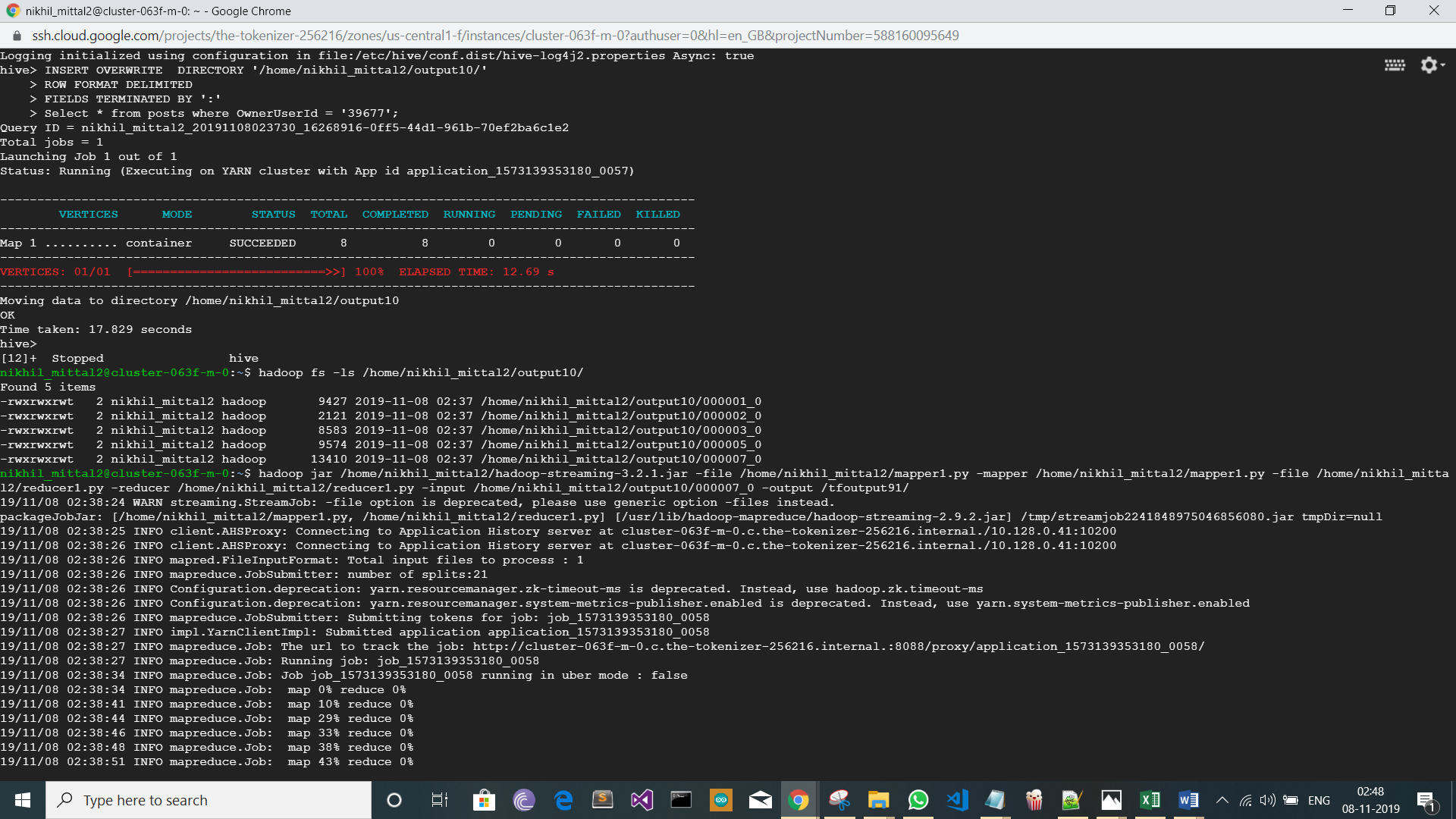
INSERT OVERWRITE DIRECTORY '/home/nikhil\_mittal2/output10/'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ':'

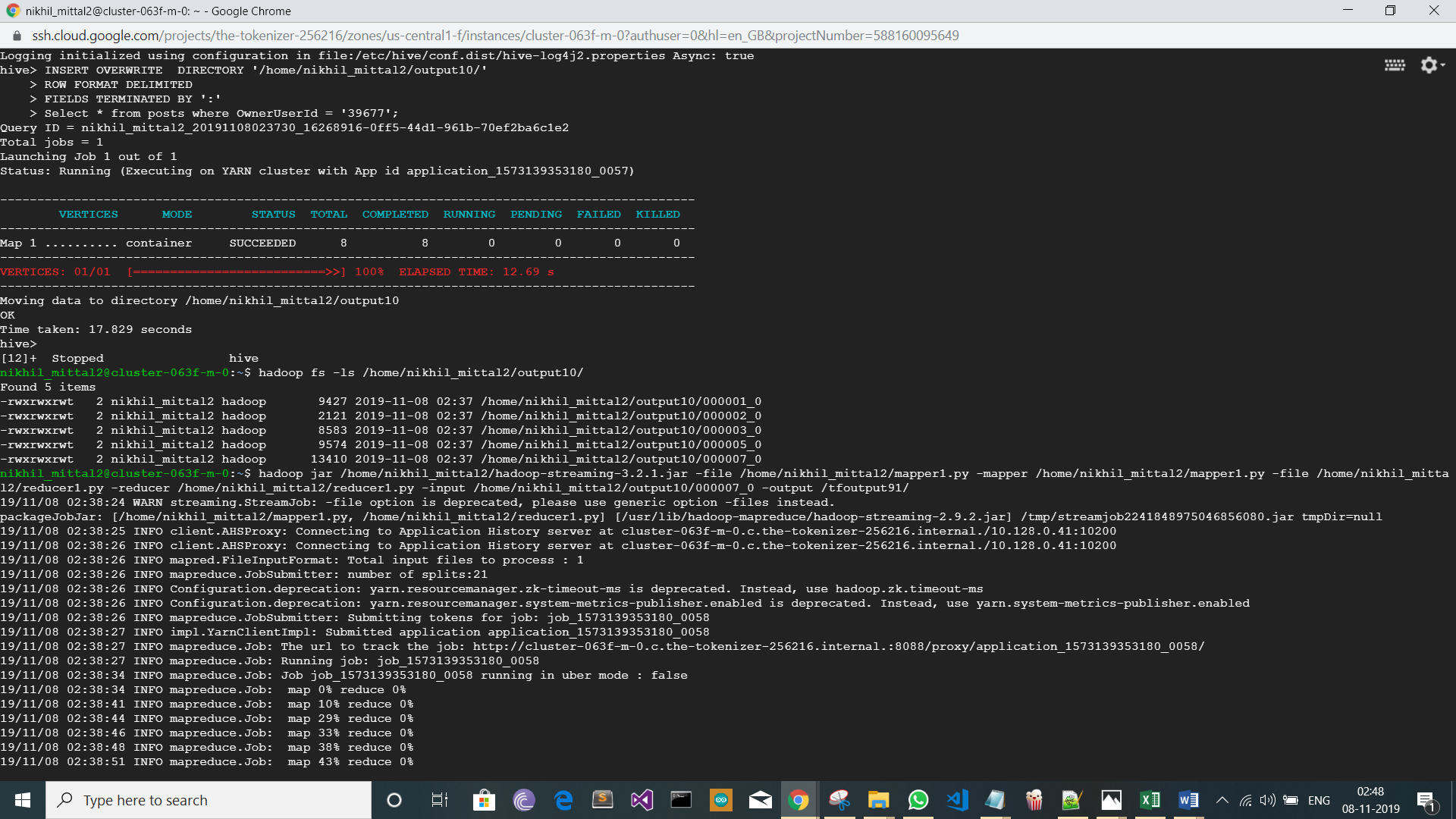
Select \* from posts where OwnerUserId = '39677';

For User ID:- 39677

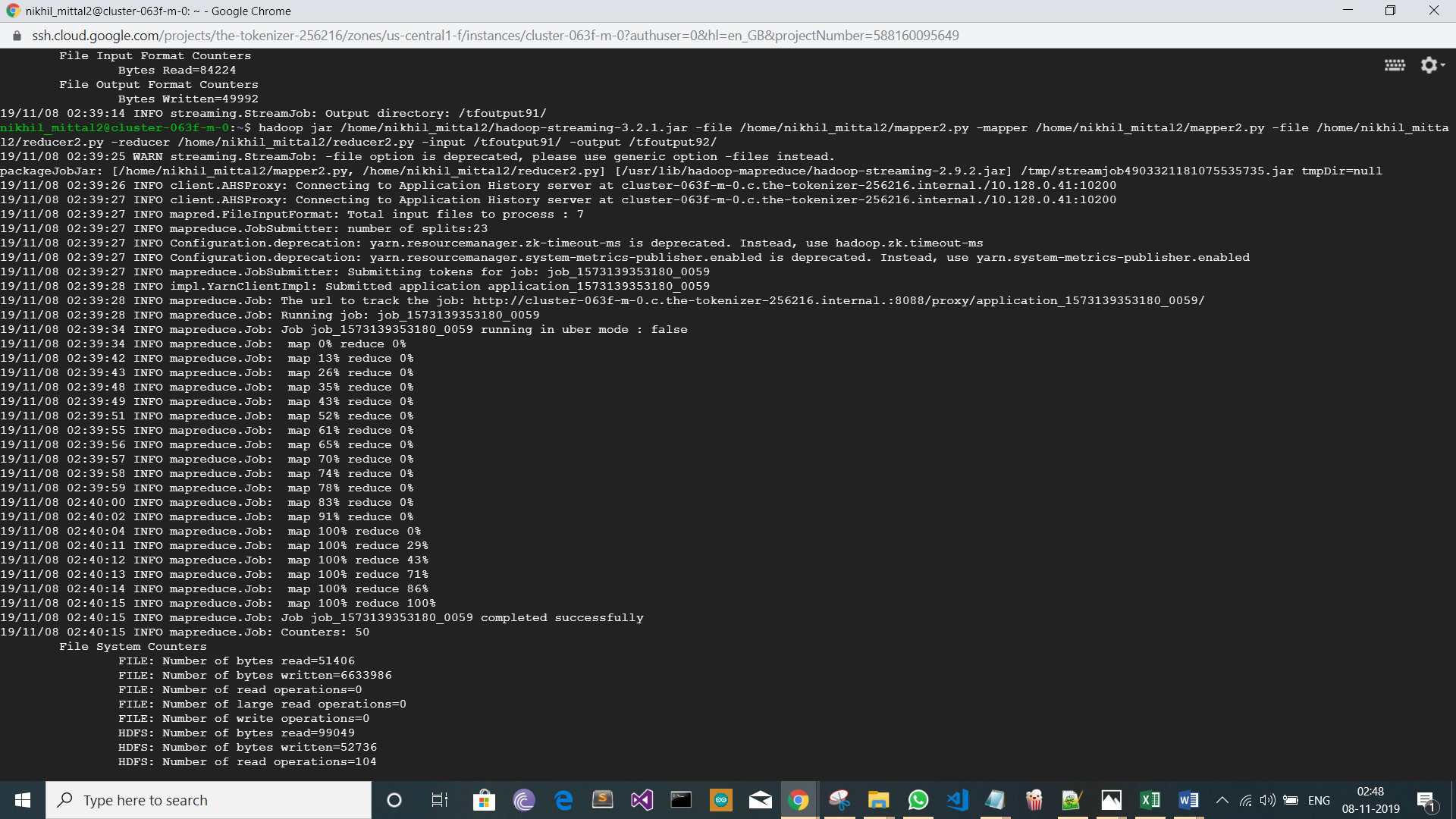


Hadoop Cluster Mapper Reducer is performed four times.

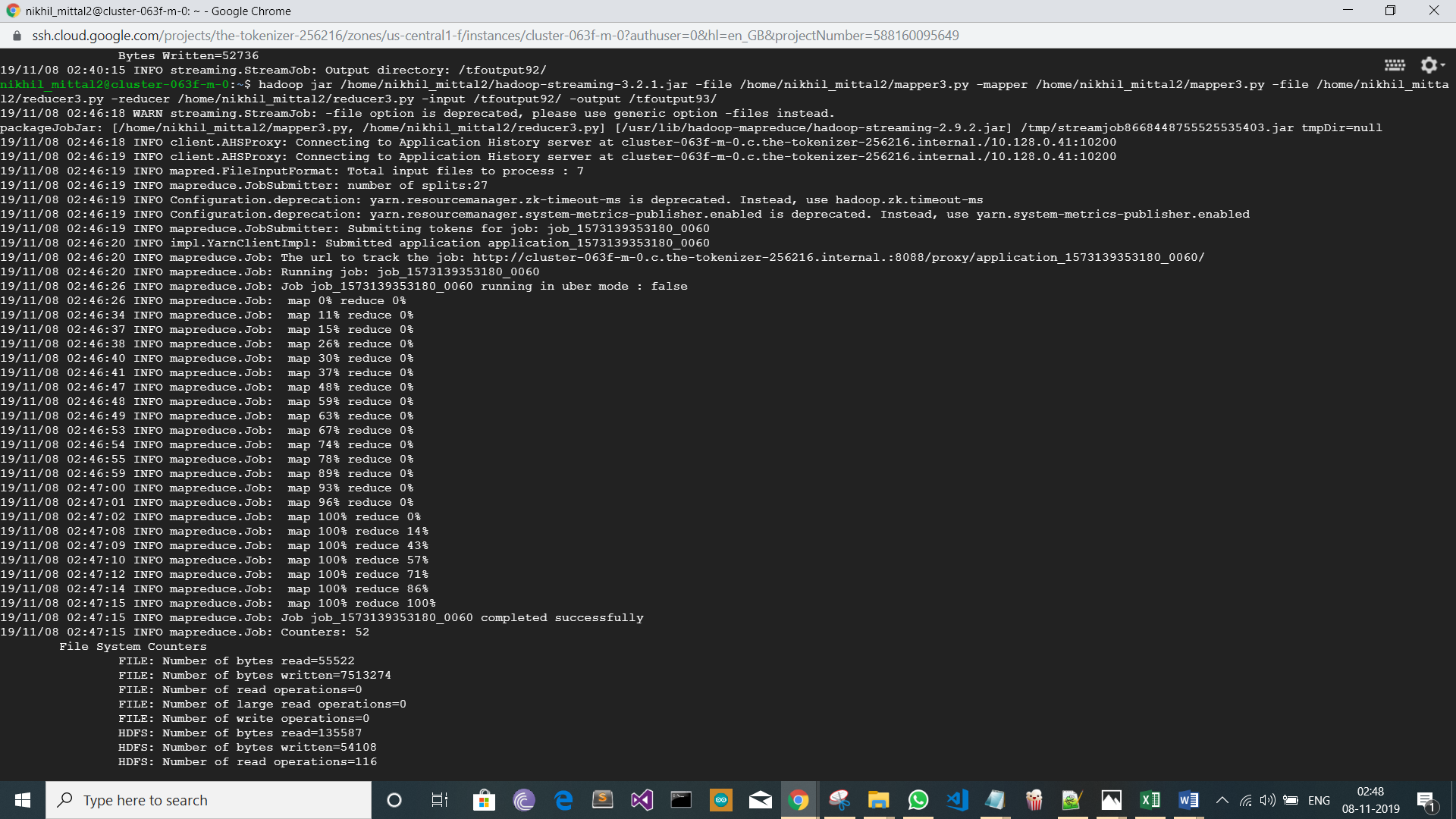
1. hadoop jar /home/nikhil\_mittal2/hadoop-streaming-3.2.1.jar -file /home/nikhil\_mittal2/mapper1.py -mapper /home/nikhil\_mittal2/mapper1.py -file /home/nikhil\_mittal2/reducer1.py -reducer /home/nikhil\_mittal2/reducer1.py -input /home/nikhil\_mittal2/output10/000007\_0 -output /tfoutput91/



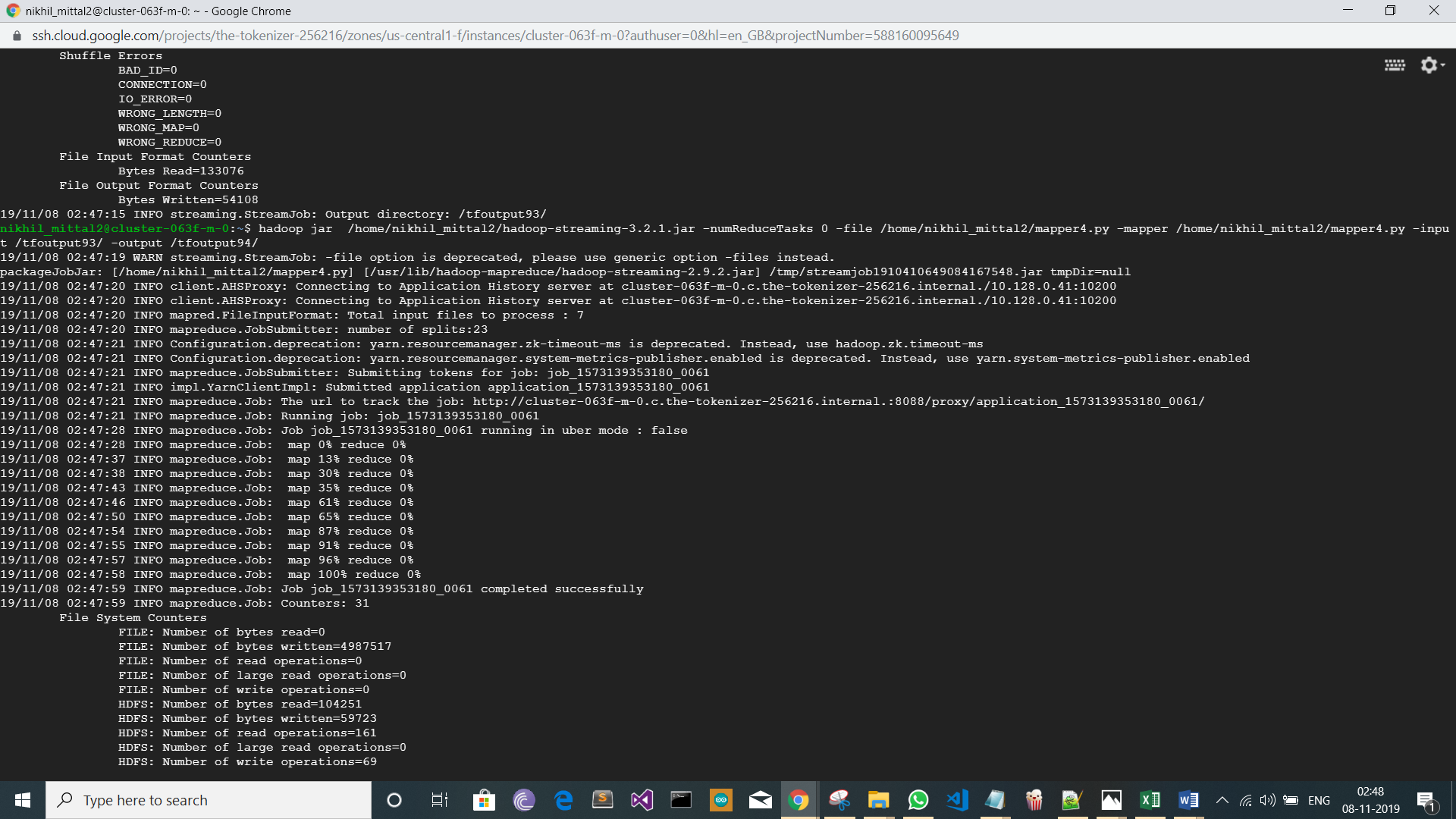
1. hadoop jar /home/nikhil\_mittal2/hadoop-streaming-3.2.1.jar -file /home/nikhil\_mittal2/mapper2.py -mapper /home/nikhil\_mittal2/mapper2.py -file /home/nikhil\_mittal2/reducer2.py -reducer /home/nikhil\_mittal2/reducer2.py -input /tfoutput91/ -output /tfoutput92/



1. hadoop jar /home/nikhil\_mittal2/hadoop-streaming-3.2.1.jar -file /home/nikhil\_mittal2/mapper3.py -mapper /home/nikhil\_mittal2/mapper3.py -file /home/nikhil\_mittal2/reducer3.py -reducer /home/nikhil\_mittal2/reducer3.py -input /tfoutput92/ -output /tfoutput93/

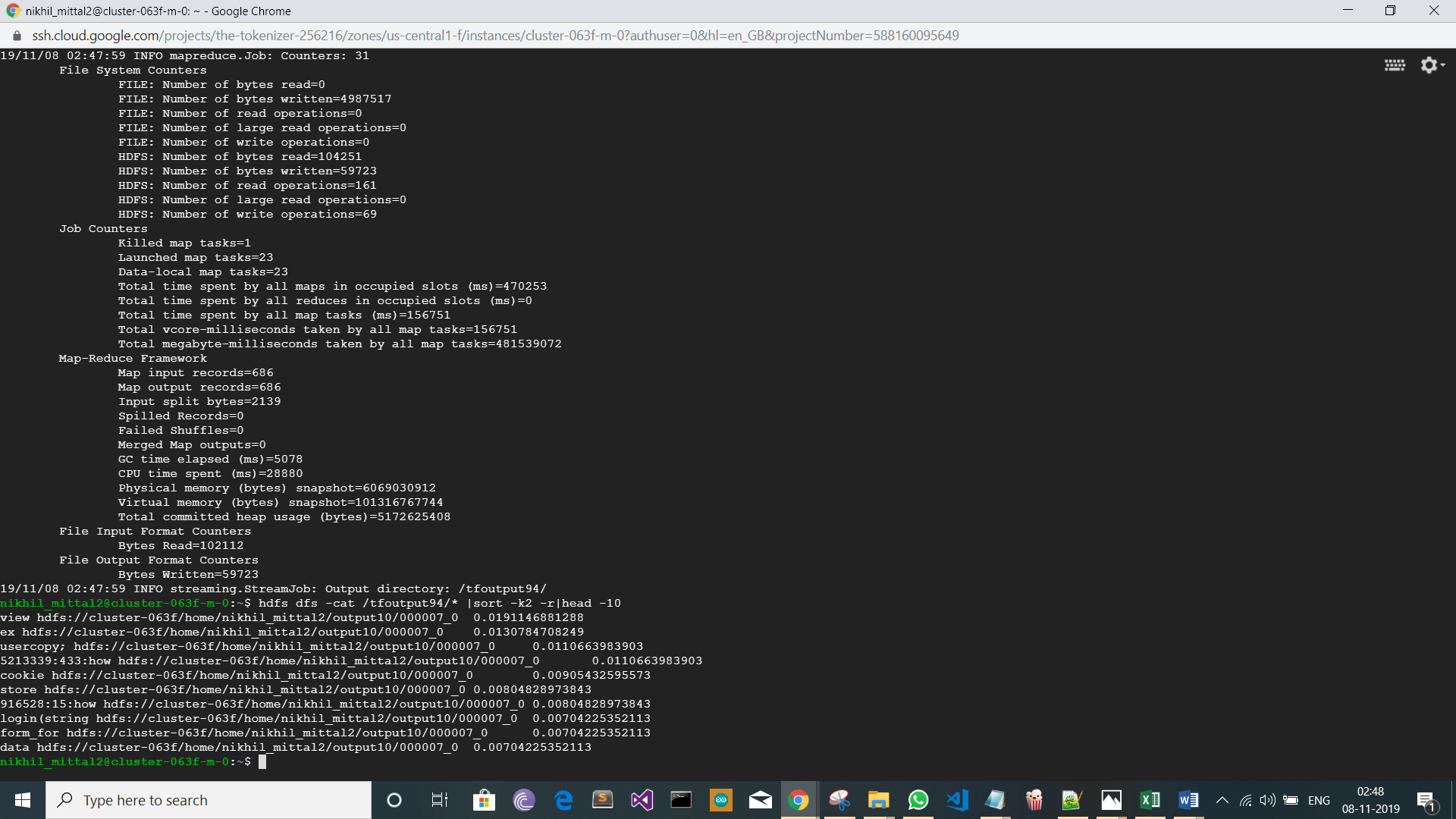


1. hadoop jar /home/nikhil\_mittal2/hadoop-streaming-3.2.1.jar -numReduceTasks 0 -file /home/nikhil\_mittal2/mapper4.py -mapper /home/nikhil\_mittal2/mapper4.py -input /tfoutput93/ -output /tfoutput94/



Final Output for mapper reducer using TFIDF

hdfs dfs -cat /tfoutput94/\* |sort -k2 -r|head -10



Rest of the user’s data are done with same structure, but different variable and the outputs are there in the Screenshots>TF-IDF.Please See that folder because then the document would have more pages.

Screenshots can be viewed from this Link(https://drive.google.com/drive/folders/1lo\_Ca5s7W0PXBkyf9Mwv7R-eqilg2ke2?usp=sharing). Works only with a DCU email id.

REFERENCES:-

1. <https://www.tutorialspoint.com/apache_pig/index.htm>.
2. <https://www.tutorialspoint.com/hive/index.htm>
3. <https://github.com/devangpatel01/TF-IDF-implementation-using-map-reduce-Hadoop-python->
4. <https://jar-download.com/artifacts/org.apache.hadoop>