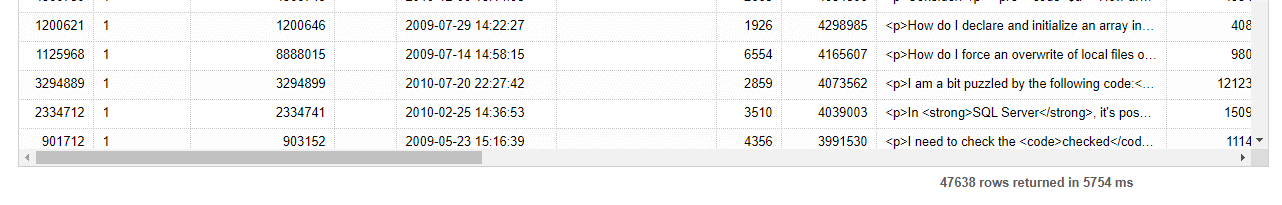
**Extracting Data From Stack Exchange**

1. :- Collecting 200000 records from stack Exchange having maximum View count.

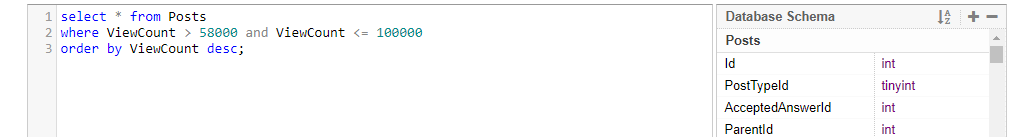
The stack Exchange will allow you to collect 50000 records at a time.

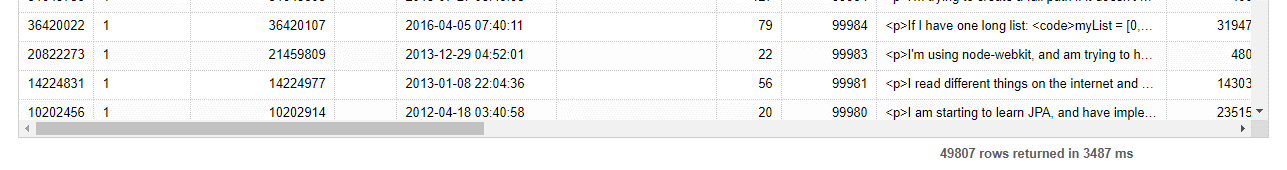
Query 1:- select \* from Posts Where ViewCount > 100000 order by ViewCount desc;



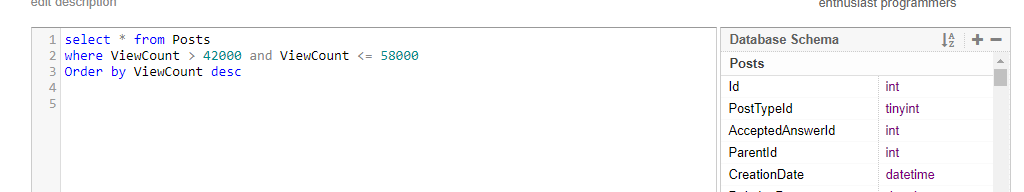


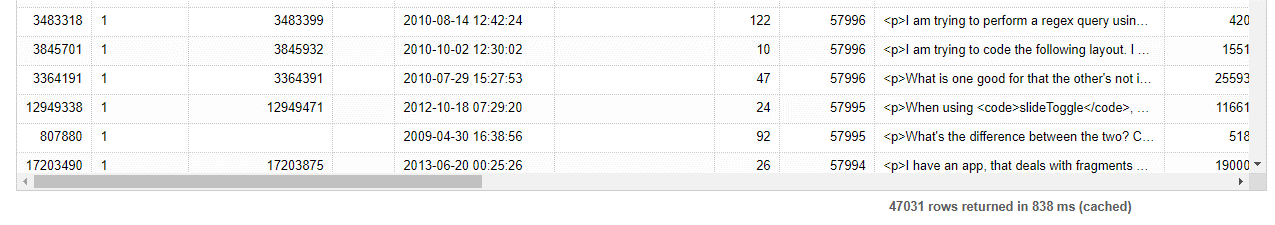
Query 2:select \* from Posts where ViewCount > 58000 and ViewCount <= 100000 order by ViewCount desc.



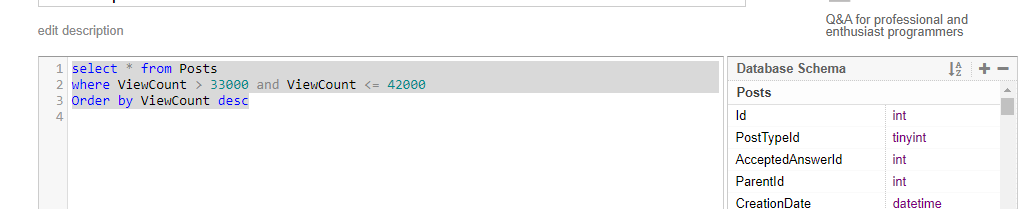


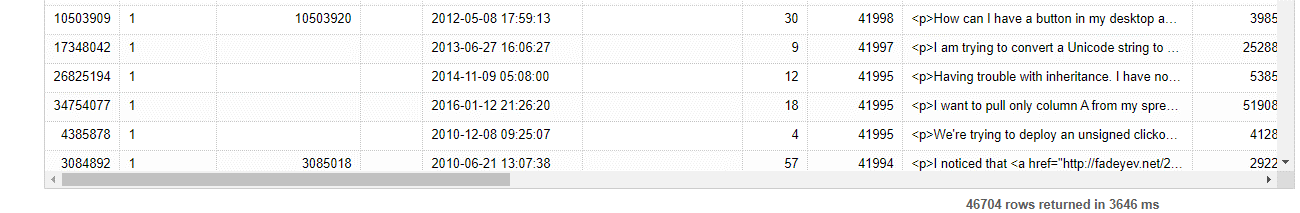
Query3:- select \* from Posts where ViewCount > 42000 and ViewCount <= 58000 Order by ViewCount desc





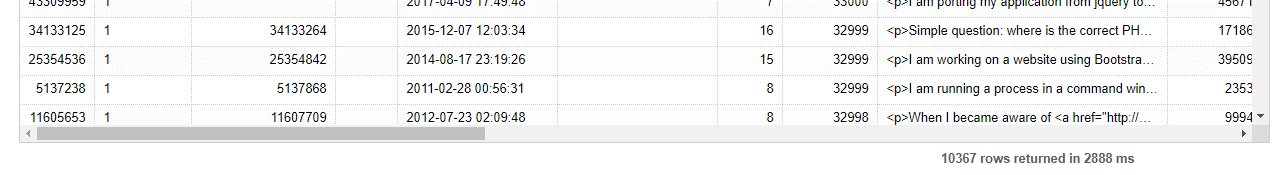
Query4:- select \* from Posts where ViewCount > 33000 and ViewCount <= 42000 Order by ViewCount desc



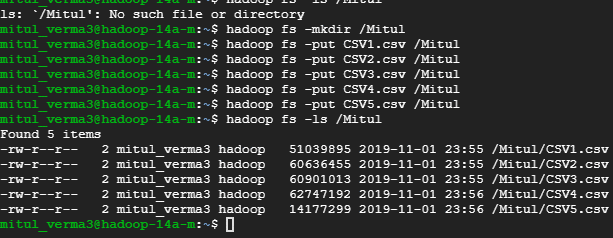


Query5:- select \* from Posts where ViewCount > 31500 and ViewCount <= 33000 Order by ViewCount desc





**ETL Using PIG**



**Loading data into Pig variable. The below command need to be followed for all the CSV uploaded.**

f1 = LOAD '/Mitul/csv1.csv' 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);



1. **Removing commas,/n,<.\*?> from Body, Title and Tags. Follow the below commands for all the loaded csv. The below action are performed only on 1 CSV. Do it for all.**

c1 = FOREACH f1 GENERATE Id AS Id, Score AS Score, REPLACE(Body,',\*','') AS Body, OwnerUserId AS OwnerUserId,

REPLACE (Title,',\*','') AS Title, REPLACE(Tags,',\*','') AS Tags;

1. **Removing ‘/n’**

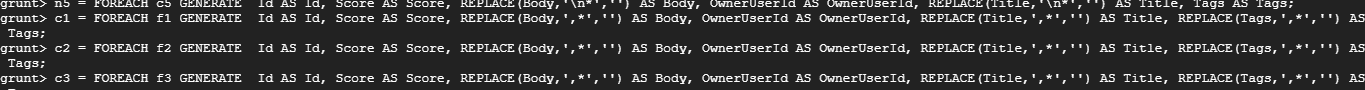
n1 = FOREACH c1 GENERATE Id AS Id, Score AS Score, REPLACE(Body,'\n\*','') AS Body, OwnerUserId AS OwnerUserId,

REPLACE(Title,'\n\*','') AS Title, REPLACE(Tags,'\n\*','') AS Tags;

1. **Removing <.\*?>**

nc1 = FOREACH n1 GENERATE Id AS Id, Score AS Score, REPLACE(Body,'<.\*?>','') AS Body, OwnerUserId AS

OwnerUserId, REPLACE(Title,'<.\*?>','') AS Title, Tags AS Tags;



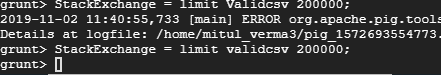
**Combining all the csv after cleaning.**



**Filter data to remove null records from OwnerID and Score**



**Storing Data in a variable Stack Exchange that contain exactly 200000 records.**



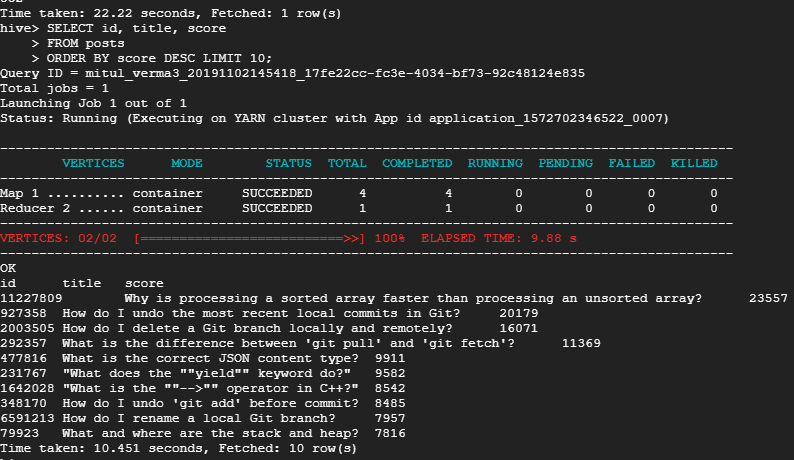
**Storing the records in HDFS**



**Working on Hive.**

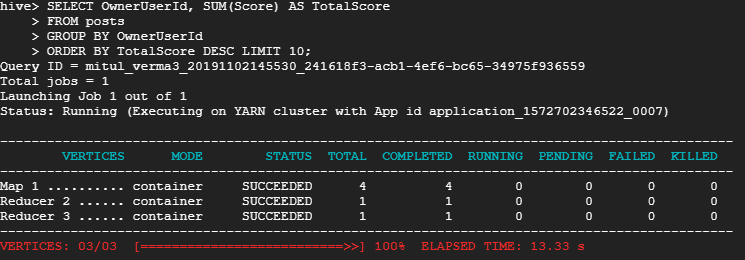
1. **The top 10 posts by score**

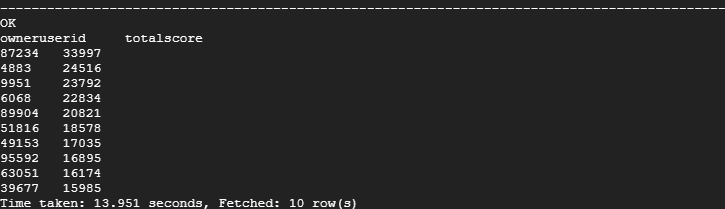
Select id, title, score from posts order by score DESC limit 10;



**Q2 ).** **The top 10 users by post score**

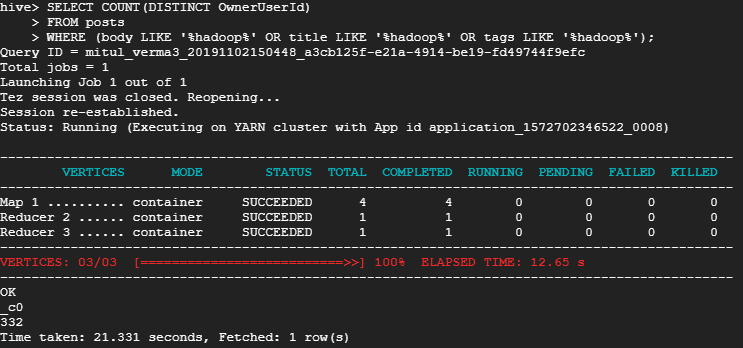
Select OwnerUserID, SUM(score) AS TotalScore from posts group by OwnerUserID order by TotalScore DESC limit 10;



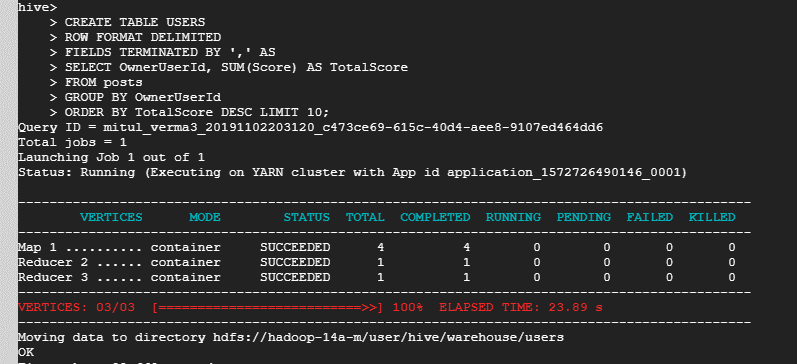


**Q3) The number of distinct users, who used the word “Hadoop” in one of their posts**Select Count(Distinct OwnerUserId) from posts Where (body LIKE ‘%hadoop%’ OR title LIKE ‘%hadoop%’

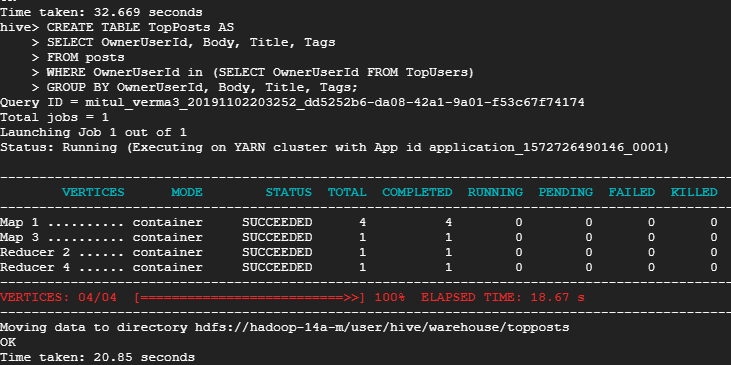
Or tags LIKE ‘%hadoop%’);



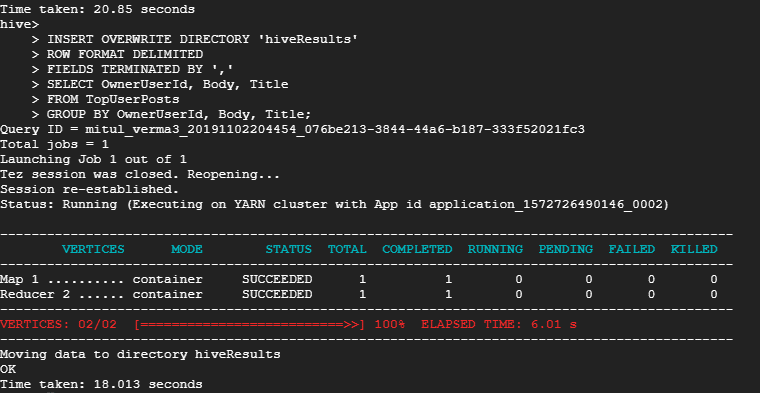
**TF-IDF**

1).Store top users in a separate table called Users:

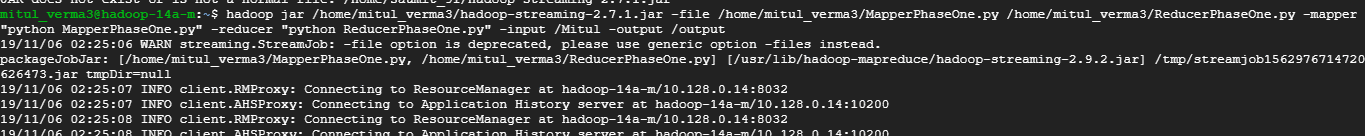
2) Creating a new table as TopPosts as shown below. Select owneruserid , body, Title, Tags from post and combine with the table created above.

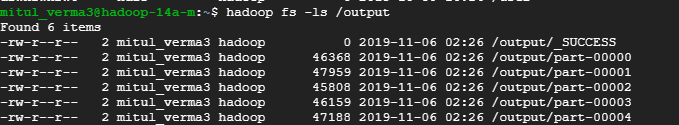


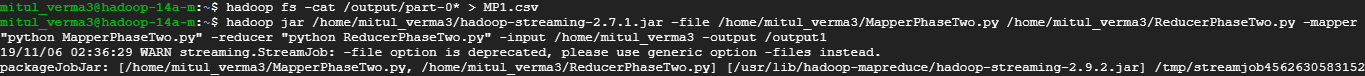
4)Saving the Result of Hive Table into HDFS Directory.



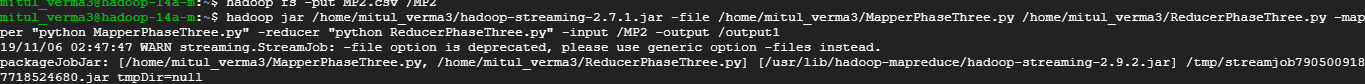
1)hadoop jar /home/mitul\_verma3/hadoop-streaming-2.7.1.jar -file /home/mitul\_verma3/MapperPhaseOne.py /home/mitul\_verma3/ReducerPhaseOne.py -mapper "python MapperPhaseOne.py" -reducer "python ReducerPhaseOne.py" -input /Mitul -output /output

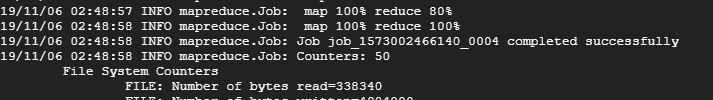


2) hadoop jar /home/mitul\_verma3/hadoop-streaming-2.7.1.jar -file /home/mitul\_verma3/MapperPhaseTwo.py /home/mitul\_verma3/ReducerPhaseTwo.py -mapper "python MapperPhaseTwo.py" -reducer "python ReducerPhasePTwo.py" -input /MP1 -output /output1



3) /home/mitul\_verma3/MapperPhaseThree.py /home/mitul\_verma3/ReducerPhaseThree.py -mapper "python MapperPhaseThree.py" -reducer "python ReducerPhasePThree.py" -input /MP2 -output /output2





**4)Create an external table in hive and load the csv into the table.**

sed -e 's/\s/,/g' Final\_Result.csv > out1.csv

create table tfidf(term String, Id int, tfidf float)

row format delimited

FIELDS TERMINATED BY ',' ;

load data local inpath 'out1.csv' overwrite into table tfidf;

5)Run the following query to get the top 10 terms for each of the top 10 users from Query 3.II.

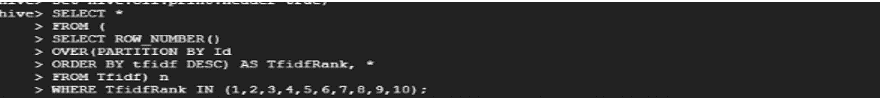
SELECT \*

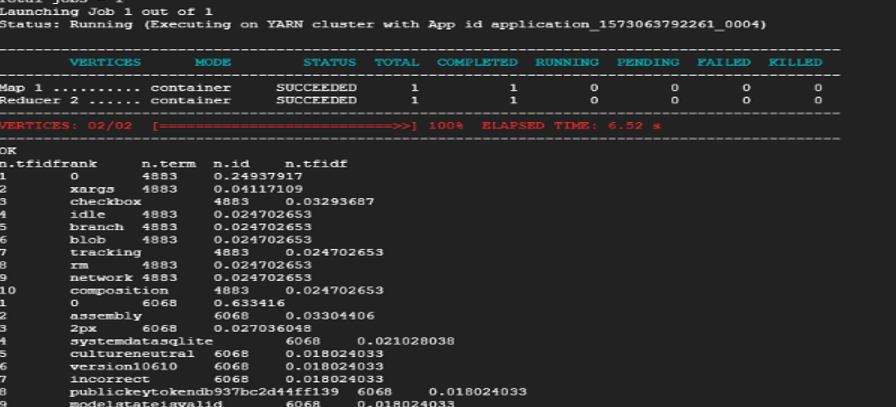
FROM (

SELECT ROW\_NUMBER ()

OVER(PARTITION BY Id

ORDER BY tfidf DESC) AS TfidfRank, \* FROM Tfidf) n WHERE TfidfRank IN (1,2,3,4,5,6,7,8,9,10);





**Important Note:**

The map-reduce programs are from <https://github.com/SatishUC15/TFIDF-HadoopMapReduce#tfidf-hadoop> with minor changes (i.e. addition of stop words and the related if condition in MapperPhaseOne.py)