Assignment 1: Data Analysis

Teenu Prathyush
DCU Student Id: 21262966
Email: teenu.prathyush2@mail.dcu.ie

In this assignment, I have perform data analysis on the top 200000 posts by ViewCount acquired on the platform StackExchange. The dataset contains 23 columns and around 200000 records which needs to be cleaned and processed before it is used for analysis as it contains a lot of messy data which is not needed in the analysis. The following technologies/platforms have been used in this assignment: Excel, RStudio, and Pig/Hive/MapReduce on Dataproc in Google Cloud Platform.

1. Documentation and steps undertaken for acquiring the top 200000 posts by ViewCount from StackExchange (Task 1):

Step 1: The following queries were executed one after the other using the data explorer feature on StackExchange (https://data.stackexchange.com/stackoverflow/query/new).

1. select count(*) from posts where posts.ViewCount > 100000
2. select top 50000 * from posts where posts.ViewCount > 100000 ORDER BY posts.ViewCount DESC
3. select top 50000 * from posts where posts.ViewCount < 127755 and posts.Id != 8618374 ORDER BY posts.ViewCount DESC
4. select top 50000 * from posts where posts.ViewCount < 74786 and posts.Id != 13836848 ORDER BY posts.ViewCount DESC
5. select top 50000 * from posts where posts.ViewCount < 53403 ORDER BY posts.ViewCount DESC

The 1st query is run to determine the number of records close to 50000 as StackExchange only allows to download 50000 records at a time. We assume that the view count is greater than 100000. In this case, we get a total of 68943 records.

The 2nd query is run to acquire the top 50000 posts having the most view count. From the previous query we know that the number of records having view count greater than 100000 is 68943 which is around 50000. So to extract the data we use – **ORDER by posts.ViewCount DESC** as it will determine the top 50000 records having the most view count, in descending order.

The 3rd query is run to determine the next 50000 posts having the most view count. We use the where clause - **posts.ViewCount < 127755 and posts.Id != 8618374**. This will ensure that there are no missing records in between queries.

The 4th query is run to determine the next 50000 posts having the most view count. We use the where clause - **posts.ViewCount < 74786 and posts.Id != 13836848.** This will ensure that there are no missing records and finally the 5th query is run to get the remaining posts having the most view count.

Step 2: After the queries have been executed, I have downloaded the four CSV files into the local system i.e., QueryResult1.csv, QueryResult2.csv, QueryResult3.csv, and QueryResult4.csv.

2. Steps undertaken to clean the data using RStudio (Tasks 2 & 3): The downloaded CSV files contains a lot of messy data which needs to be removed before loading it into the Google Cloud Platform for further processing. I have chosen RStudio to remove some of the messy data in the CSV files as I found RStudio to be easy to navigate and code. The commands used in the R-Script are fairly simple and processes the data very quickly when compared to Python, Excel, or any other platform. In this case, RStudio has been used to clean only some of the messy data, further cleaning/processing of data has been done using PIG on GCP.

Link to the R-Script:
https://gitlab.computing.dcu.ie/prathyt2/ca675_cloud_technologies_assignment_1/blob/master/Code/clean_data.R

3. Steps undertaken to further clean and process the data using Apache Pig.

Apache Pig has been used in this assignment to further clean and process the data before loading into hive for querying. I found Pig Latin to be easy to use for processing the data when compared to other Hadoop platforms. In this case, Pig has been used to load the data, remove unnecessary columns, process the data and join the four CSV files which was cleaned earlier.

Link to the Pig Script:
https://gitlab.computing.dcu.ie/prathyt2/ca675_cloud_technologies_assignment_1/blob/master/Code/PigScript.pig

Step 1: Uploading the CSV files into the folder - cleaned_data on the Google Cloud Staging Bucket.

Step 2: Opening the SSH connection and entering the command pig. It will take us to the grunt shell. All the code executed below is provided in the link to the Pig Script mentioned above. The complete script is executed with the command:

```
exec gs://dataproc-staging-us-central1-795277444073-rzswho19/PigScript.pig
```

The following command is used for loading CSV files with support of multi-line fields.

```
define CSVExcelStorage org.apache.pig.piggybank.storage.CSVExcelStorage();
```

Step 3: Loading the data into Pig using the following code snippet.

```
csv1=load'gs://dataproc-staging-us-central1-795277444073rzswho19/cleaned_data/QueryResult1.csv' using 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, ContentLicense:chararray);
```

The above code snippet is used to load the first CSV file – QueryResult1.csv into Pig environment. The YES_MULTILINE argument allows new lines inside of fields and the SKIP_INPUT_HEADER field skips the headers in the CSV file. The code to load the other three CSV files has been provided in the link to the Pig Script mentioned above.

Step 4: Joining all the four processed CSV files using the UNION command.

```
union_data = UNION csv1, csv2, csv3, csv4;
```

Step 5: Keeping only the necessary columns for processing.

```
required_data = FOREACH union_data GENERATE Id AS Id, Score AS Score, ViewCount AS ViewCount, Body AS Body, OwnerUserId AS OwnerUserId, Title AS Title, Tags AS Tags;
```

Step 6: Filtering data to remove all the null values from Score and OwnerUserId columns.

```
filter_data = FILTER required_data BY (OwnerUserId is NOT NULL);
```

```
final_data = FILTER filter_data BY (Score IS NOT NULL);
```

Step 7: Using the STORE command to store the processed data in the Google Cloud Staging Bucket

```
STORE final_data INTO 'gs://dataproc-staging-us-central1-795277444073-rzswho19/processed_data'
USING org.apache.pig.piggybank.storage.CSVExcelStorage(',');
```

4. Using Hive to query the data to get the following:

Hive has been used in this assignment to query the data. I found Hive to be very useful for analysing the data as it is very similar to MySQL. It only requires a few lines of code and it processes data very quickly.

Link to the Hive Queries:
https://gitlab.computing.dcu.ie/prathyt2/ca675_cloud_technologies_assignment_1/blob/master/Code/HiveQueries.sql

Step 1: Loading processed_data into the local file system and then on to HDFS.

```
hdfs dfs -get "gs://dataproc-staging-us-central1-795277444073-rzswho19/processed_data"
"/home/teenu_prathyush2/"
```

```
hdfs dfs -put "/home/teenu_prathyush2/processed_data/" "/user/pig"
```

Step 2: Opening Hive using the hive command on the terminal and then creating a table called as **stack_posts** using the create table command.

```
CREATE TABLE IF NOT EXISTS stack_posts(Id INT, Score INT, ViewCount INT, Body STRING,
OwnerUserId STRING, Title STRING, Tags STRING)  ROW FORMAT DELIMITED  FIELDS
TERMINATED BY ','  LOCATION '/user/pig/processed_data';
```

Step 3: Querying the table to check if the data has been loaded properly

```
SELECT Id, Score FROM stack_posts LIMIT 10;
```

Step 4: Query the table to get the results

Task 2.2.1. The top 10 posts by score

```
SELECT Id, Score, ViewCount, OwnerUserId, Title FROM stack_posts ORDER BY Score DESC
LIMIT 10;
```

Task 2.2.2. The top 10 users by post score

```
SELECT OwnerUserId, SUM(Score) AS TOTAL_SCORE FROM stack_posts GROUP BY
OwnerUserId ORDER BY TOTAL_SCORE DESC LIMIT 10;
```

Task 2.2.3. The number of distinct users, who used the word “cloud” in one of their posts

```
SELECT COUNT(DISTINCT OwnerUserId) AS distinct_users_count FROM stack_posts
WHERE(lower(body) LIKE '%cloud%' OR lower(title) LIKE '%cloud%' OR lower(tags) LIKE
'%cloud%');
```

Step 5: Creating a table called as top_users_scores to store the top 10 users by post scores

```
CREATE TABLE IF NOT EXISTS top_users_scores(OwnerUserId INT, TotalScore INT) ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

```
INSERT INTO top_users_scores SELECT OwnerUserId, SUM(Score) AS TOTAL_SCORE FROM
stack_posts GROUP BY OwnerUserId ORDER BY TOTAL_SCORE DESC LIMIT 10;
```

Step 6: Creating another table called as top_users_posts to store the text content from all of the top 10 users.

```
CREATE TABLE IF NOT EXISTS top_users_posts(OwnerUserId INT, Body STRING, Title
STRING, Tags STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

```
INSERT INTO top_users_posts SELECT OwnerUserId, Body, Title, Tags FROM stack_posts WHERE
OwnerUserId IN (SELECT OwnerUserId from top_users_scores) GROUP BY OwnerUserID, Body,
Title, Tags;
```

Step 7: Copying the data into HDFS

```
INSERT OVERWRITE DIRECTORY '/user/hive/stack_data' ROW FORMAT DELIMITED FIELDS
TERMINATED BY ',' SELECT OwnerUserId, Body, Title, Tags FROM top_users_posts GROUP BY
OwnerUserId, Body, Title, Tags;
```

5. Steps undertaken to calculate the per-user TF-IDF of the top 10 terms for each of the top 10 users. (Task 4)

The TF-IDF implementation has been accomplished in four different phases using three mappers and reducers each. A fourth mapper file is used to get the TF-IDF values of the top 10 terms for each of the top 10 users in a file. Since I have multiple mappers and reducer files, I have used MapReduce to implement TF-IDF as it is a programming model which processes data in a parallel manner and it can be implemented with Java/Python. I have used Python as it is user-friendly and easy to code.

Link to the mapper and reducer files:
https://gitlab.computing.dcu.ie/prathyt2/ca675_cloud_technologies_assignment_1/tree/master/Code/python_files

Step 1: Loading the mapreduce programs into local profile: /home/teenu_prathyush2 and giving full permissions to the directory – **python_files/**

```
hdfs dfs -get 'gs://dataproc-staging-us-central1-795277444073-rzswho19/python_files'
'/home/teenu_prathyush2/'
```

```
chmod 777 -R /home/teenu_prathyush2/python_files/
```

Step 2: Executing the first mapper and reducer file. The output will be stored in HDFS in the directory /user/hive/output1

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file
/home/teenu_prathyush2/python_files/mapper1.py /home/teenu_prathyush2/python_files/reducer1.py
-mapper "python mapper1.py" -reducer "python reducer1.py" -input /user/hive/stack_data/000000_0 -
output /user/hive/output1
```

Step 3: Executing the second mapper and reducer file. The output will be stored in HDFS in the directory /user/hive/output2

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar –file
```

```
/home/teenu_prathyush2/python_files/mapper2.py /home/teenu_prathyush2/python_files/reducer2.py -mapper "python mapper2.py" -reducer "python reducer2.py" -input /user/hive/output1 -output /user/hive/output2
```

Step 4: Executing the third mapper and reducer file. The output will be stored in HDFS in the directory /user/hive/output3

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -file
/home/teenu_prathyush2/python_files/mapper3.py /home/teenu_prathyush2/python_files/reducer3.py -mapper "python mapper3.py" -reducer "python reducer3.py" -input /user/hive/output2 -output /user/hive/output3
```

Step 5: Executing the fourth mapper file. The output will be stored in HDFS in the directory /user/hive/output4

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar –file
/home/teenu_prathyush2/python_files/mapper4.py -mapper "python mapper4.py" -input
/user/hive/output3 -output /user/hive/output4
```

Step 6: Creating an empty file – **output.csv** and merging all the files from outputdata4 into output.csv then replacing all the spaces and loading it into another CSV file – **output1.csv**.

```
touch /home/teenu_prathyush2/output.csv
```

```
hadoop fs -getmerge /user/hive/output4 /home/teenu_prathyush2/output.csv
```

```
sed -e 's/\s/,/g' /home/teenu_prathyush2/output.csv > /home/teenu_prathyush2/output1.csv
```

Step 7: Creating a table called **tfidf_data** and then loading the output (output1.csv) into the newly created table and then to view the per-user TF-IDF of the top 10 terms for each of the top 10 users.

```
CREATE TABLE IF NOT EXISTS tfidf_data(Term STRING, OwnerUserId INT, tfidf DOUBLE) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

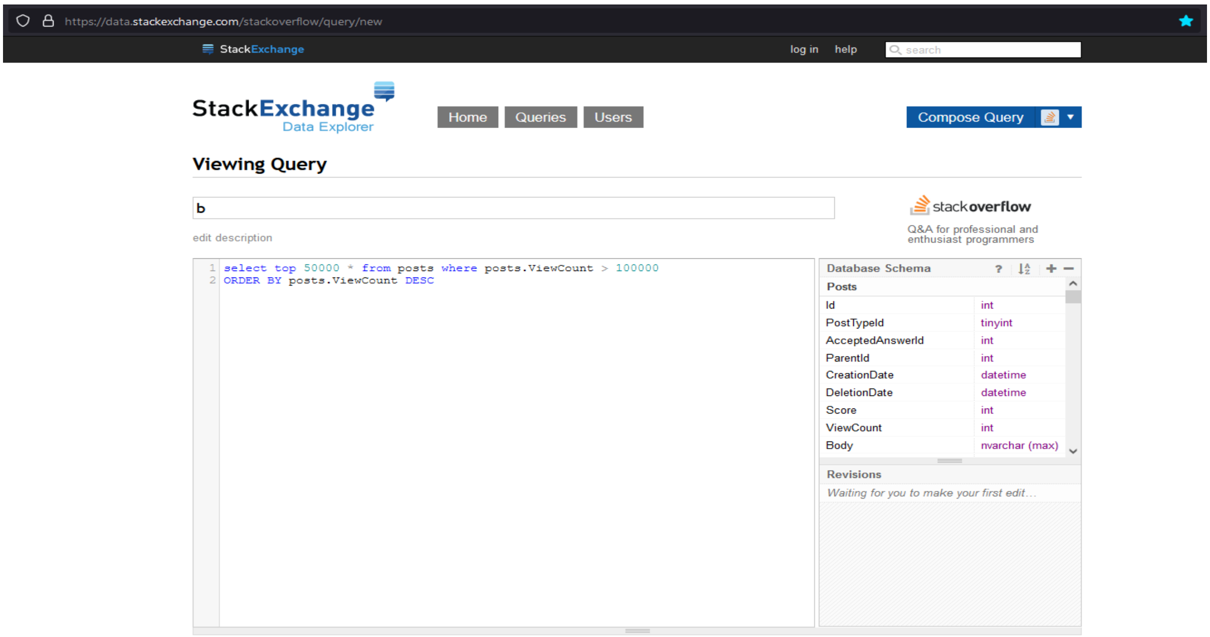
```
LOAD DATA LOCAL INPATH '/home/teenu_prathyush2/output1.csv' OVERWRITE INTO TABLE tfidf_data;
```

```
SELECT rank, OwnerUserId as userid, tfidf as tfidf_value, term FROM (SELECT ROW_NUMBER() OVER(PARTITION BY OwnerUserId ORDER BY tfidf DESC) AS rank, * FROM tfidf_data) n WHERE rank <= 10 and OwnerUserID IS NOT NULL;
```

REFERENCES:

1. https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
2. https://pig.apache.org/docs/r0.17.0/
3. https://cwiki.apache.org/confluence/display/Hive/Tutorial
4. https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/
5. https://github.com/kirthy21/Data-Analysis-Stack-Exchange-Hadoop-Pig-Hive-MapReduce-TFIDF
6. https://github.com/swathikiran86/pig-Hive-programmming-on-StackExchange-data
7. https://github.com/rajesh-codes/Stack-Exchange-Data-Analysis
8. https://gitlab.computing.dcu.ie/khaira2/ca675_cloud_technologies_assignment_1/tree/master

SCREENSHOTS


https://data.stackexchange.com/stackoverflow/query/new
StackExchange
log in
help
search
StackExchange Data Explorer
Home
Queries
Users
Compose Query
Viewing Query
stackoverflow
Q&A for professional and enthusiast programmers
edit description
Database Schema
Revisions
Waiting for you to make your first edit…

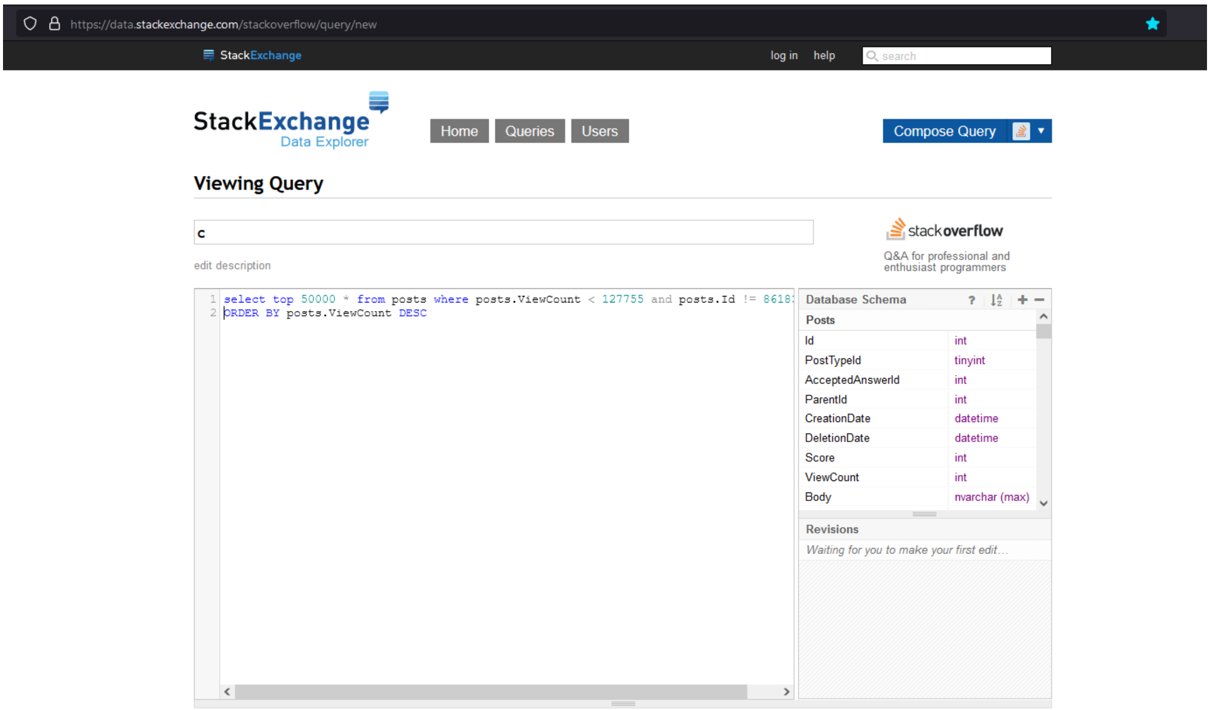
b

```
select top 50000 * from posts where posts.ViewCount > 100000
ORDER BY posts.ViewCount DESC
```

Posts	
Id	int
PostTypeId	tinyint
AcceptedAnswerId	int
ParentId	int
CreationDate	datetime
DeletionDate	datetime
Score	int
ViewCount	int
Body	nvarchar (max)

Figure 1: StackExchange Query 1

Figure 1 shows the first query being executed to acquire the top 50000 posts by ViewCount from StackExchange


https://data.stackexchange.com/stackoverflow/query/new
StackExchange
log in
help
search
StackExchange Data Explorer
Home
Queries
Users
Compose Query
Viewing Query
stackoverflow
Q&A for professional and enthusiast programmers
edit description
Database Schema
Revisions
Waiting for you to make your first edit…

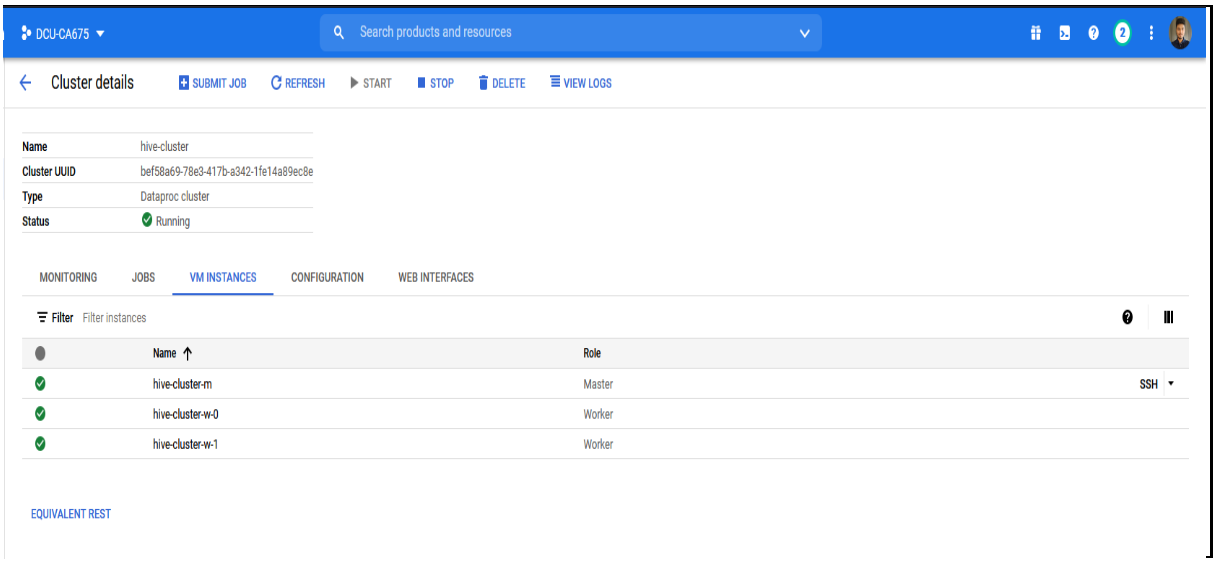
c

```
select top 50000 * from posts where posts.ViewCount < 127755 and posts.Id != 8618
ORDER BY posts.ViewCount DESC
```

Posts	
Id	int
PostTypeId	tinyint
AcceptedAnswerId	int
ParentId	int
CreationDate	datetime
DeletionDate	datetime
Score	int
ViewCount	int
Body	nvarchar (max)

Figure 2: StackExchange Query 2

Figure 2 shows the second query being executed to acquire the next 50000 posts by ViewCount from StackExchange. The remaining 2 queries are run to get the next 100000 posts by ViewCount on StackExchange

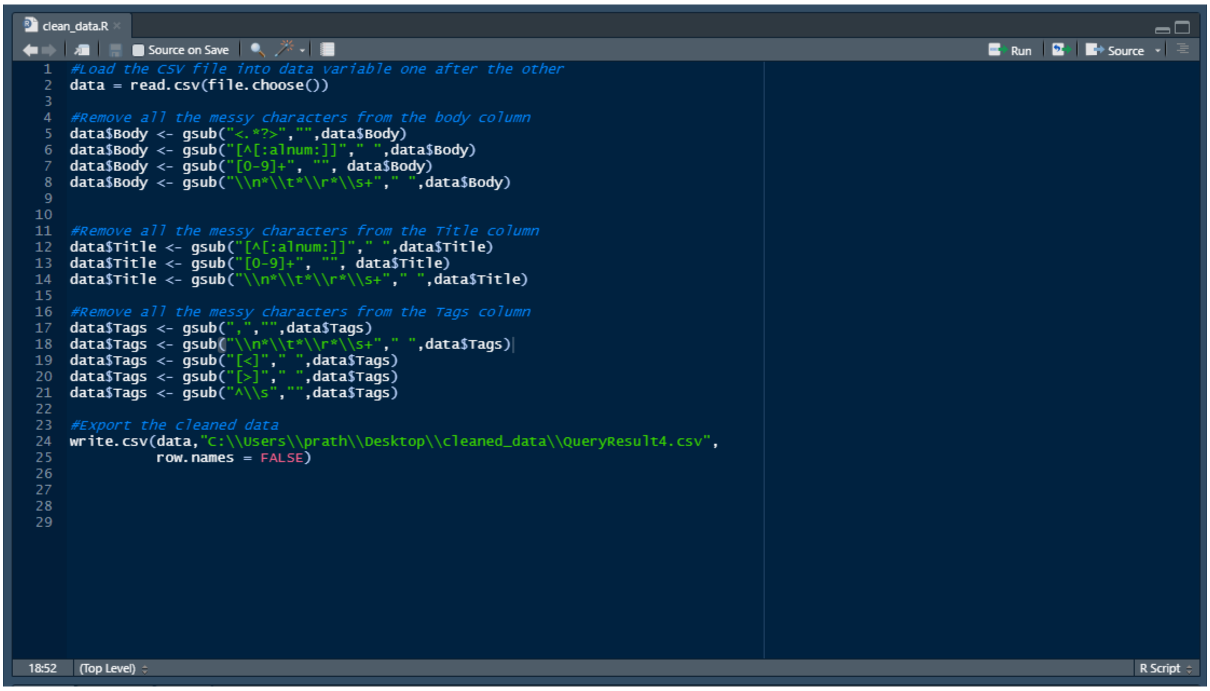


Name	hive-cluster
Cluster UUID	bef58a69-78e3-417b-a342-1fe14a89ec8e
Type	Dataproc cluster
Status	✓ Running

	Name ↑	Role
✓	hive-cluster-m	Master
✓	hive-cluster-w-0	Worker
✓	hive-cluster-w-1	Worker

Figure 3: Cluster Details

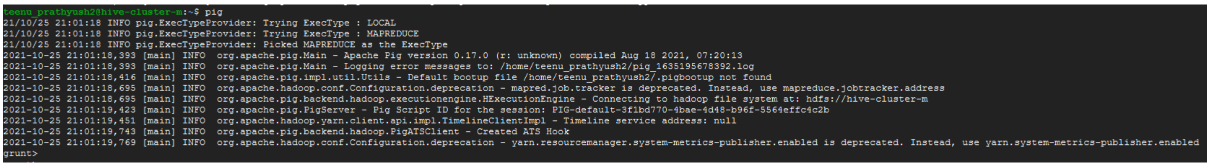
Figure 3 shows the cluster details on Google Cloud Platform. The cluster name is **hive-cluster**



```r
#Load the CSV file into data variable one after the other
data = read.csv(file.choose())

#Remove all the messy characters from the body column
data$Body <- gsub("<.*?>","",data$Body)
data$Body <- gsub("[^[:alnum:]]"," ",data$Body)
data$Body <- gsub("[0-9]+", "", data$Body)
data$Body <- gsub("\\n*\\t*\\r*\\s+"," ",data$Body)


#Remove all the messy characters from the Title column
data$Title <- gsub("[^[:alnum:]]"," ",data$Title)
data$Title <- gsub("[0-9]+", "", data$Title)
data$Title <- gsub("\\n*\\t*\\r*\\s+"," ",data$Title)

#Remove all the messy characters from the Tags column
data$Tags <- gsub(",","",data$Tags)
data$Tags <- gsub("\\n*\\t*\\r*\\s+"," ",data$Tags)
data$Tags <- gsub("[<]"," ",data$Tags)
data$Tags <- gsub("[>]"," ",data$Tags)
data$Tags <- gsub("^\\s","",data$Tags)

#Export the cleaned data
write.csv(data,"C:\\Users\\prath\\Desktop\\cleaned_data\\QueryResult4.csv",
          row.names = FALSE)
```

Figure 4: R-Script

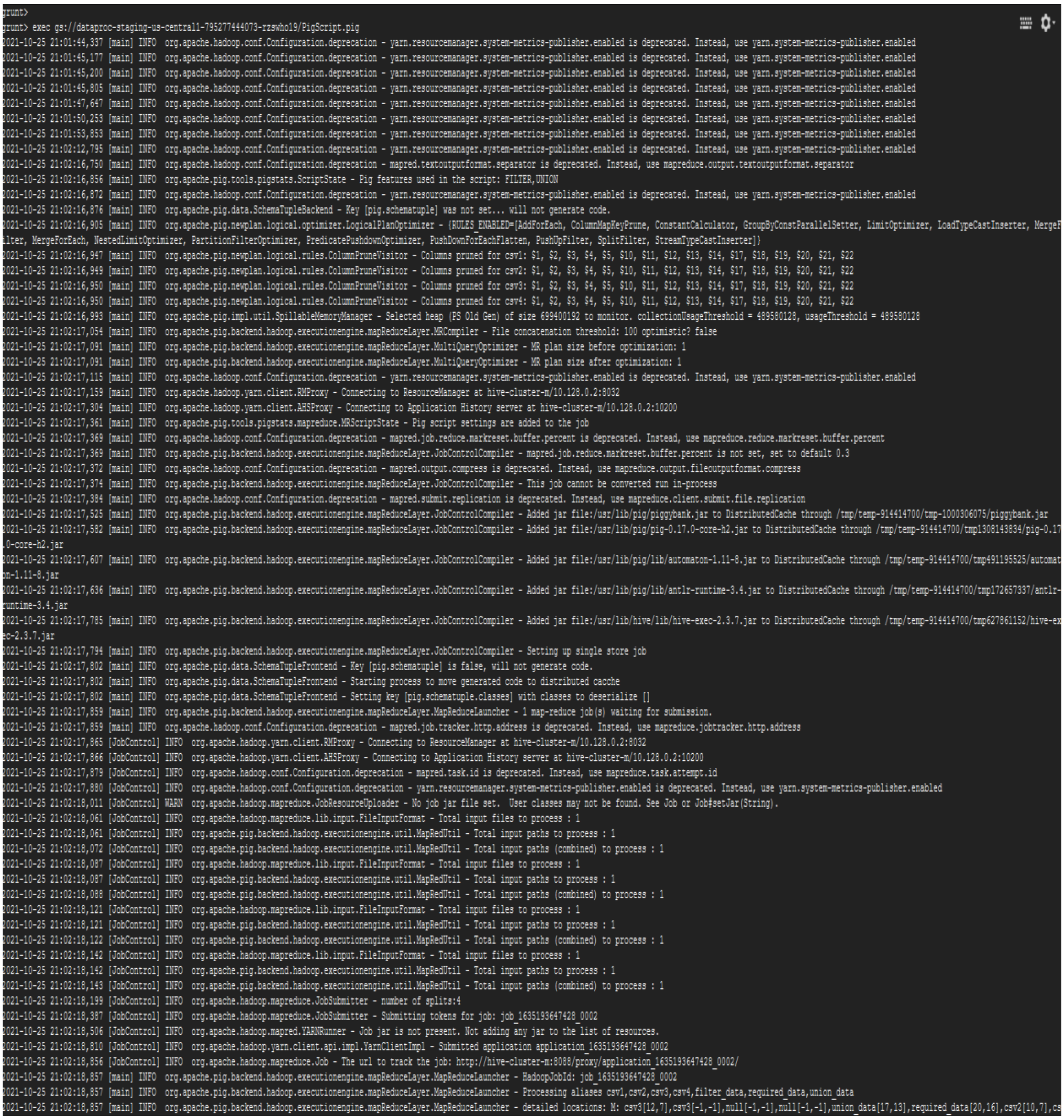
Figure 4 shows the R-Script used for cleaning the messy data in the dataset. This includes removing special characters, numbers, new lines, tab spaces etc. Cleaning is done on the body, title, and tags column.



```
teenu_prathyush2@hive-cluster-m:~$ pig
21/10/25 21:01:18 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
21/10/25 21:01:18 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
21/10/25 21:01:18 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2021-10-25 21:01:18,393 [main] INFO  org.apache.pig.Main - Apache Pig version 0.17.0 (r: unknown) compiled Aug 18 2021, 07:20:13
2021-10-25 21:01:18,393 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/teenu_prathyush2/pig_1635195678392.log
2021-10-25 21:01:18,416 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/teenu_prathyush2/.pigbootup not found
2021-10-25 21:01:18,695 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2021-10-25 21:01:18,695 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://hive-cluster-m
2021-10-25 21:01:19,423 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-3f1bd770-4bae-4d48-b96f-5564effc4c2b
2021-10-25 21:01:19,451 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: null
2021-10-25 21:01:19,743 [main] INFO  org.apache.pig.backend.hadoop.PigATSClient - Created ATS Hook
2021-10-25 21:01:19,769 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
grunt>
```

Figure 5: Grunt Shell

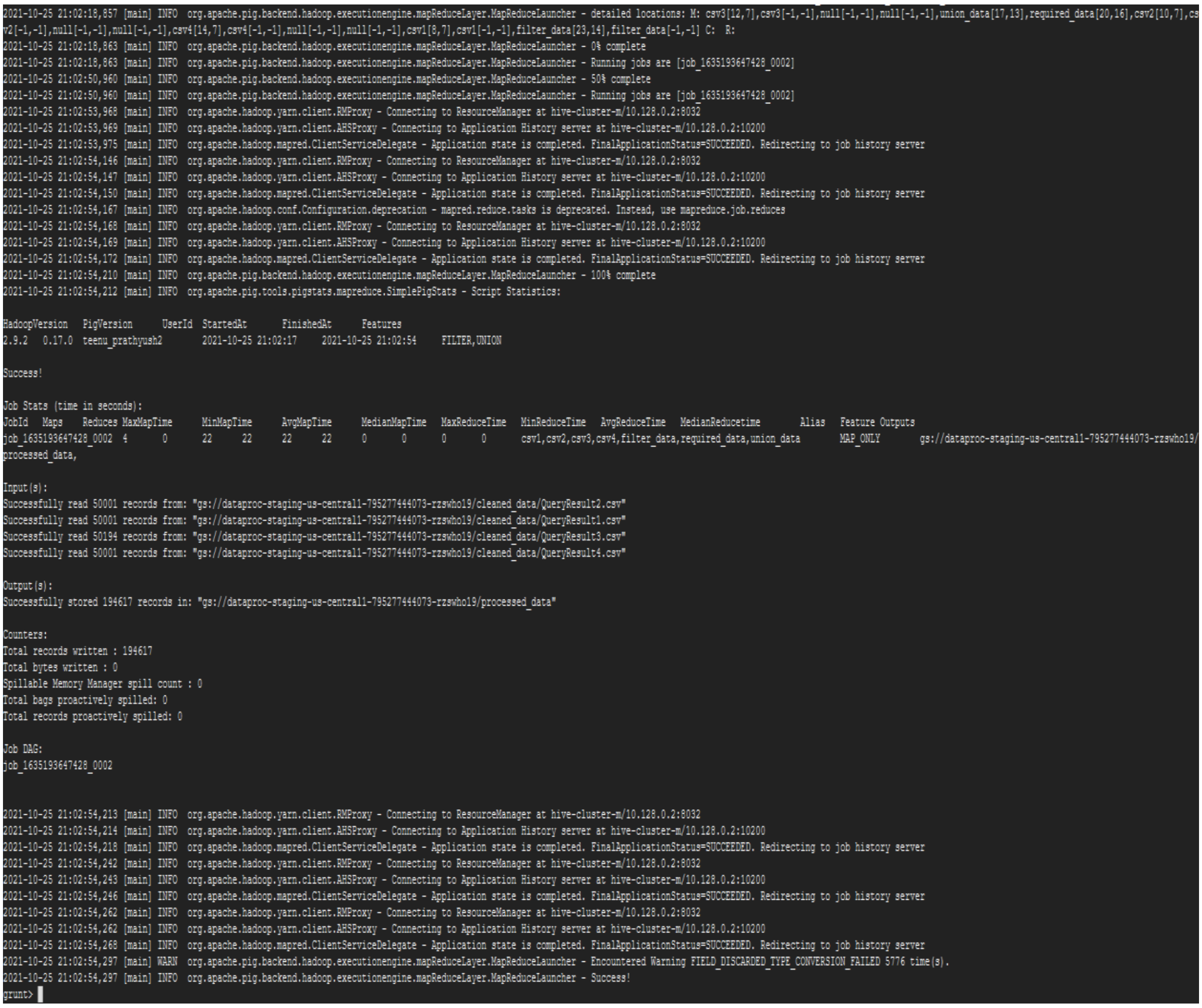
Figure 5 shows the Grunt shell being loaded using the PIG command



```
grunt>
grunt> exec gs://dataproc-staging-us-central1-795277444073-rzswho19/PigScript.pig
2021-10-25 21:01:44,337 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:01:45,177 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:01:45,200 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:01:45,805 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:01:47,647 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:01:50,253 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:01:53,853 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:02:12,795 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:02:16,750 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.textoutputformat.separator is deprecated. Instead, use mapreduce.output.textoutputformat.separator
2021-10-25 21:02:16,856 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: FILTER,UNION
2021-10-25 21:02:16,872 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:02:16,876 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2021-10-25 21:02:16,905 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, ConstantCalculator, GroupByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeF
ilter, MergeForEach, NestedLimitOptimizer, PartitionFilterOptimizer, PredicatePushdownOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter]}
2021-10-25 21:02:16,947 [main] INFO  org.apache.pig.newplan.logical.rules.ColumnPruneVisitor - Columns pruned for csv1: $1, $2, $3, $4, $5, $10, $11, $12, $13, $14, $17, $18, $19, $20, $21, $22
2021-10-25 21:02:16,949 [main] INFO  org.apache.pig.newplan.logical.rules.ColumnPruneVisitor - Columns pruned for csv2: $1, $2, $3, $4, $5, $10, $11, $12, $13, $14, $17, $18, $19, $20, $21, $22
2021-10-25 21:02:16,950 [main] INFO  org.apache.pig.newplan.logical.rules.ColumnPruneVisitor - Columns pruned for csv3: $1, $2, $3, $4, $5, $10, $11, $12, $13, $14, $17, $18, $19, $20, $21, $22
2021-10-25 21:02:16,950 [main] INFO  org.apache.pig.newplan.logical.rules.ColumnPruneVisitor - Columns pruned for csv4: $1, $2, $3, $4, $5, $10, $11, $12, $13, $14, $17, $18, $19, $20, $21, $22
2021-10-25 21:02:16,993 [main] INFO  org.apache.pig.impl.util.SpillableMemoryManager - Selected heap (PS Old Gen) of size 699400192 to monitor. collectionUsageThreshold = 489580128, usageThreshold = 489580128
2021-10-25 21:02:17,054 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenation threshold: 100 optimistic? false
2021-10-25 21:02:17,091 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size before optimization: 1
2021-10-25 21:02:17,091 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size after optimization: 1
2021-10-25 21:02:17,115 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:02:17,159 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at hive-cluster-m/10.128.0.2:8032
2021-10-25 21:02:17,304 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at hive-cluster-m/10.128.0.2:10200
2021-10-25 21:02:17,361 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.MRScriptState - Pig script settings are added to the job
2021-10-25 21:02:17,369 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.reduce.markreset.buffer.percent is deprecated. Instead, use mapreduce.reduce.markreset.buffer.percent
2021-10-25 21:02:17,369 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - mapred.job.reduce.markreset.buffer.percent is not set, set to default 0.3
2021-10-25 21:02:17,372 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.output.compress is deprecated. Instead, use mapreduce.output.fileoutputformat.compress
2021-10-25 21:02:17,374 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - This job cannot be converted run in-process
2021-10-25 21:02:17,384 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.submit.replication is deprecated. Instead, use mapreduce.client.submit.file.replication
2021-10-25 21:02:17,525 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Added jar file:/usr/lib/pig/piggybank.jar to DistributedCache through /tmp/temp-914414700/tmp-1000306075/piggybank.jar
2021-10-25 21:02:17,582 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Added jar file:/usr/lib/pig/pig-0.17.0-core-h2.jar to DistributedCache through /tmp/temp-914414700/tmp1308143834/pig-0.17
.0-core-h2.jar
2021-10-25 21:02:17,607 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Added jar file:/usr/lib/pig/lib/automaton-1.11-8.jar to DistributedCache through /tmp/temp-914414700/tmp491195525/automat
on-1.11-8.jar
2021-10-25 21:02:17,636 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Added jar file:/usr/lib/pig/lib/antlr-runtime-3.4.jar to DistributedCache through /tmp/temp-914414700/tmp172657337/antlr-
runtime-3.4.jar
2021-10-25 21:02:17,785 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Added jar file:/usr/lib/hive/lib/hive-exec-2.3.7.jar to DistributedCache through /tmp/temp-914414700/tmp627861152/hive-ex
ec-2.3.7.jar
2021-10-25 21:02:17,794 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Setting up single store job
2021-10-25 21:02:17,802 [main] INFO  org.apache.pig.data.SchemaTupleFrontend - Key [pig.schematuple] is false, will not generate code.
2021-10-25 21:02:17,802 [main] INFO  org.apache.pig.data.SchemaTupleFrontend - Starting process to move generated code to distributed cacche
2021-10-25 21:02:17,802 [main] INFO  org.apache.pig.data.SchemaTupleFrontend - Setting key [pig.schematuple.classes] with classes to deserialize []
2021-10-25 21:02:17,859 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 1 map-reduce job(s) waiting for submission.
2021-10-25 21:02:17,859 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker.http.address is deprecated. Instead, use mapreduce.jobtracker.http.address
2021-10-25 21:02:17,865 [JobControl] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at hive-cluster-m/10.128.0.2:8032
2021-10-25 21:02:17,866 [JobControl] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at hive-cluster-m/10.128.0.2:10200
2021-10-25 21:02:17,879 [JobControl] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.task.id is deprecated. Instead, use mapreduce.task.attempt.id
2021-10-25 21:02:17,880 [JobControl] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2021-10-25 21:02:18,011 [JobControl] WARN  org.apache.hadoop.mapreduce.JobResourceUploader - No job jar file set.  User classes may not be found. See Job or Job#setJar(String).
2021-10-25 21:02:18,061 [JobControl] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2021-10-25 21:02:18,061 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2021-10-25 21:02:18,072 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths (combined) to process : 1
2021-10-25 21:02:18,087 [JobControl] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2021-10-25 21:02:18,087 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2021-10-25 21:02:18,088 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths (combined) to process : 1
2021-10-25 21:02:18,121 [JobControl] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2021-10-25 21:02:18,121 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2021-10-25 21:02:18,122 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths (combined) to process : 1
2021-10-25 21:02:18,142 [JobControl] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2021-10-25 21:02:18,142 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2021-10-25 21:02:18,143 [JobControl] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths (combined) to process : 1
2021-10-25 21:02:18,199 [JobControl] INFO  org.apache.hadoop.mapreduce.JobSubmitter - number of splits:4
2021-10-25 21:02:18,387 [JobControl] INFO  org.apache.hadoop.mapreduce.JobSubmitter - Submitting tokens for job: job_1635193647428_0002
2021-10-25 21:02:18,506 [JobControl] INFO  org.apache.hadoop.mapred.YARNRunner - Job jar is not present. Not adding any jar to the list of resources.
2021-10-25 21:02:18,810 [JobControl] INFO  org.apache.hadoop.yarn.client.api.impl.YarnClientImpl - Submitted application application_1635193647428_0002
2021-10-25 21:02:18,856 [JobControl] INFO  org.apache.hadoop.mapreduce.Job - The url to track the job: http://hive-cluster-m:8088/proxy/application_1635193647428_0002/
2021-10-25 21:02:18,857 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - HadoopJobId: job_1635193647428_0002
2021-10-25 21:02:18,857 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Processing aliases csv1,csv2,csv3,csv4,filter_data,required_data,union_data
2021-10-25 21:02:18,857 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - detailed locations: M: csv3[12,7],csv3[-1,-1],null[-1,-1],null[-1,-1],union_data[17,13],required_data[20,16],csv2[10,7],cs
```

Figure 6: Executing PigScript.pig

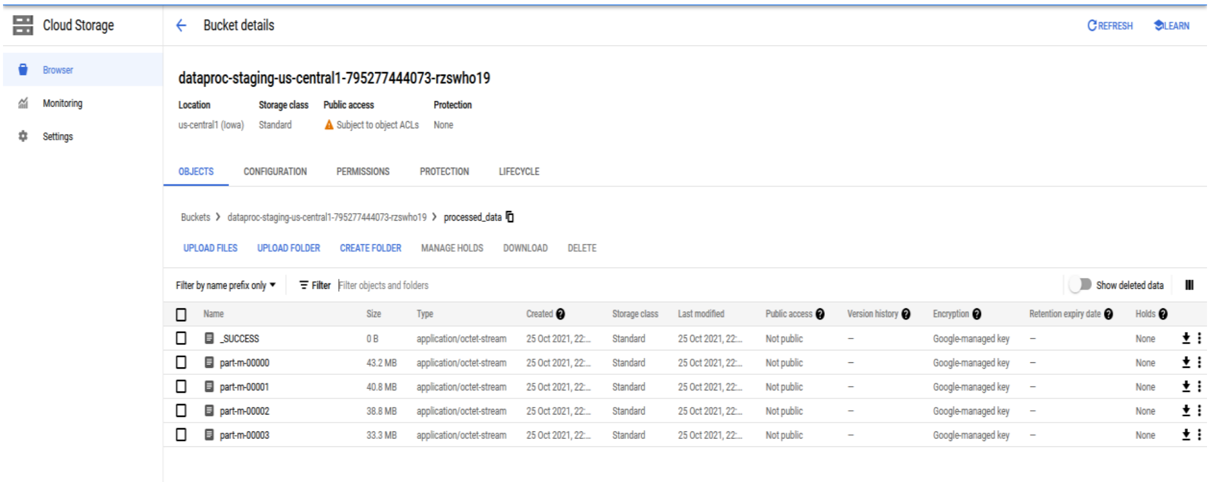
Figure 6 shows the pig script – PigScript.pig being executed.



```
2021-10-25 21:02:18,857 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - detailed locations: M: csv3[12,7],csv3[-1,-1],null[-1,-1],null[-1,-1],union_data[17,13],required_data[20,16],csv2[10,7],cs
v2[-1,-1],null[-1,-1],null[-1,-1],csv4[14,7],csv4[-1,-1],null[-1,-1],null[-1,-1],csv1[8,7],csv1[-1,-1],filter_data[23,14],filter_data[-1,-1] C:  R:
2021-10-25 21:02:18,863 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 0% complete
2021-10-25 21:02:18,863 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Running jobs are [job_1635193647428_0002]
2021-10-25 21:02:50,960 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 50% complete
2021-10-25 21:02:50,960 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Running jobs are [job_1635193647428_0002]
2021-10-25 21:02:53,968 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at hive-cluster-m/10.128.0.2:8032
2021-10-25 21:02:53,969 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at hive-cluster-m/10.128.0.2:10200
2021-10-25 21:02:53,975 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2021-10-25 21:02:54,146 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at hive-cluster-m/10.128.0.2:8032
2021-10-25 21:02:54,147 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at hive-cluster-m/10.128.0.2:10200
2021-10-25 21:02:54,150 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2021-10-25 21:02:54,167 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
2021-10-25 21:02:54,168 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at hive-cluster-m/10.128.0.2:8032
2021-10-25 21:02:54,169 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at hive-cluster-m/10.128.0.2:10200
2021-10-25 21:02:54,172 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2021-10-25 21:02:54,210 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2021-10-25 21:02:54,212 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.9.2   0.17.0  teenu_prathyush2        2021-10-25 21:02:17     2021-10-25 21:02:54     FILTER,UNION

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1635193647428_0002  4       0       22      22      22      22      0       0       0       0       csv1,csv2,csv3,csv4,filter_data,required_data,union_data        MAP_ONLY        gs://dataproc-staging-us-central1-795277444073-rzswho19/
processed_data,

Input(s):
Successfully read 50001 records from: "gs://dataproc-staging-us-central1-795277444073-rzswho19/cleaned_data/QueryResult2.csv"
Successfully read 50001 records from: "gs://dataproc-staging-us-central1-795277444073-rzswho19/cleaned_data/QueryResult1.csv"
Successfully read 50194 records from: "gs://dataproc-staging-us-central1-795277444073-rzswho19/cleaned_data/QueryResult3.csv"
Successfully read 50001 records from: "gs://dataproc-staging-us-central1-795277444073-rzswho19/cleaned_data/QueryResult4.csv"

Output(s):
Successfully stored 194617 records in: "gs://dataproc-staging-us-central1-795277444073-rzswho19/processed_data"

Counters:
Total records written : 194617
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1635193647428_0002


2021-10-25 21:02:54,213 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at hive-cluster-m/10.128.0.2:8032
2021-10-25 21:02:54,214 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at hive-cluster-m/10.128.0.2:10200
2021-10-25 21:02:54,218 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2021-10-25 21:02:54,242 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at hive-cluster-m/10.128.0.2:8032
2021-10-25 21:02:54,243 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at hive-cluster-m/10.128.0.2:10200
2021-10-25 21:02:54,246 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2021-10-25 21:02:54,262 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at hive-cluster-m/10.128.0.2:8032
2021-10-25 21:02:54,262 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at hive-cluster-m/10.128.0.2:10200
2021-10-25 21:02:54,268 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2021-10-25 21:02:54,297 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED 5776 time(s).
2021-10-25 21:02:54,297 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt> 
```

Figure 7 (a): Output of PigScript.pig

Figure 7(a) shows the output of PigScript.pig. It shows that 194617 records were processed successfully and stored in the folder – **processed_data** on Cloud Storage.



dataproc-staging-us-central1-795277444073-rzswho19

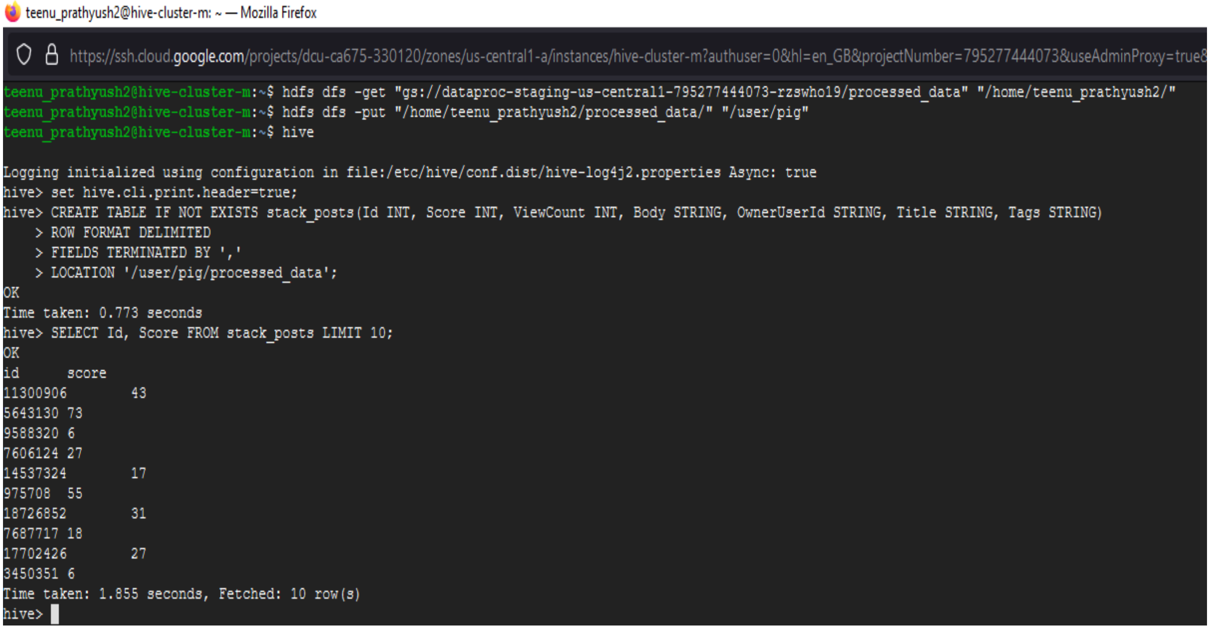
Location	Storage class	Public access	Protection
us-central1 (Iowa)	Standard	⚠ Subject to object ACLs	None

Buckets › dataproc-staging-us-central1-795277444073-rzswho19 › processed_data

□	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiry date	Holds
□	_SUCCESS	0 B	application/octet-stream	25 Oct 2021, 22:...	Standard	25 Oct 2021, 22:...	Not public	–	Google-managed key	–	None
□	part-m-00000	43.2 MB	application/octet-stream	25 Oct 2021, 22:...	Standard	25 Oct 2021, 22:...	Not public	–	Google-managed key	–	None
□	part-m-00001	40.8 MB	application/octet-stream	25 Oct 2021, 22:...	Standard	25 Oct 2021, 22:...	Not public	–	Google-managed key	–	None
□	part-m-00002	38.8 MB	application/octet-stream	25 Oct 2021, 22:...	Standard	25 Oct 2021, 22:...	Not public	–	Google-managed key	–	None
□	part-m-00003	33.3 MB	application/octet-stream	25 Oct 2021, 22:...	Standard	25 Oct 2021, 22:...	Not public	–	Google-managed key	–	None

Figure 7 (b): Output of PigScript.pig

Figure 7(b) shows the output being stored in Cloud Storage. The output will be then moved to the HDFS.

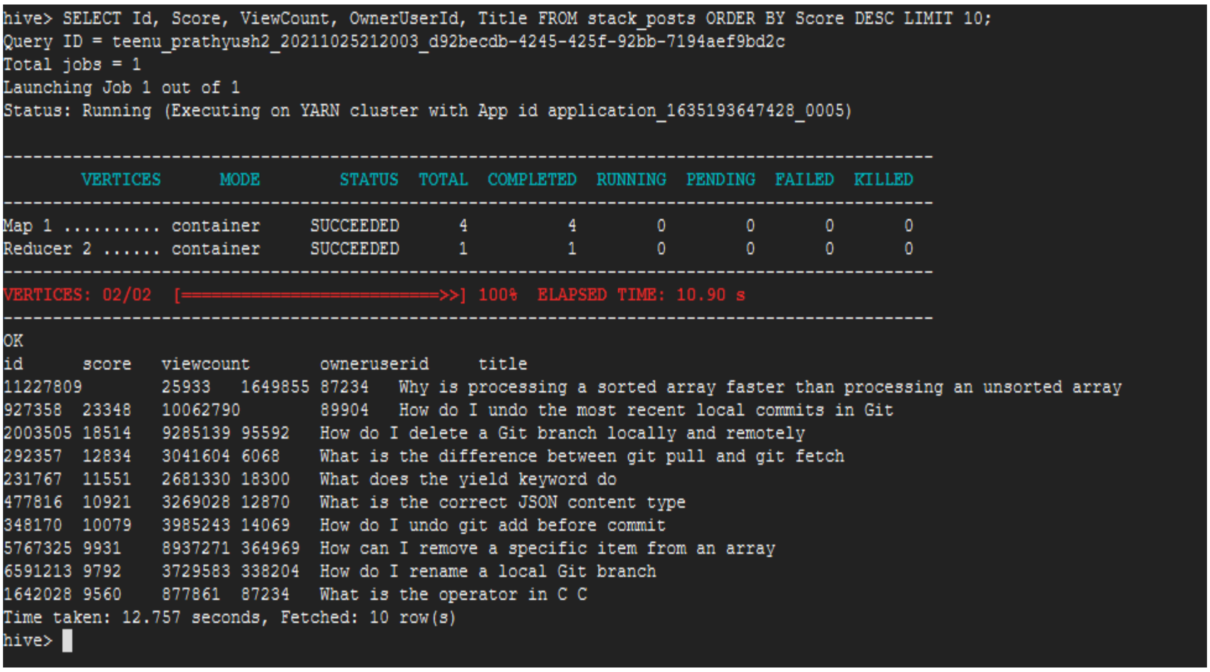


teenu_prathyush2@hive-cluster-m: ~ — Mozilla Firefox
https://ssh.cloud.google.com/projects/dcu-ca675-330120/zones/us-central1-a/instances/hive-cluster-m?authuser=0&hl=en_GB&projectNumber=795277444073&useAdminProxy=true&

```
teenu_prathyush2@hive-cluster-m:~$ hdfs dfs -get "gs://dataproc-staging-us-central1-795277444073-rzswho19/processed_data" "/home/teenu_prathyush2/"
teenu_prathyush2@hive-cluster-m:~$ hdfs dfs -put "/home/teenu_prathyush2/processed_data/" "/user/pig"
teenu_prathyush2@hive-cluster-m:~$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: true
hive> set hive.cli.print.header=true;
hive> CREATE TABLE IF NOT EXISTS stack_posts(Id INT, Score INT, ViewCount INT, Body STRING, OwnerUserId STRING, Title STRING, Tags STRING)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > LOCATION '/user/pig/processed_data';
OK
Time taken: 0.773 seconds
hive> SELECT Id, Score FROM stack_posts LIMIT 10;
OK
id      score
11300906        43
5643130 73
9588320 6
7606124 27
14537324        17
975708  55
18726852        31
7687717 18
17702426        27
3450351 6
Time taken: 1.855 seconds, Fetched: 10 row(s)
hive> 
```

Figure 8: Create table stack_posts

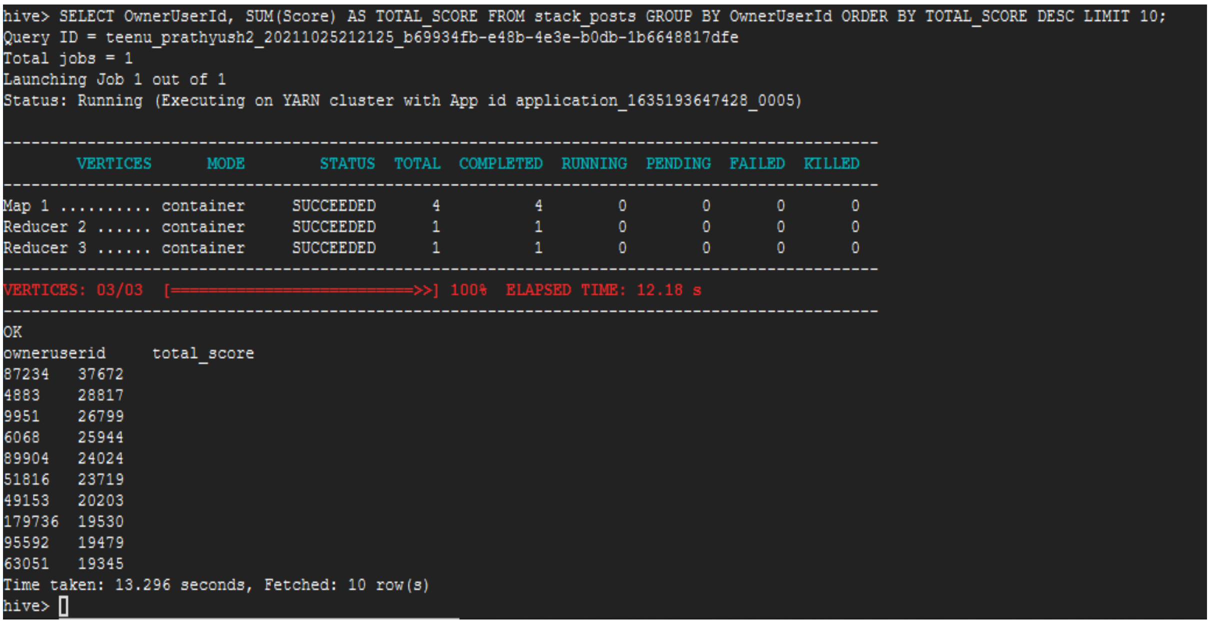
In figure 8, a table called as stack_posts is created using the create table command and the previously processed data from Pig is loaded into the Hive table.



```
hive> SELECT Id, Score, ViewCount, OwnerUserId, Title FROM stack_posts ORDER BY Score DESC LIMIT 10;
Query ID = teenu_prathyush2_20211025212003_d92becdb-4245-425f-92bb-7194aef9bd2c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635193647428_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 10.90 s
----------------------------------------------------------------------------------------------
OK
id      score   viewcount       owneruserid     title
11227809        25933   1649855 87234   Why is processing a sorted array faster than processing an unsorted array
927358  23348   10062790        89904   How do I undo the most recent local commits in Git
2003505 18514   9285139 95592   How do I delete a Git branch locally and remotely
292357  12834   3041604 6068    What is the difference between git pull and git fetch
231767  11551   2681330 18300   What does the yield keyword do
477816  10921   3269028 12870   What is the correct JSON content type
348170  10079   3985243 14069   How do I undo git add before commit
5767325 9931    8937271 364969  How can I remove a specific item from an array
6591213 9792    3729583 338204  How do I rename a local Git branch
1642028 9560    877861  87234   What is the operator in C C
Time taken: 12.757 seconds, Fetched: 10 row(s)
hive> 
```

Figure 9: Query to get the top 10 posts by score

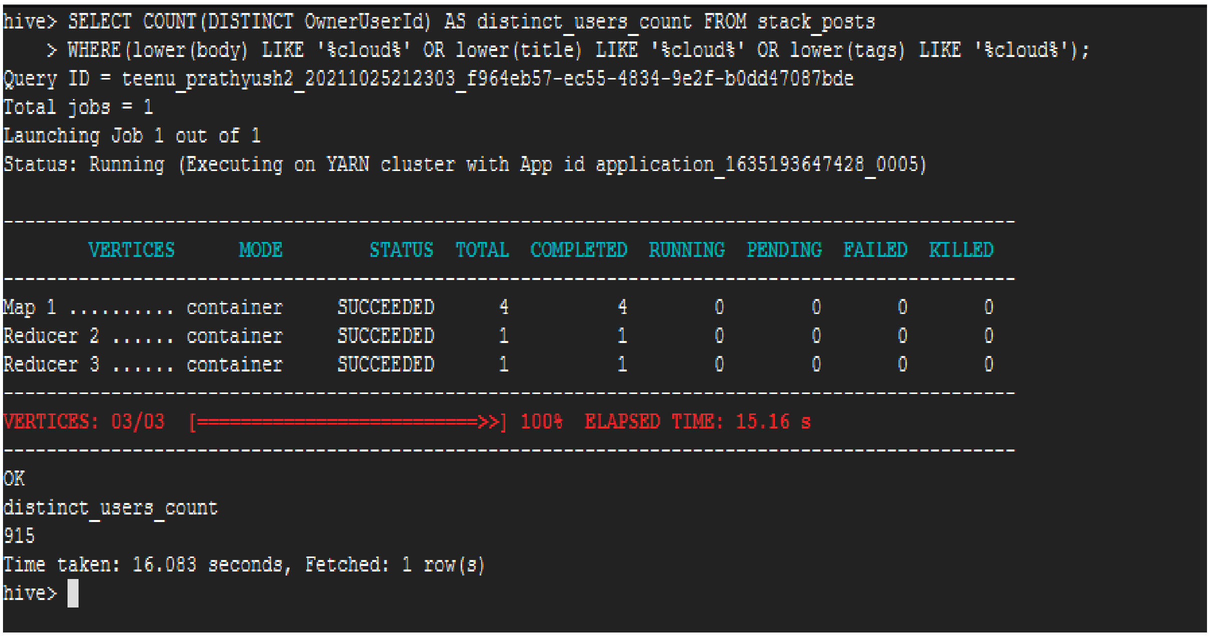
Figure 9 shows the query to get the top 10 posts by score and the result of the query (Task 2.2.1).



```
hive> SELECT OwnerUserId, SUM(Score) AS TOTAL_SCORE FROM stack_posts GROUP BY OwnerUserId ORDER BY TOTAL_SCORE DESC LIMIT 10;
Query ID = teenu_prathyush2_20211025212125_b69934fb-e48b-4e3e-b0db-1b6648817dfe
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635193647428_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 12.18 s
----------------------------------------------------------------------------------------------
OK
owneruserid     total_score
87234   37672
4883    28817
9951    26799
6068    25944
89904   24024
51816   23719
49153   20203
179736  19530
95592   19479
63051   19345
Time taken: 13.296 seconds, Fetched: 10 row(s)
hive> 
```

Figure 10: Query to get the Top 10 users by post score

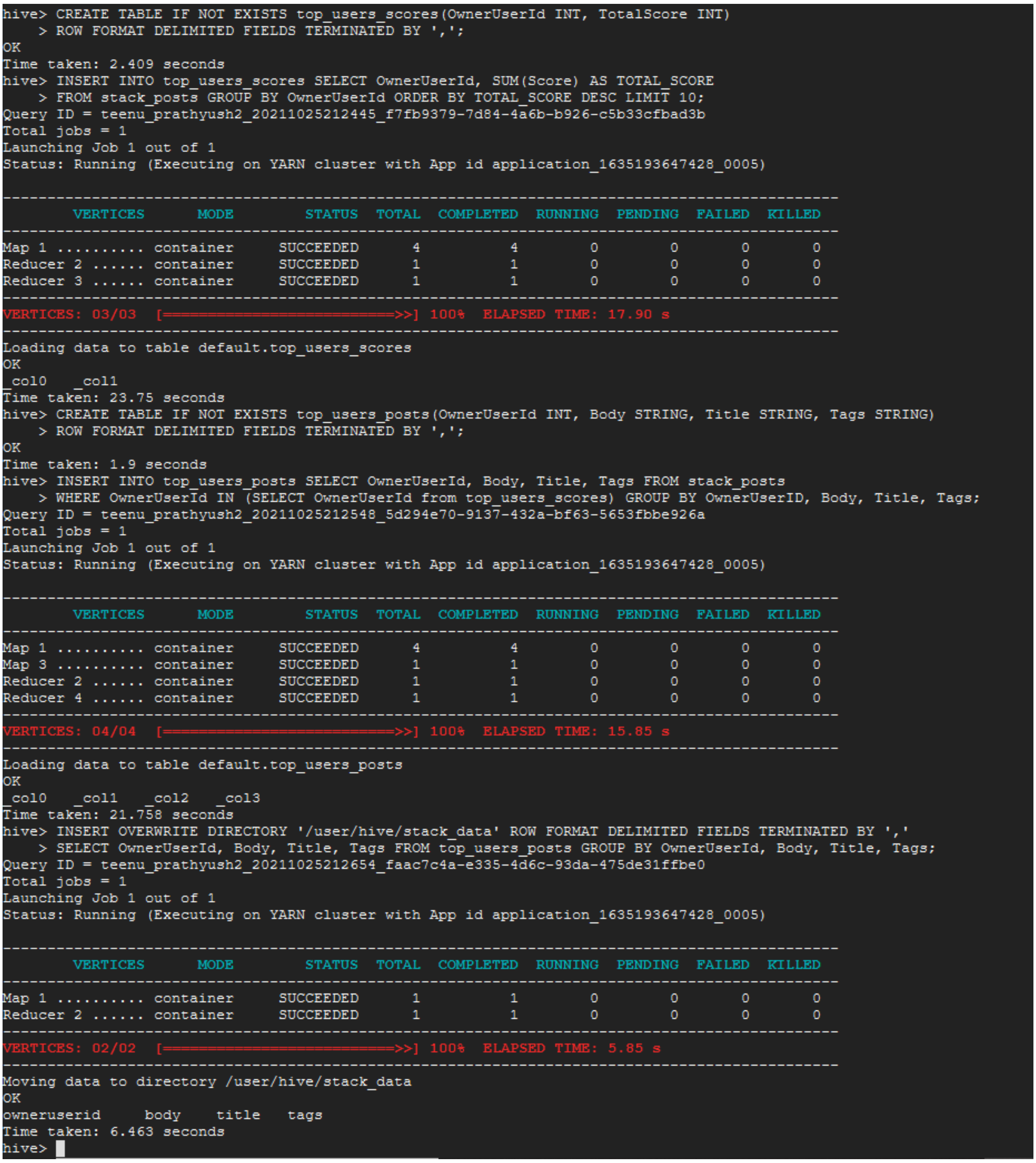
Figure 10 shows the query to get the top 10 users by post score and the result of the query (Task 2.2.2)



```
hive> SELECT COUNT(DISTINCT OwnerUserId) AS distinct_users_count FROM stack_posts
    > WHERE(lower(body) LIKE '%cloud%' OR lower(title) LIKE '%cloud%' OR lower(tags) LIKE '%cloud%');
Query ID = teenu_prathyush2_20211025212303_f964eb57-ec55-4834-9e2f-b0dd47087bde
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635193647428_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 15.16 s
----------------------------------------------------------------------------------------------
OK
distinct_users_count
915
Time taken: 16.083 seconds, Fetched: 1 row(s)
hive> 
```

Figure 11: No. of Distinct Users using the word cloud

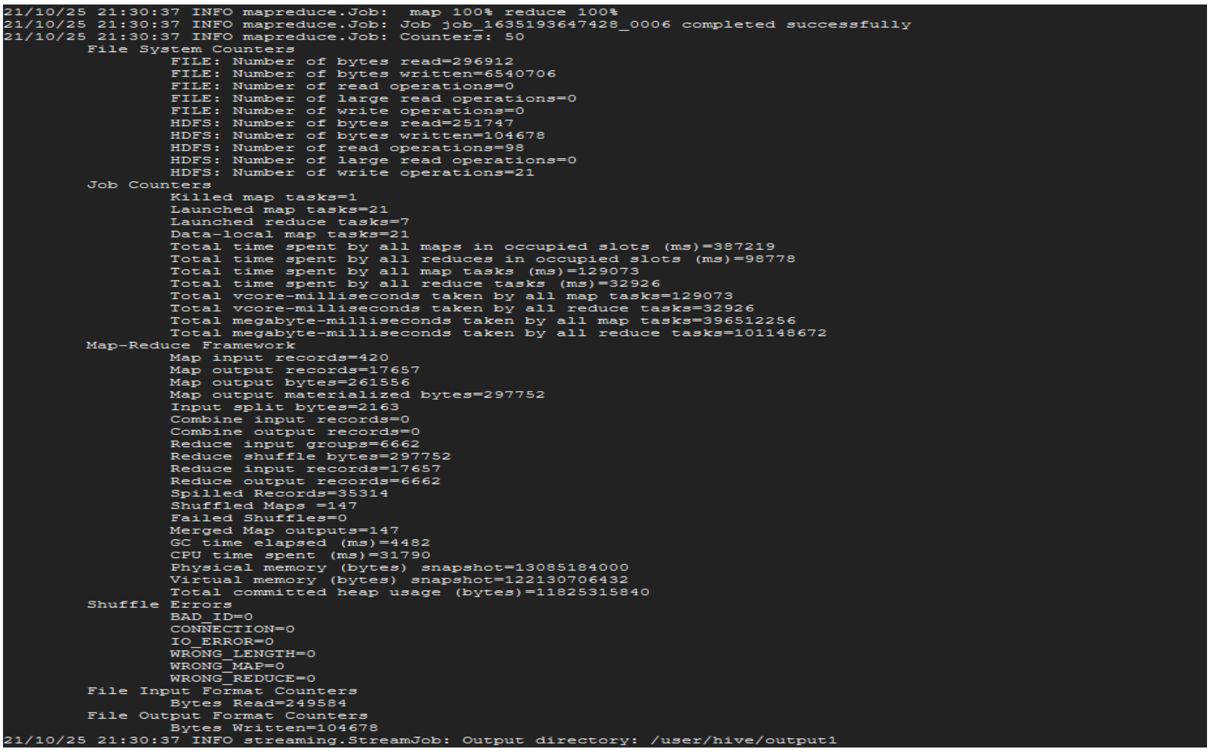
Figure 11 shows the query to get the number of distinct users who have used the word cloud in their posts and the result of the query (Task 2.2.3).



```
hive> CREATE TABLE IF NOT EXISTS top_users_scores(OwnerUserId INT, TotalScore INT)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 2.409 seconds
hive> INSERT INTO top_users_scores SELECT OwnerUserId, SUM(Score) AS TOTAL_SCORE
    > FROM stack_posts GROUP BY OwnerUserId ORDER BY TOTAL_SCORE DESC LIMIT 10;
Query ID = teenu_prathyush2_20211025212445_f7fb9379-7d84-4a6b-b926-c5b33cfbad3b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635193647428_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 17.90 s
----------------------------------------------------------------------------------------------
Loading data to table default.top_users_scores
OK
_col0   _col1
Time taken: 23.75 seconds
hive> CREATE TABLE IF NOT EXISTS top_users_posts(OwnerUserId INT, Body STRING, Title STRING, Tags STRING)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 1.9 seconds
hive> INSERT INTO top_users_posts SELECT OwnerUserId, Body, Title, Tags FROM stack_posts
    > WHERE OwnerUserId IN (SELECT OwnerUserId from top_users_scores) GROUP BY OwnerUserID, Body, Title, Tags;
Query ID = teenu_prathyush2_20211025212548_5d294e70-9137-432a-bf63-5653fbbe926a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635193647428_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Map 3 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 04/04  [==========================>>] 100%  ELAPSED TIME: 15.85 s
----------------------------------------------------------------------------------------------
Loading data to table default.top_users_posts
OK
_col0   _col1   _col2   _col3
Time taken: 21.758 seconds
hive> INSERT OVERWRITE DIRECTORY '/user/hive/stack_data' ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > SELECT OwnerUserId, Body, Title, Tags FROM top_users_posts GROUP BY OwnerUserId, Body, Title, Tags;
Query ID = teenu_prathyush2_20211025212654_faac7c4a-e335-4d6c-93da-475de31ffbe0
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635193647428_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.85 s
----------------------------------------------------------------------------------------------
Moving data to directory /user/hive/stack_data
OK
owneruserid     body    title   tags
Time taken: 6.463 seconds
hive>
```

Figure 12: Creating table top_users_scores and top_users_posts

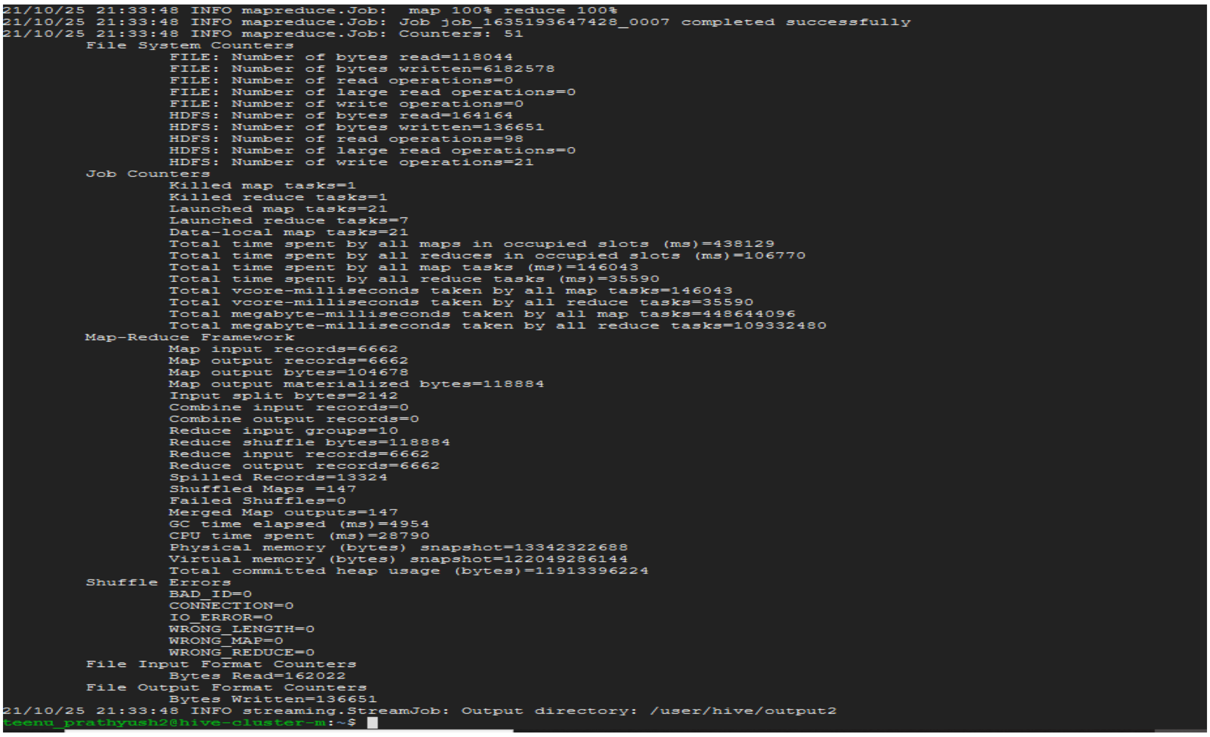
Figure 12 shows the creation of tables top_users_scores and top_users_posts which will be needed to calculate TFIDF later.



```
21/10/25 21:30:37 INFO mapreduce.Job:  map 100% reduce 100%
21/10/25 21:30:37 INFO mapreduce.Job: Job job_1635193647428_0006 completed successfully
21/10/25 21:30:37 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=296912
                FILE: Number of bytes written=6540706
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=251747
                HDFS: Number of bytes written=104678
                HDFS: Number of read operations=98
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=21
        Job Counters
                Killed map tasks=1
                Launched map tasks=21
                Launched reduce tasks=7
                Data-local map tasks=21
                Total time spent by all maps in occupied slots (ms)=387219
                Total time spent by all reduces in occupied slots (ms)=98778
                Total time spent by all map tasks (ms)=129073
                Total time spent by all reduce tasks (ms)=32926
                Total vcore-milliseconds taken by all map tasks=129073
                Total vcore-milliseconds taken by all reduce tasks=32926
                Total megabyte-milliseconds taken by all map tasks=396512256
                Total megabyte-milliseconds taken by all reduce tasks=101148672
        Map-Reduce Framework
                Map input records=420
                Map output records=17657
                Map output bytes=261556
                Map output materialized bytes=297752
                Input split bytes=2163
                Combine input records=0
                Combine output records=0
                Reduce input groups=6662
                Reduce shuffle bytes=297752
                Reduce input records=17657
                Reduce output records=6662
                Spilled Records=35314
                Shuffled Maps =147
                Failed Shuffles=0
                Merged Map outputs=147
                GC time elapsed (ms)=4482
                CPU time spent (ms)=31790
                Physical memory (bytes) snapshot=13085184000
                Virtual memory (bytes) snapshot=122130706432
                Total committed heap usage (bytes)=11825315840
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=249584
        File Output Format Counters
                Bytes Written=104678
21/10/25 21:30:37 INFO streaming.StreamJob: Output directory: /user/hive/output1
```

Figure 13: Output of Mapper1 and Reducer1 files

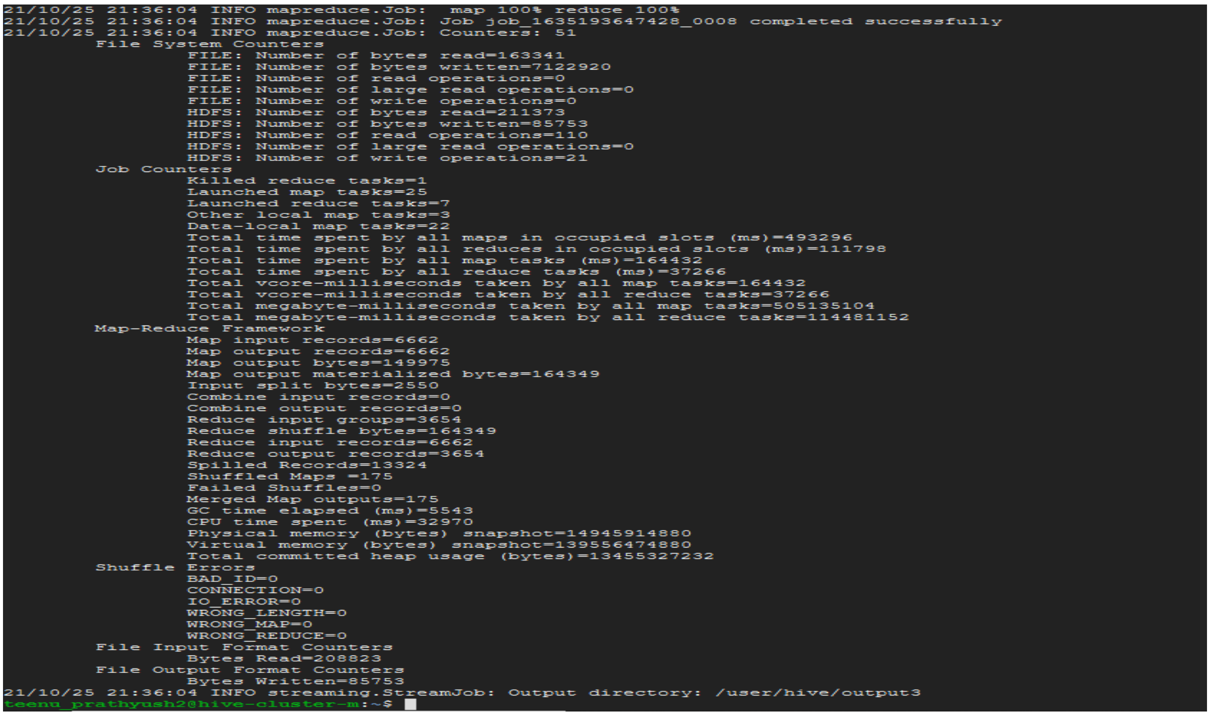
Figure 13 shows the output of the 1st Mapper and 1st Reducer files. The output is stored in HDFS - /user/hive/output1



```
21/10/25 21:33:48 INFO mapreduce.Job:  map 100% reduce 100%
21/10/25 21:33:48 INFO mapreduce.Job: Job job_1635193647428_0007 completed successfully
21/10/25 21:33:48 INFO mapreduce.Job: Counters: 51
        File System Counters
                FILE: Number of bytes read=118044
                FILE: Number of bytes written=6182578
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=164164
                HDFS: Number of bytes written=136651
                HDFS: Number of read operations=98
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=21
        Job Counters
                Killed map tasks=1
                Killed reduce tasks=1
                Launched map tasks=21
                Launched reduce tasks=7
                Data-local map tasks=21
                Total time spent by all maps in occupied slots (ms)=438129
                Total time spent by all reduces in occupied slots (ms)=106770
                Total time spent by all map tasks (ms)=146043
                Total time spent by all reduce tasks (ms)=35590
                Total vcore-milliseconds taken by all map tasks=146043
                Total vcore-milliseconds taken by all reduce tasks=35590
                Total megabyte-milliseconds taken by all map tasks=448644096
                Total megabyte-milliseconds taken by all reduce tasks=109332480
        Map-Reduce Framework
                Map input records=6662
                Map output records=6662
                Map output bytes=104678
                Map output materialized bytes=118884
                Input split bytes=2142
                Combine input records=0
                Combine output records=0
                Reduce input groups=10
                Reduce shuffle bytes=118884
                Reduce input records=6662
                Reduce output records=6662
                Spilled Records=13324
                Shuffled Maps =147
                Failed Shuffles=0
                Merged Map outputs=147
                GC time elapsed (ms)=4954
                CPU time spent (ms)=28790
                Physical memory (bytes) snapshot=13342322688
                Virtual memory (bytes) snapshot=122049286144
                Total committed heap usage (bytes)=11913396224
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=162022
        File Output Format Counters
                Bytes Written=136651
21/10/25 21:33:48 INFO streaming.StreamJob: Output directory: /user/hive/output2
teenu_prathyush2@hive-cluster-m:~$
```

Figure 14: Output of Mapper2 and Reducer2 files

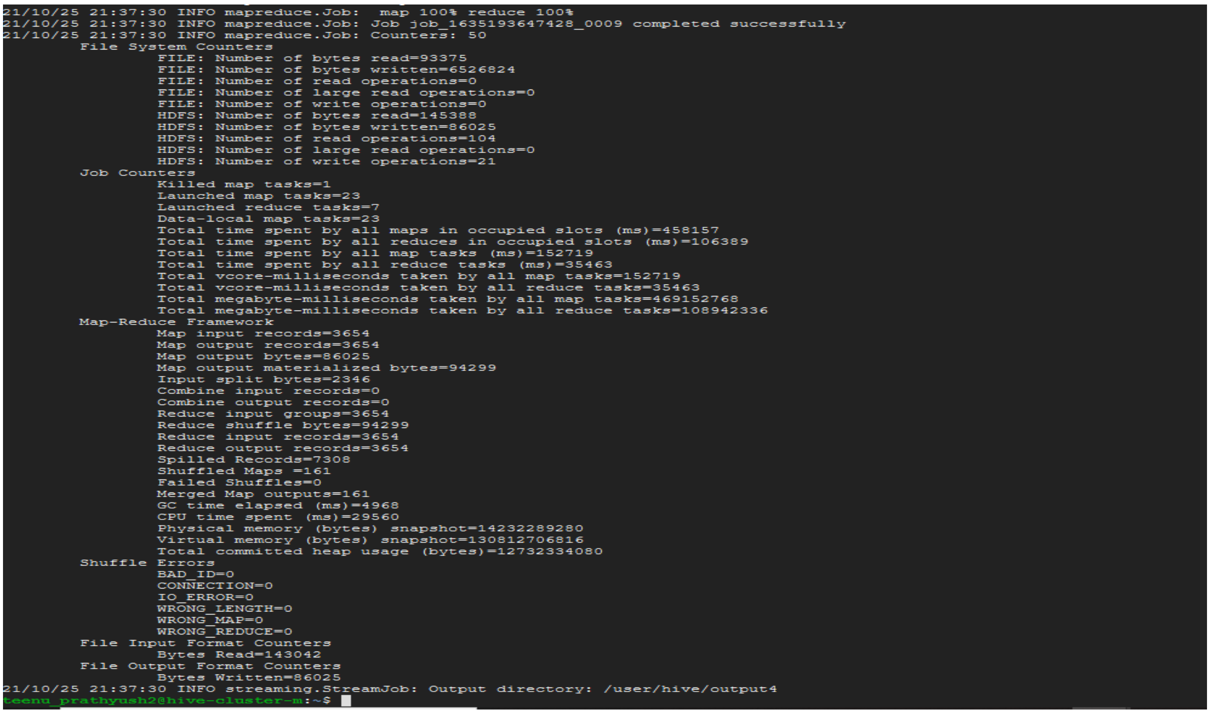
Figure 14 shows the output of the 2nd Mapper and 2nd Reducer files. The output is stored in HDFS - /user/hive/output2



```
21/10/25 21:36:04 INFO mapreduce.Job:  map 100% reduce 100%
21/10/25 21:36:04 INFO mapreduce.Job: Job job_1635193647428_0008 completed successfully
21/10/25 21:36:04 INFO mapreduce.Job: Counters: 51
        File System Counters
                FILE: Number of bytes read=163341
                FILE: Number of bytes written=7122920
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=211373
                HDFS: Number of bytes written=85753
                HDFS: Number of read operations=110
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=21
        Job Counters
                Killed reduce tasks=1
                Launched map tasks=25
                Launched reduce tasks=7
                Other local map tasks=3
                Data-local map tasks=22
                Total time spent by all maps in occupied slots (ms)=493296
                Total time spent by all reduces in occupied slots (ms)=111798
                Total time spent by all map tasks (ms)=164432
                Total time spent by all reduce tasks (ms)=37266
                Total vcore-milliseconds taken by all map tasks=164432
                Total vcore-milliseconds taken by all reduce tasks=37266
                Total megabyte-milliseconds taken by all map tasks=505135104
                Total megabyte-milliseconds taken by all reduce tasks=114481152
        Map-Reduce Framework
                Map input records=6662
                Map output records=6662
                Map output bytes=149975
                Map output materialized bytes=164349
                Input split bytes=2550
                Combine input records=0
                Combine output records=0
                Reduce input groups=3654
                Reduce shuffle bytes=164349
                Reduce input records=6662
                Reduce output records=3654
                Spilled Records=13324
                Shuffled Maps =175
                Failed Shuffles=0
                Merged Map outputs=175
                GC time elapsed (ms)=5543
                CPU time spent (ms)=32970
                Physical memory (bytes) snapshot=14945914880
                Virtual memory (bytes) snapshot=139556474880
                Total committed heap usage (bytes)=13455327232
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=208823
        File Output Format Counters
                Bytes Written=85753
21/10/25 21:36:04 INFO streaming.StreamJob: Output directory: /user/hive/output3
teenu_prathyush2@hive-cluster-m:~$
```

Figure 15: Output of Mapper3 and Reducer3 files

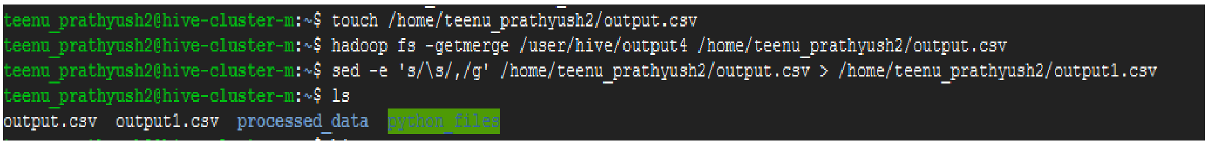
Figure 15 shows the output of the 3rd Mapper and 3rd Reducer files. The output is stored in HDFS - /user/hive/output3



```
21/10/25 21:37:30 INFO mapreduce.Job:  map 100% reduce 100%
21/10/25 21:37:30 INFO mapreduce.Job: Job job_1635193647428_0009 completed successfully
21/10/25 21:37:30 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=93375
                FILE: Number of bytes written=6526824
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=145388
                HDFS: Number of bytes written=86025
                HDFS: Number of read operations=104
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=21
        Job Counters
                Killed map tasks=1
                Launched map tasks=23
                Launched reduce tasks=7
                Data-local map tasks=23
                Total time spent by all maps in occupied slots (ms)=458157
                Total time spent by all reduces in occupied slots (ms)=106389
                Total time spent by all map tasks (ms)=152719
                Total time spent by all reduce tasks (ms)=35463
                Total vcore-milliseconds taken by all map tasks=152719
                Total vcore-milliseconds taken by all reduce tasks=35463
                Total megabyte-milliseconds taken by all map tasks=469152768
                Total megabyte-milliseconds taken by all reduce tasks=108942336
        Map-Reduce Framework
                Map input records=3654
                Map output records=3654
                Map output bytes=86025
                Map output materialized bytes=94299
                Input split bytes=2346
                Combine input records=0
                Combine output records=0
                Reduce input groups=3654
                Reduce shuffle bytes=94299
                Reduce input records=3654
                Reduce output records=3654
                Spilled Records=7308
                Shuffled Maps =161
                Failed Shuffles=0
                Merged Map outputs=161
                GC time elapsed (ms)=4968
                CPU time spent (ms)=29560
                Physical memory (bytes) snapshot=14232289280
                Virtual memory (bytes) snapshot=130812706816
                Total committed heap usage (bytes)=12732334080
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=143042
        File Output Format Counters
                Bytes Written=86025
21/10/25 21:37:30 INFO streaming.StreamJob: Output directory: /user/hive/output4
teenu_prathyush2@hive-cluster-m:~$
```

Figure 16: Output of Mapper4 file

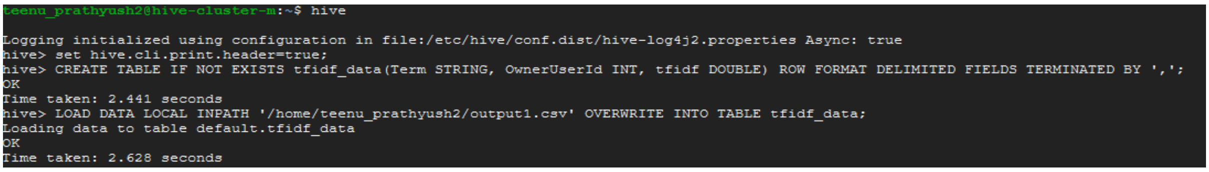
Figure 16 shows the output of the 4th Mapper file. The final output is stored in HDFS - /user/hive/output4



```
teenu_prathyush2@hive-cluster-m:~$ touch /home/teenu_prathyush2/output.csv
teenu_prathyush2@hive-cluster-m:~$ hadoop fs -getmerge /user/hive/output4 /home/teenu_prathyush2/output.csv
teenu_prathyush2@hive-cluster-m:~$ sed -e 's/\s/,/g' /home/teenu_prathyush2/output.csv > /home/teenu_prathyush2/output1.csv
teenu_prathyush2@hive-cluster-m:~$ ls
output.csv  output1.csv  processed_data  python_files
```

Figure 17: Merging the output and replacing the spaces

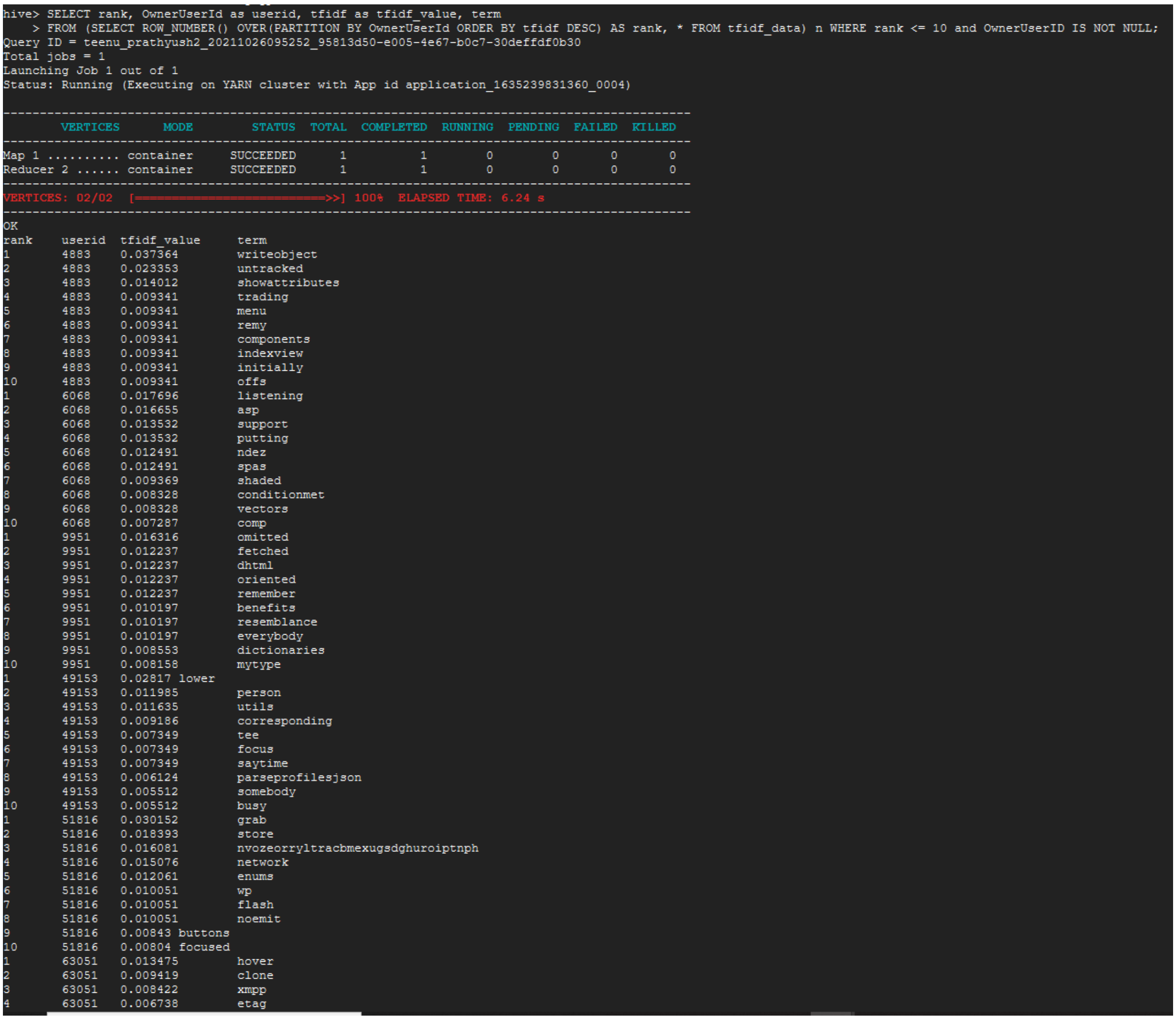
Figure 17 shows the merging of all the output parts into a csv file (output.csv) and then replacing all the spaces and copying it to another csv file (output1.csv).



```
teenu_prathyush2@hive-cluster-m:~$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: true
hive> set hive.cli.print.header=true;
hive> CREATE TABLE IF NOT EXISTS tfidf_data(Term STRING, OwnerUserId INT, tfidf DOUBLE) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
OK
Time taken: 2.441 seconds
hive> LOAD DATA LOCAL INPATH '/home/teenu_prathyush2/output1.csv' OVERWRITE INTO TABLE tfidf_data;
Loading data to table default.tfidf_data
OK
Time taken: 2.628 seconds
```

Figure 18: Loading the output into the table - tfidf_data

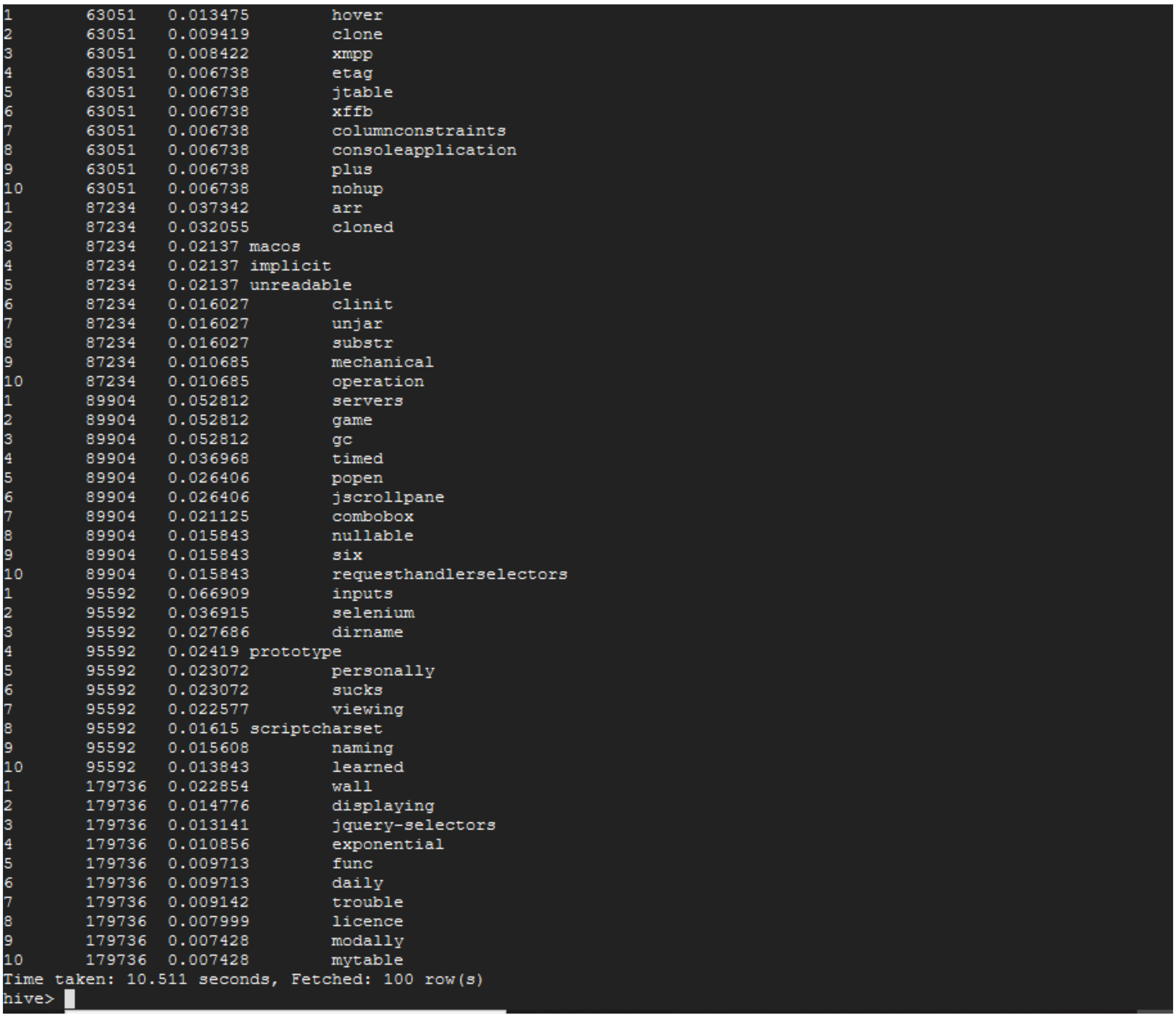
Figure 18 shows the creation of the table – tfidf_data and loading of the contents of output1.csv into tfidf_data table.



```
hive> SELECT rank, OwnerUserId as userid, tfidf as tfidf_value, term
    > FROM (SELECT ROW_NUMBER() OVER(PARTITION BY OwnerUserId ORDER BY tfidf DESC) AS rank, * FROM tfidf_data) n WHERE rank <= 10 and OwnerUserID IS NOT NULL;
Query ID = teenu_prathyush2_20211026095252_95813d50-e005-4e67-b0c7-30deffdf0b30
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635239831360_0004)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.24 s
----------------------------------------------------------------------------------------------
OK
rank    userid  tfidf_value     term
1       4883    0.037364        writeobject
2       4883    0.023353        untracked
3       4883    0.014012        showattributes
4       4883    0.009341        trading
5       4883    0.009341        menu
6       4883    0.009341        remy
7       4883    0.009341        components
8       4883    0.009341        indexview
9       4883    0.009341        initially
10      4883    0.009341        offs
1       6068    0.017696        listening
2       6068    0.016655        asp
3       6068    0.013532        support
4       6068    0.013532        putting
5       6068    0.012491        ndez
6       6068    0.012491        spas
7       6068    0.009369        shaded
8       6068    0.008328        conditionmet
9       6068    0.008328        vectors
10      6068    0.007287        comp
1       9951    0.016316        omitted
2       9951    0.012237        fetched
3       9951    0.012237        dhtml
4       9951    0.012237        oriented
5       9951    0.012237        remember
6       9951    0.010197        benefits
7       9951    0.010197        resemblance
8       9951    0.010197        everybody
9       9951    0.008553        dictionaries
10      9951    0.008158        mytype
1       49153   0.02817 lower
2       49153   0.011985        person
3       49153   0.011635        utils
4       49153   0.009186        corresponding
5       49153   0.007349        tee
6       49153   0.007349        focus
7       49153   0.007349        saytime
8       49153   0.006124        parseprofilesjson
9       49153   0.005512        somebody
10      49153   0.005512        busy
1       51816   0.030152        grab
2       51816   0.018393        store
3       51816   0.016081        nvozeorryltracbmexugsdghuroiptnph
4       51816   0.015076        network
5       51816   0.012061        enums
6       51816   0.010051        wp
7       51816   0.010051        flash
8       51816   0.010051        noemit
9       51816   0.00843 buttons
10      51816   0.00804 focused
1       63051   0.013475        hover
2       63051   0.009419        clone
3       63051   0.008422        xmpp
4       63051   0.006738        etag
```

Figure 19(a): (Task 4) Per-User TFIDF of the top 10 terms for each of the top 10 users

Figure 19(a) shows the Per-User TFIDF values of the top 10 terms for each of the top 10 users.



```
1       63051   0.013475        hover
2       63051   0.009419        clone
3       63051   0.008422        xmpp
4       63051   0.006738        etag
5       63051   0.006738        jtable
6       63051   0.006738        xffb
7       63051   0.006738        columnconstraints
8       63051   0.006738        consoleapplication
9       63051   0.006738        plus
10      63051   0.006738        nohup
1       87234   0.037342        arr
2       87234   0.032055        cloned
3       87234   0.02137 macos
4       87234   0.02137 implicit
5       87234   0.02137 unreadable
6       87234   0.016027        clinit
7       87234   0.016027        unjar
8       87234   0.016027        substr
9       87234   0.010685        mechanical
10      87234   0.010685        operation
1       89904   0.052812        servers
2       89904   0.052812        game
3       89904   0.052812        gc
4       89904   0.036968        timed
5       89904   0.026406        popen
6       89904   0.026406        jscrollpane
7       89904   0.021125        combobox
8       89904   0.015843        nullable
9       89904   0.015843        six
10      89904   0.015843        requesthandlerselectors
1       95592   0.066909        inputs
2       95592   0.036915        selenium
3       95592   0.027686        dirname
4       95592   0.02419 prototype
5       95592   0.023072        personally
6       95592   0.023072        sucks
7       95592   0.022577        viewing
8       95592   0.01615 scriptcharset
9       95592   0.015608        naming
10      95592   0.013843        learned
1       179736  0.022854        wall
2       179736  0.014776        displaying
3       179736  0.013141        jquery-selectors
4       179736  0.010856        exponential
5       179736  0.009713        func
6       179736  0.009713        daily
7       179736  0.009142        trouble
8       179736  0.007999        licence
9       179736  0.007428        modally
10      179736  0.007428        mytable
Time taken: 10.511 seconds, Fetched: 100 row(s)
hive>
```

Figure 19(b): (Task 4) Per-User TFIDF of the top 10 terms for each of the top 10 users

Figure 19(b) shows the Per-User TFIDF values of the top 10 terms for each of the top 10 users.

APPENDIX (SCREENSHOTS)