CA675 ASSIGNMENT 1 – DATA ANALYSIS

Name	**SMIT MEHTA**
Student ID	**20210108**
Email ID	**smit.mehta4@mail.dcu.ie**
Programme Module Code	**CA675**
Github Link	**https://github.com/smitmehta19/CA675-Assignment_1**
Date of Submission	**28/10/2021**

Task 1: Getting data from Stack Exchange

Top 200,000 posts were obtained from Stack exchange (Data Explorer feature) using 4 queries to fetch records each at a time. Text files obtained from the queries were downloaded to perform data analysis.

Getting first 50,000 posts:

```
select top 50000 * from posts where posts.ViewCount > 127150 ORDER BY
posts.ViewCount DESC
```

Next 50,000 posts:

```
select top 50000 * from posts where posts.ViewCount < 127150 AND
posts.ViewCount >74000 ORDER BY posts.ViewCount DESC
```

Next 50,000 posts:

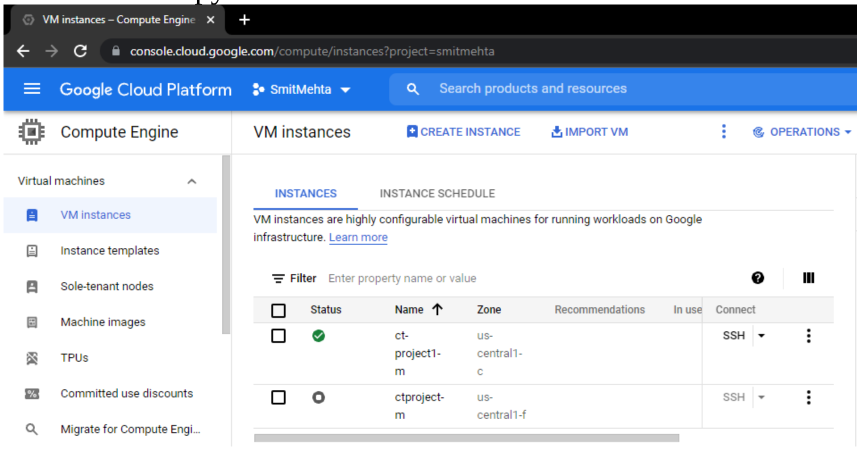
```
select top 50000 * from posts where posts.ViewCount < 74000 AND
posts.ViewCount >53800 ORDER BY posts.ViewCount DESC
```

Last 50,000 posts:

```
select top 50000 * from posts where posts.ViewCount < 53800 AND
posts.ViewCount >30000 ORDER BY posts.ViewCount DESC
```

Task 2: Extract, Transform and load the data

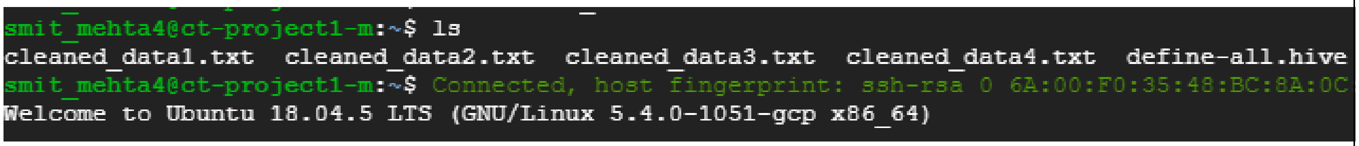
Preprocessed the files downloaded from Stack Exchange to remove few unwanted elements in the raw Text files in Jupyter Notebook. After that created a Data Proc cluster on GCP



VM instances are highly configurable virtual machines for running workloads on Google infrastructure. Learn more

Status	Name	Zone	Recommendations	In use	Connect
✓	ct-project1-m	us-central1-c			SSH
○	ctproject-m	us-central1-f			SSH

CSV files were first uploaded to the cluster like seen below



```
smit_mehta4@ct-project1-m:~$ ls
cleaned_data1.txt  cleaned_data2.txt  cleaned_data3.txt  cleaned_data4.txt  define-all.hive
smit_mehta4@ct-project1-m:~$ Connected, host fingerprint: ssh-rsa 0 6A:00:F0:35:48:BC:8A:0C
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1051-gcp x86_64)
```

Tasks completed next:

- Loaded the text files into pig as (file1, file2, file3, file4) using the LOAD command and stored using PigStorage command
- Created a new file (Combined_Data) which is a union of all the 4 files which were uploaded
- Filtered the unwanted NULL data in OwnerUserId and Score columns. Removed them and stored the data file in Data_Filter file
- Transformed the filtered data using the Generate command and removed the unwanted columns in the data set. Kept 7 useful columns.
- Finally stored the filtered and clean data in Final_Data.txt using PigStorage.

Attaching the code and Screenshot below:

1. Loading Data in PIG

```
file1 = LOAD 'cleaned_data1.txt' USING PigStorage(',')
AS(dummy:chararray,Id:chararray, PostTypeId:chararray,
AcceptedAnswerId:chararray, ParentId:chararray, CreationDate:chararray,
DeletionDate:chararray, Score:int, ViewCount:int,Body:chararray,
OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray,
LastEditorDisplayName:chararray, LastEditDate:chararray,
LastActivityDate:chararray, Title:chararray, Tags:chararray,
AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray,
ClosedDate:chararray, CommunityOwnedDate:chararray, ContentLicense:chararray);
```

```
file2 = LOAD 'cleaned_data2.txt' USING PigStorage(',')
AS(dummy:chararray,Id:chararray, PostTypeId:chararray,
AcceptedAnswerId:chararray, ParentId:chararray, CreationDate:chararray,
DeletionDate:chararray, Score:int, ViewCount:int,Body:chararray,
OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray,
LastEditorDisplayName:chararray, LastEditDate:chararray,
LastActivityDate:chararray, Title:chararray, Tags:chararray,
AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray,
ClosedDate:chararray, CommunityOwnedDate:chararray, ContentLicense:chararray);
```

```
file3 = LOAD 'cleaned_data3.txt' USING PigStorage(',')
AS(dummy:chararray,Id:chararray, PostTypeId:chararray,
AcceptedAnswerId:chararray, ParentId:chararray, CreationDate:chararray,
DeletionDate:chararray, Score:int, ViewCount:int,Body:chararray,
OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray,
LastEditorDisplayName:chararray, LastEditDate:chararray,
LastActivityDate:chararray, Title:chararray, Tags:chararray,
AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray,
ClosedDate:chararray, CommunityOwnedDate:chararray, ContentLicense:chararray);
```

```
file4 = LOAD 'cleaned_data4.txt' USING PigStorage(',')
AS(dummy:chararray,Id:chararray, PostTypeId:chararray,
AcceptedAnswerId:chararray, ParentId:chararray, CreationDate:chararray,
DeletionDate:chararray, Score:int, ViewCount:int,Body:chararray,
OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray,
LastEditorDisplayName:chararray, LastEditDate:chararray,
LastActivityDate:chararray, Title:chararray, Tags:chararray,
AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray,
ClosedDate:chararray, CommunityOwnedDate:chararray, ContentLicense:chararray);
```

2. Concatenating all the four files:

```
Combined_Data  = UNION  file1,file2,file3,file4;
```

3. Removing entries with null values:

```
Data_Filter = filter Combined_Data by (OwnerUserId is not null) and
(Score is not null);
```

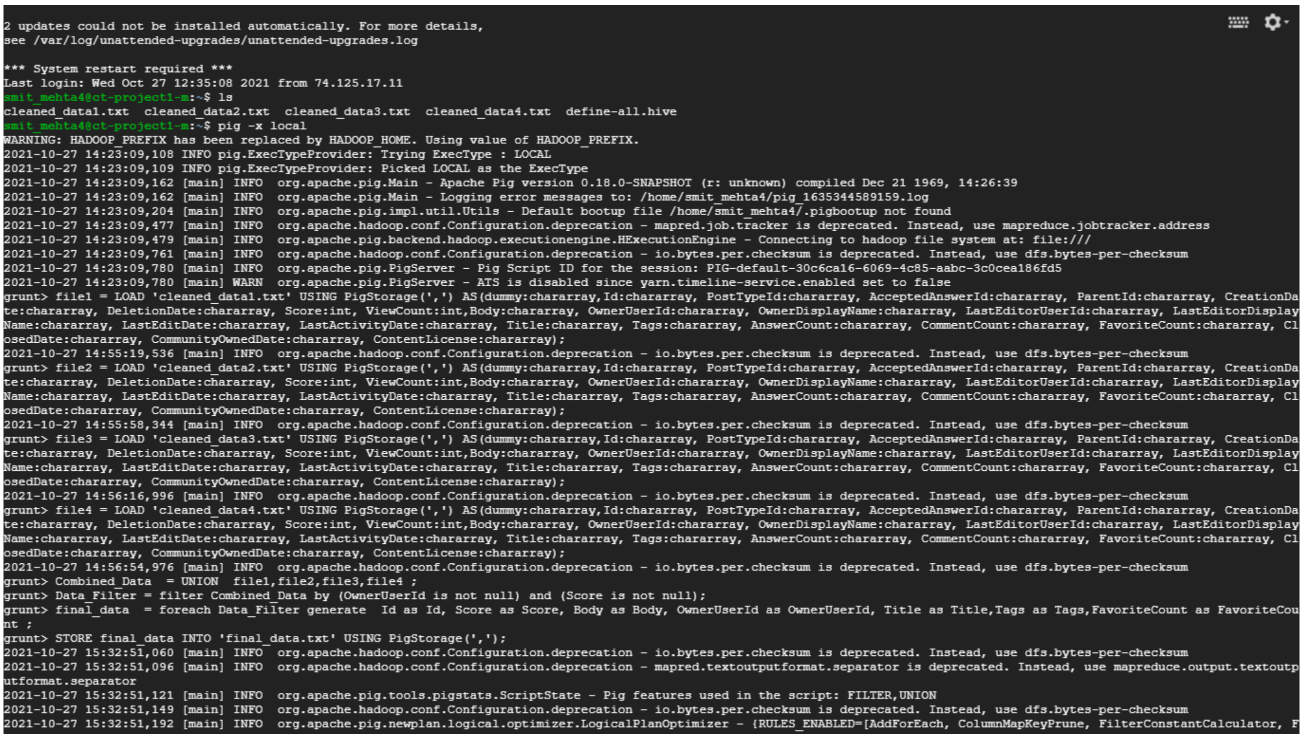
4. Transforming and removing unwanted columns:

```
final_data  = foreach Data_Filter generate  Id as Id, Score as Score,
Body as Body, OwnerUserId as OwnerUserId, Title as Title,Tags as
Tags,FavoriteCount as FavoriteCount ;
```

5. Storing the final result into pig:

```
STORE final_data INTO 'final_data.txt' USING PigStorage(',');
```

Proof in the form of screenshot for the above-mentioned tasks is given below



```
2 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

*** System restart required ***
Last login: Wed Oct 27 12:35:08 2021 from 74.125.17.11
smit_mehta4@ct-project1-m:~$ ls
cleaned_data1.txt  cleaned_data2.txt  cleaned_data3.txt  cleaned_data4.txt  define-all.hive
smit_mehta4@ct-project1-m:~$ pig -x local
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
2021-10-27 14:23:09,108 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2021-10-27 14:23:09,109 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2021-10-27 14:23:09,162 [main] INFO  org.apache.pig.Main - Apache Pig version 0.18.0-SNAPSHOT (r: unknown) compiled Dec 21 1969, 14:26:39
2021-10-27 14:23:09,162 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/smit_mehta4/pig_1635344589159.log
2021-10-27 14:23:09,204 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/smit_mehta4/.pigbootup not found
2021-10-27 14:23:09,477 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2021-10-27 14:23:09,479 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: file:///
2021-10-27 14:23:09,761 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2021-10-27 14:23:09,780 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-30c6ca16-6069-4c85-aabc-3c0cea186fd5
2021-10-27 14:23:09,780 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt> file1 = LOAD 'cleaned_data1.txt' USING PigStorage(',') AS(dummy:chararray,Id:chararray, PostTypeId:chararray, AcceptedAnswerId:chararray, ParentId:chararray, CreationDa
te:chararray, DeletionDate:chararray, Score:int, ViewCount:int,Body:chararray, OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray, LastEditorDisplay
Name:chararray, LastEditDate:chararray, LastActivityDate:chararray, Title:chararray, Tags:chararray, AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray, Cl
osedDate:chararray, CommunityOwnedDate:chararray, ContentLicense:chararray);
2021-10-27 14:55:19,536 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> file2 = LOAD 'cleaned_data2.txt' USING PigStorage(',') AS(dummy:chararray,Id:chararray, PostTypeId:chararray, AcceptedAnswerId:chararray, ParentId:chararray, CreationDa
te:chararray, DeletionDate:chararray, Score:int, ViewCount:int,Body:chararray, OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray, LastEditorDisplay
Name:chararray, LastEditDate:chararray, LastActivityDate:chararray, Title:chararray, Tags:chararray, AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray, Cl
osedDate:chararray, CommunityOwnedDate:chararray, ContentLicense:chararray);
2021-10-27 14:55:58,344 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> file3 = LOAD 'cleaned_data3.txt' USING PigStorage(',') AS(dummy:chararray,Id:chararray, PostTypeId:chararray, AcceptedAnswerId:chararray, ParentId:chararray, CreationDa
te:chararray, DeletionDate:chararray, Score:int, ViewCount:int,Body:chararray, OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray, LastEditorDisplay
Name:chararray, LastEditDate:chararray, LastActivityDate:chararray, Title:chararray, Tags:chararray, AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray, Cl
osedDate:chararray, CommunityOwnedDate:chararray, ContentLicense:chararray);
2021-10-27 14:56:16,996 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> file4 = LOAD 'cleaned_data4.txt' USING PigStorage(',') AS(dummy:chararray,Id:chararray, PostTypeId:chararray, AcceptedAnswerId:chararray, ParentId:chararray, CreationDa
te:chararray, DeletionDate:chararray, Score:int, ViewCount:int,Body:chararray, OwnerUserId:chararray, OwnerDisplayName:chararray, LastEditorUserId:chararray, LastEditorDisplay
Name:chararray, LastEditDate:chararray, LastActivityDate:chararray, Title:chararray, Tags:chararray, AnswerCount:chararray, CommentCount:chararray, FavoriteCount:chararray, Cl
osedDate:chararray, CommunityOwnedDate:chararray, ContentLicense:chararray);
2021-10-27 14:56:54,976 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> Combined_Data  = UNION  file1,file2,file3,file4 ;
grunt> Data_Filter = filter Combined_Data by (OwnerUserId is not null) and (Score is not null);
grunt> final_data  = foreach Data_Filter generate  Id as Id, Score as Score, Body as Body, OwnerUserId as OwnerUserId, Title as Title,Tags as Tags,FavoriteCount as FavoriteCou
nt ;
grunt> STORE final_data INTO 'final_data.txt' USING PigStorage(',');
2021-10-27 15:32:51,060 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2021-10-27 15:32:51,096 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.textoutputformat.separator is deprecated. Instead, use mapreduce.output.textoutp
utformat.separator
2021-10-27 15:32:51,121 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: FILTER,UNION
2021-10-27 15:32:51,149 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2021-10-27 15:32:51,192 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, FilterConstantCalculator, F
```

6. Copied the 'final_data.txt' from local to HDFS using the following command:

```
hdfs dfs -copyFromLocal /home/smit_mehta4/final_data.txt/
```

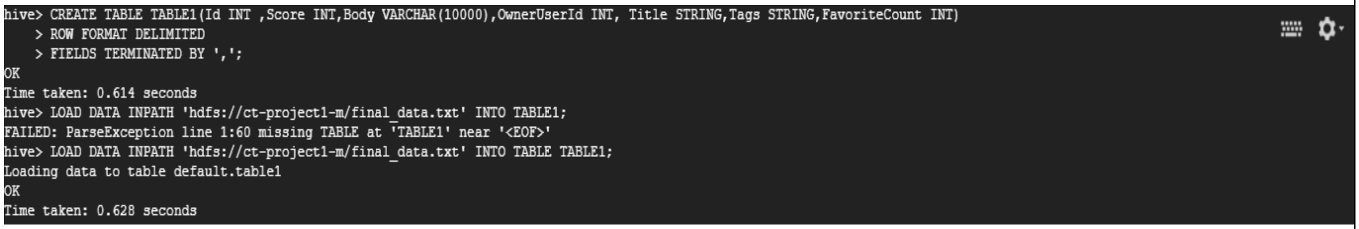
Source : http://pig.apache.org/docs/r0.17.0/basic.html

Task 3: Performing Hive Queries:

Connecting to hive, creating a table & loading the data:

```
CREATE TABLE TABLE1(Id INT ,Score INT,Body VARCHAR(10000),OwnerUserId
INT, Title STRING,Tags STRING,FavoriteCount INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

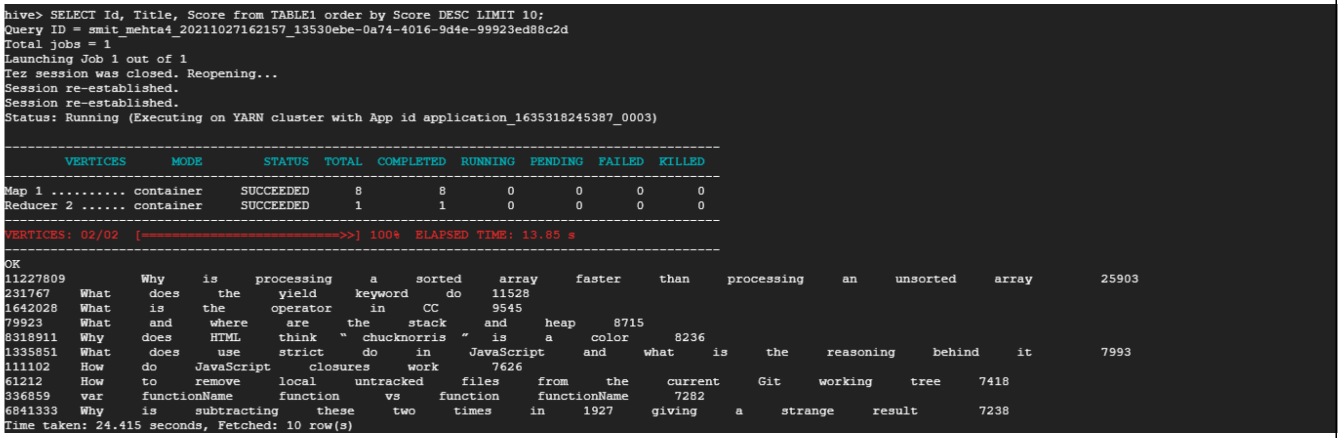
```
LOAD DATA INPATH 'hdfs://ct-project1-m/final_data.txt' INTO TABLE
TABLE1;
```



```
hive> CREATE TABLE TABLE1(Id INT ,Score INT,Body VARCHAR(10000),OwnerUserId INT, Title STRING,Tags STRING,FavoriteCount INT)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ',';
OK
Time taken: 0.614 seconds
hive> LOAD DATA INPATH 'hdfs://ct-project1-m/final_data.txt' INTO TABLE1;
FAILED: ParseException line 1:60 missing TABLE at 'TABLE1' near '<EOF>'
hive> LOAD DATA INPATH 'hdfs://ct-project1-m/final_data.txt' INTO TABLE TABLE1;
Loading data to table default.table1
OK
Time taken: 0.628 seconds
```

1. To get top 10 posts by score:

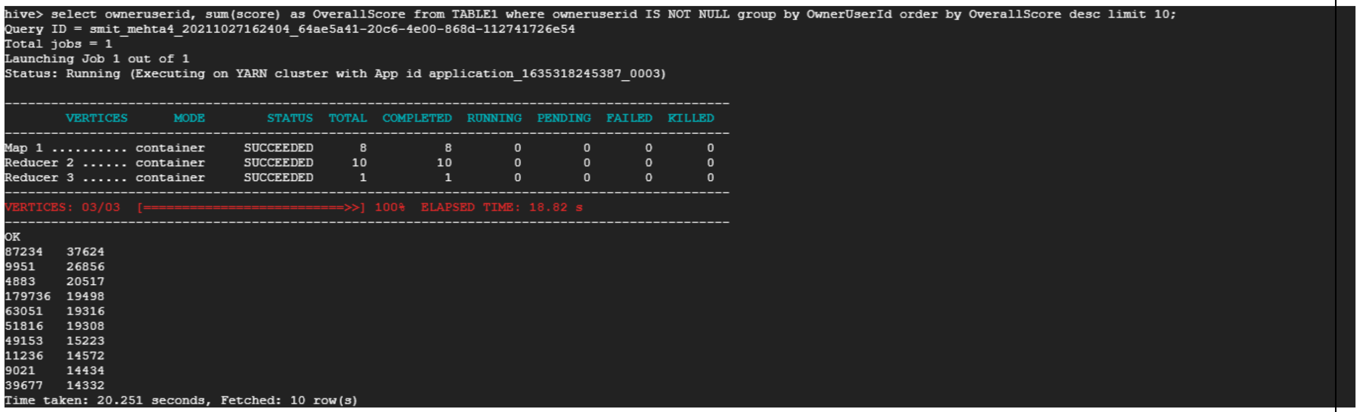
```
SELECT Id, Title, Score from TABLE1 order by Score DESC LIMIT
10;;
```



```
hive> SELECT Id, Title, Score from TABLE1 order by Score DESC LIMIT 10;
Query ID = smit_mehta4_20211027162157_13530ebe-0a74-4016-9d4e-99923ed88c2d
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1635318245387_0003)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 13.85 s
----------------------------------------------------------------------------------------------
OK
11227809         Why    is    processing    a    sorted    array    faster    than    processing    an    unsorted    array         25903
231767    What    does    the    yield    keyword    do    11528
1642028   What    is    the    operator    in    CC    9545
79923     What    and    where    are    the    stack    and    heap    8715
8318911   Why    does    HTML    think  "  chucknorris  "  is    a    color    8236
1335851   What    does    use    strict    do    in    JavaScript    and    what    is    the    reasoning    behind    it         7993
111102    How    do    JavaScript    closures    work    7626
61212     How    to    remove    local    untracked    files    from    the    current    Git    working    tree    7418
336859    var    functionName    function    vs    function    functionName    7282
6841333   Why    is    subtracting    these    two    times    in    1927    giving    a    strange    result         7238
Time taken: 24.415 seconds, Fetched: 10 row(s)
```

2. To get top 10 users by post score:

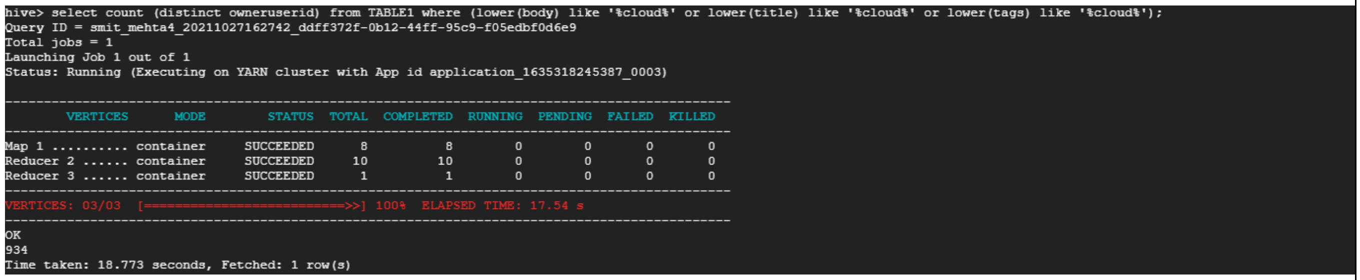
```
select owneruserid, sum(score) as OverallScore from TABLE1 where
owneruserid IS NOT NULL group by OwnerUserId order by OverallScore
desc limit 10;
```



```
hive> select owneruserid, sum(score) as OverallScore from TABLE1 where owneruserid IS NOT NULL group by OwnerUserId order by OverallScore desc limit 10;
Query ID = smit_mehta4_20211027162404_64ae5a41-20c6-4e00-868d-112741726e54
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635318245387_0003)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 18.82 s
----------------------------------------------------------------------------------------------
OK
87234   37624
9951    26856
4883    20517
179736  19498
63051   19316
51816   19308
49153   15223
11236   14572
9021    14434
39677   14332
Time taken: 20.251 seconds, Fetched: 10 row(s)
```

3. To get the number of distinct users who used the word “cloud” in one of their posts:

```
select count (distinct owneruserid) from TABLE1 where (lower(body)
like '%cloud%' or lower(title) like '%cloud%' or lower(tags) like
'%cloud%');
```



```
hive> select count (distinct owneruserid) from TABLE1 where (lower(body) like '%cloud%' or lower(title) like '%cloud%' or lower(tags) like '%cloud%');
Query ID = smit_mehta4_20211027162742_ddff372f-0b12-44ff-95c9-f05edbf0d6e9
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635318245387_0003)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 17.54 s
----------------------------------------------------------------------------------------------
OK
934
Time taken: 18.773 seconds, Fetched: 1 row(s)
```

Source : https://cwiki.apache.org/confluence/display/HIVE
https://cwiki.apache.org/confluence/display/Hive/HCatalog+CLI

Task 4: Calculating TF-IDF

```
create temporary macro max2(x INT, y INT) if(x>y,x,y);
create temporary macro tfidf(tf FLOAT, df_t INT, n_docs INT) tf * (log(10,
CAST(n_docs as FLOAT)/max2(1,df_t)) + 1.0);
```

```
create table Distinct_owner_Id as SELECT OwnerUserId, SUM(Score) AS
TotalScore FROM TABLE1 GROUP BY OwnerUserId ORDER BY TotalScore DESC LIMIT
10;
```

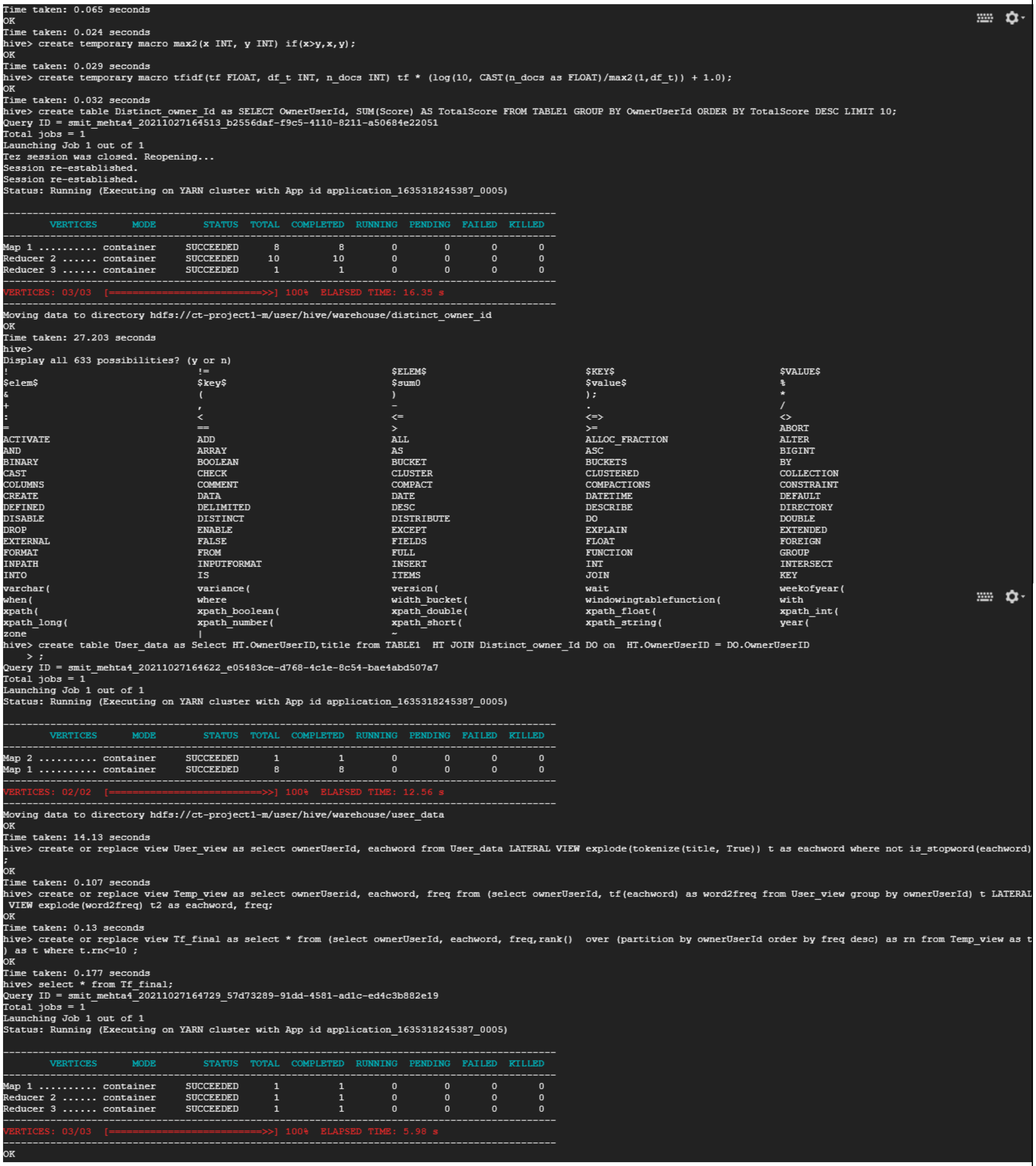
```
create table User_data as Select HT.OwnerUserID,title from TABLE1  HT JOIN
Distinct_owner_Id DO on  HT.OwnerUserID = DO.OwnerUserID;
```

```
create or replace view User_view as select ownerUserId, eachword from
User_data LATERAL VIEW explode(tokenize(title, True)) t as eachword where
not is_stopword(eachword);
```

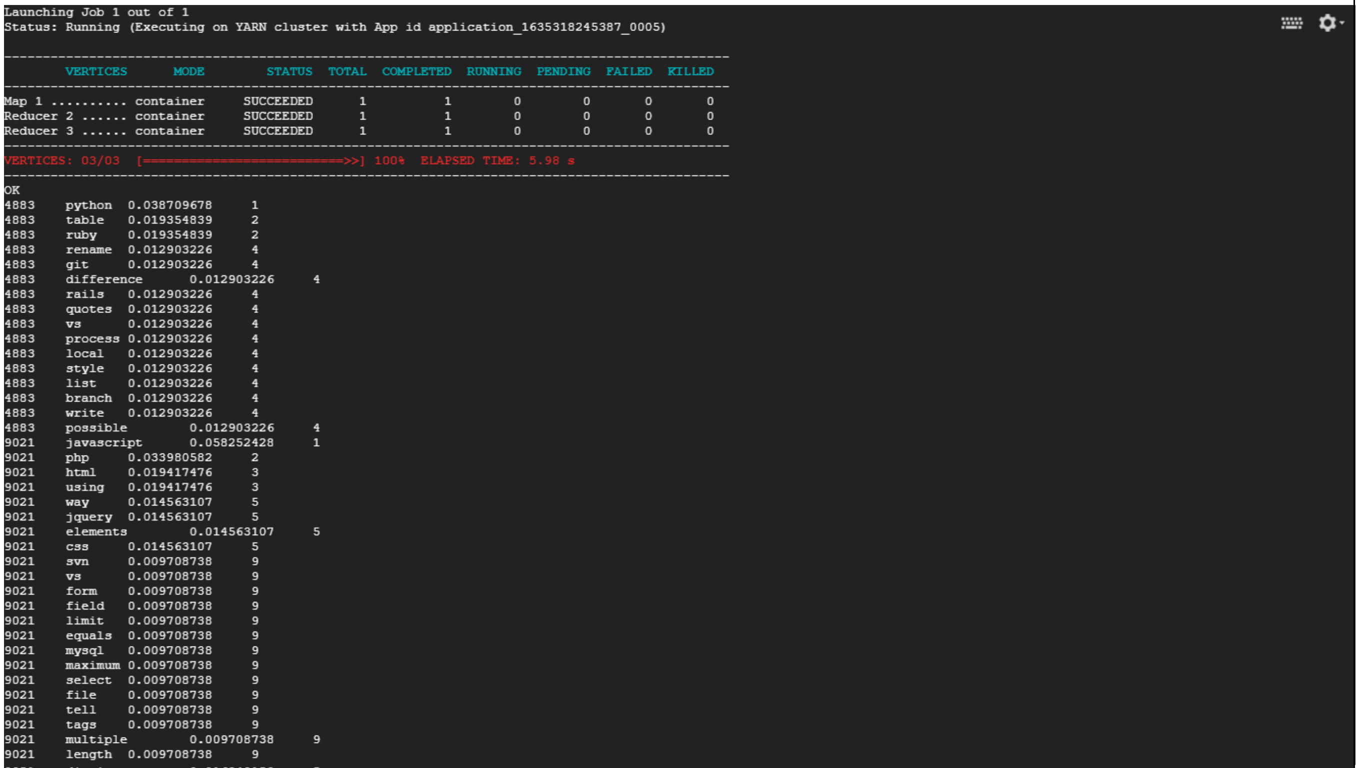
```
create or replace view Temp_view as select ownerUserid, eachword, freq from
(select ownerUserId, tf(eachword) as word2freq from User_view group by
ownerUserId) t LATERAL VIEW explode(word2freq) t2 as eachword, freq;
```

```
create or replace view Tf_final as select * from (select ownerUserId,
eachword, freq,rank()  over (partition by ownerUserId order by freq desc) as
rn from Temp_view as t) as t where t.rn<=10 ;
```

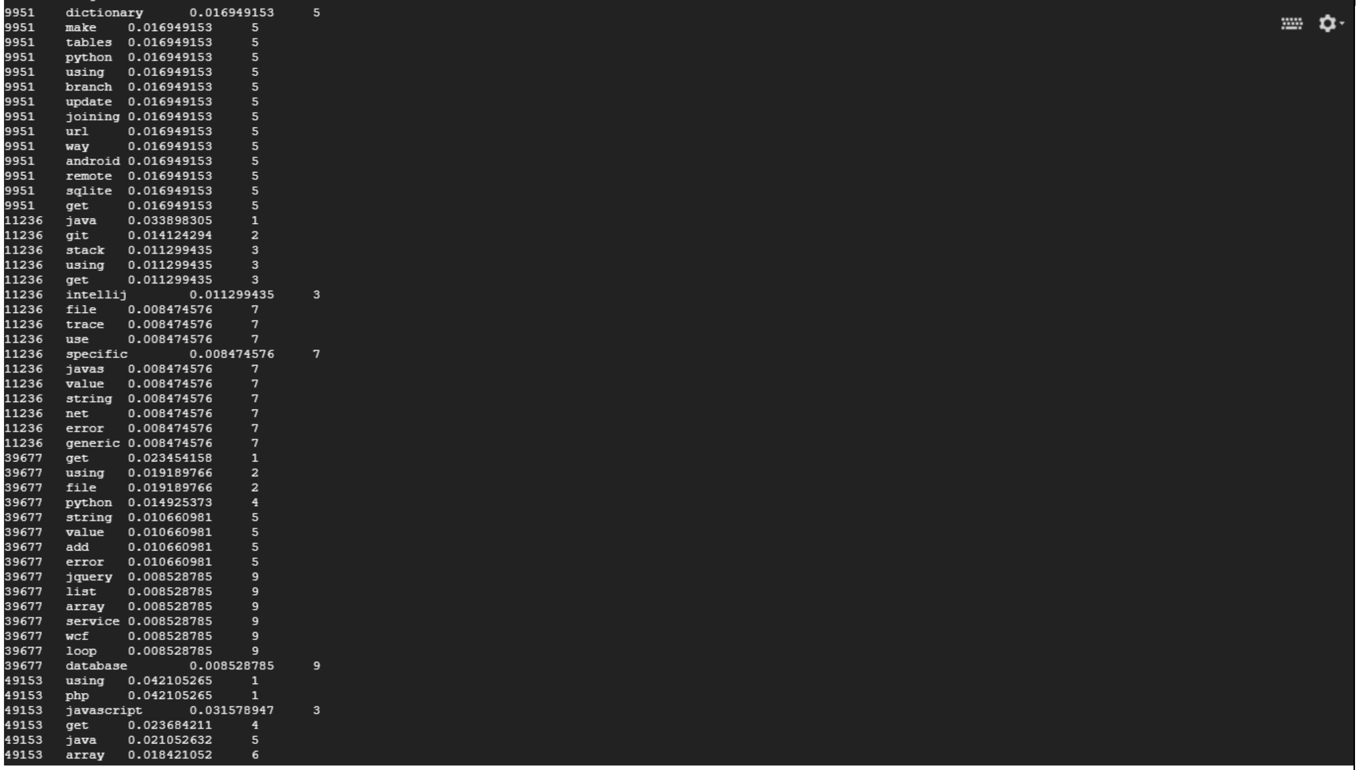
```
select * from Tf_final;
```



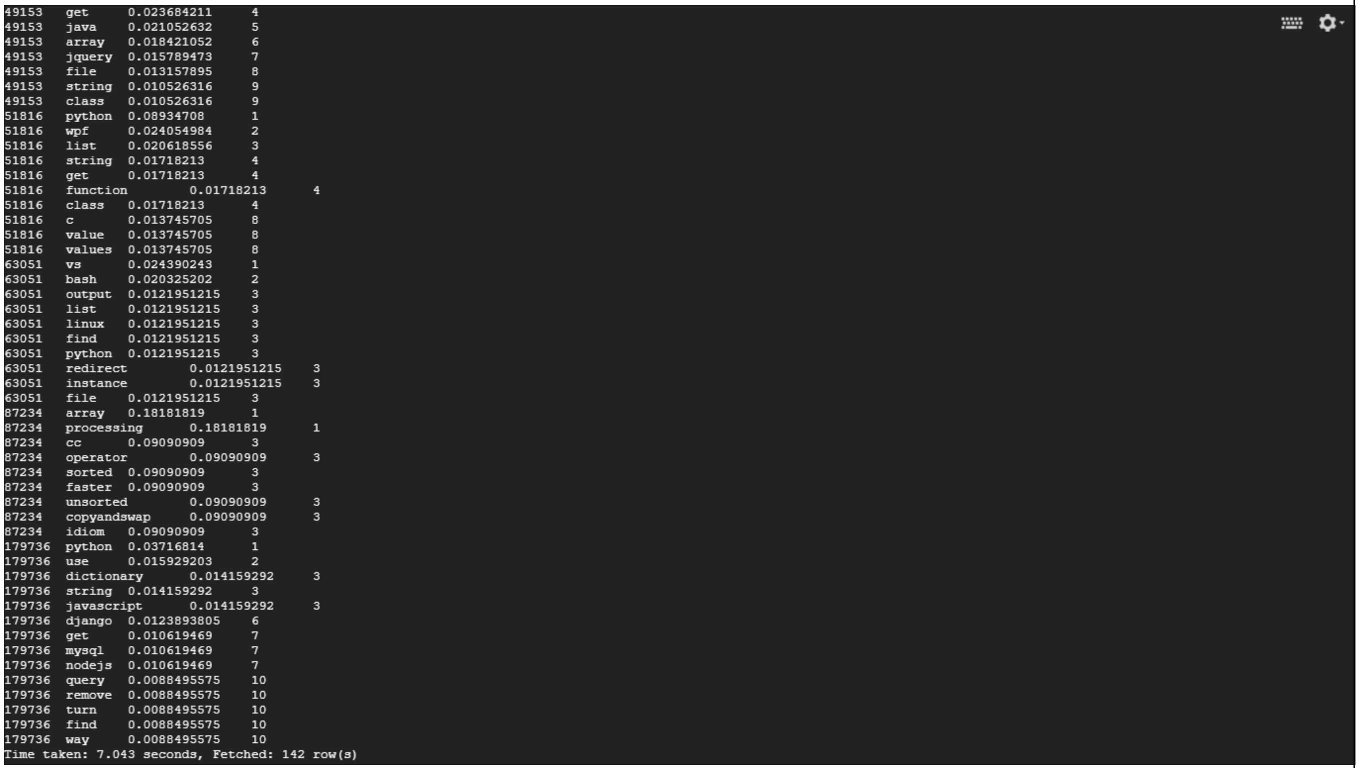
```
Time taken: 0.065 seconds
OK
Time taken: 0.024 seconds
hive> create temporary macro max2(x INT, y INT) if(x>y,x,y);
OK
Time taken: 0.029 seconds
hive> create temporary macro tfidf(tf FLOAT, df_t INT, n_docs INT) tf * (log(10, CAST(n_docs as FLOAT)/max2(1,df_t)) + 1.0);
OK
Time taken: 0.032 seconds
hive> create table Distinct_owner_Id as SELECT OwnerUserId, SUM(Score) AS TotalScore FROM TABLE1 GROUP BY OwnerUserId ORDER BY TotalScore DESC LIMIT 10;
Query ID = smit_mehta4_20211027164513_b2556daf-f9c5-4110-8211-a50684e22051
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1635318245387_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     10         10        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 16.35 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ct-project1-m/user/hive/warehouse/distinct_owner_id
OK
Time taken: 27.203 seconds
hive>
Display all 633 possibilities? (y or n)
!                                 !=                                $ELEM$                            $KEY$                             $VALUE$
$elem$                            $key$                             $sum0                             $value$                           %
&                                 (                                 )                                 );                                *
+                                 ,                                 -                                 .                                 /
:                                 <                                 <=                                <=>                               <>
=                                 ==                                >                                 >=                                ABORT
ACTIVATE                          ADD                               ALL                               ALLOC_FRACTION                    ALTER
AND                               ARRAY                             AS                                ASC                               BIGINT
BINARY                            BOOLEAN                           BUCKET                            BUCKETS                           BY
CAST                              CHECK                             CLUSTER                           CLUSTERED                         COLLECTION
COLUMNS                           COMMENT                           COMPACT                           COMPACTIONS                       CONSTRAINT
CREATE                            DATA                              DATE                              DATETIME                          DEFAULT
DEFINED                           DELIMITED                         DESC                              DESCRIBE                          DIRECTORY
DISABLE                           DISTINCT                          DISTRIBUTE                        DO                                DOUBLE
DROP                              ENABLE                            EXCEPT                            EXPLAIN                           EXTENDED
EXTERNAL                          FALSE                             FIELDS                            FLOAT                             FOREIGN
FORMAT                            FROM                              FULL                              FUNCTION                          GROUP
INPATH                            INPUTFORMAT                       INSERT                            INT                               INTERSECT
INTO                              IS                                ITEMS                             JOIN                              KEY
varchar(                          variance(                         version(                          wait                              weekofyear(
when(                             where                             width_bucket(                     windowingtablefunction(           with
xpath(                            xpath_boolean(                    xpath_double(                     xpath_float(                      xpath_int(
xpath_long(                       xpath_number(                     xpath_short(                      xpath_string(                     year(
zone                              |                                 ~
hive> create table User_data as Select HT.OwnerUserID,title from TABLE1  HT JOIN Distinct_owner_Id DO on  HT.OwnerUserID = DO.OwnerUserID
    > ;
Query ID = smit_mehta4_20211027164622_e05483ce-d768-4c1e-8c54-bae4abd507a7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635318245387_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 2 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 12.56 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ct-project1-m/user/hive/warehouse/user_data
OK
Time taken: 14.13 seconds
hive> create or replace view User_view as select ownerUserId, eachword from User_data LATERAL VIEW explode(tokenize(title, True)) t as eachword where not is_stopword(eachword)
;
OK
Time taken: 0.107 seconds
hive> create or replace view Temp_view as select ownerUserid, eachword, freq from (select ownerUserId, tf(eachword) as word2freq from User_view group by ownerUserId) t LATERAL
 VIEW explode(word2freq) t2 as eachword, freq;
OK
Time taken: 0.13 seconds
hive> create or replace view Tf_final as select * from (select ownerUserId, eachword, freq,rank()  over (partition by ownerUserId order by freq desc) as rn from Temp_view as t
) as t where t.rn<=10 ;
OK
Time taken: 0.177 seconds
hive> select * from Tf_final;
Query ID = smit_mehta4_20211027164729_57d73289-91dd-4581-ad1c-ed4c3b882e19
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635318245387_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
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 5.98 s
----------------------------------------------------------------------------------------------
OK
```



```
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635318245387_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
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 5.98 s
----------------------------------------------------------------------------------------------
OK
4883    python  0.038709678     1
4883    table   0.019354839     2
4883    ruby    0.019354839     2
4883    rename  0.012903226     4
4883    git     0.012903226     4
4883    difference      0.012903226     4
4883    rails   0.012903226     4
4883    quotes  0.012903226     4
4883    vs      0.012903226     4
4883    process 0.012903226     4
4883    local   0.012903226     4
4883    style   0.012903226     4
4883    list    0.012903226     4
4883    branch  0.012903226     4
4883    write   0.012903226     4
4883    possible        0.012903226     4
9021    javascript      0.058252428     1
9021    php     0.033980582     2
9021    html    0.019417476     3
9021    using   0.019417476     3
9021    way     0.014563107     5
9021    jquery  0.014563107     5
9021    elements        0.014563107     5
9021    css     0.014563107     5
9021    svn     0.009708738     9
9021    vs      0.009708738     9
9021    form    0.009708738     9
9021    field   0.009708738     9
9021    limit   0.009708738     9
9021    equals  0.009708738     9
9021    mysql   0.009708738     9
9021    maximum 0.009708738     9
9021    select  0.009708738     9
9021    file    0.009708738     9
9021    tell    0.009708738     9
9021    tags    0.009708738     9
9021    multiple        0.009708738     9
9021    length  0.009708738     9
```



```
9951    dictionary      0.016949153     5
9951    make    0.016949153     5
9951    tables  0.016949153     5
9951    python  0.016949153     5
9951    using   0.016949153     5
9951    branch  0.016949153     5
9951    update  0.016949153     5
9951    joining 0.016949153     5
9951    url     0.016949153     5
9951    way     0.016949153     5
9951    android 0.016949153     5
9951    remote  0.016949153     5
9951    sqlite  0.016949153     5
9951    get     0.016949153     5
11236   java    0.033898305     1
11236   git     0.014124294     2
11236   stack   0.011299435     3
11236   using   0.011299435     3
11236   get     0.011299435     3
11236   intellij        0.011299435     3
11236   file    0.008474576     7
11236   trace   0.008474576     7
11236   use     0.008474576     7
11236   specific        0.008474576     7
11236   javas   0.008474576     7
11236   value   0.008474576     7
11236   string  0.008474576     7
11236   net     0.008474576     7
11236   error   0.008474576     7
11236   generic 0.008474576     7
39677   get     0.023454158     1
39677   using   0.019189766     2
39677   file    0.019189766     2
39677   python  0.014925373     4
39677   string  0.010660981     5
39677   value   0.010660981     5
39677   add     0.010660981     5
39677   error   0.010660981     5
39677   jquery  0.008528785     9
39677   list    0.008528785     9
39677   array   0.008528785     9
39677   service 0.008528785     9
39677   wcf     0.008528785     9
39677   loop    0.008528785     9
39677   database        0.008528785     9
49153   using   0.042105265     1
49153   php     0.042105265     1
49153   javascript      0.031578947     3
49153   get     0.023684211     4
49153   java    0.021052632     5
49153   array   0.018421052     6
```



```
49153   get     0.023684211     4
49153   java    0.021052632     5
49153   array   0.018421052     6
49153   jquery  0.015789473     7
49153   file    0.013157895     8
49153   string  0.010526316     9
49153   class   0.010526316     9
51816   python  0.08934708      1
51816   wpf     0.024054984     2
51816   list    0.020618556     3
51816   string  0.01718213      4
51816   get     0.01718213      4
51816   function        0.01718213      4
51816   class   0.01718213      4
51816   c       0.013745705     8
51816   value   0.013745705     8
51816   values  0.013745705     8
63051   vs      0.024390243     1
63051   bash    0.020325202     2
63051   output  0.0121951215    3
63051   list    0.0121951215    3
63051   linux   0.0121951215    3
63051   find    0.0121951215    3
63051   python  0.0121951215    3
63051   redirect        0.0121951215    3
63051   instance        0.0121951215    3
63051   file    0.0121951215    3
87234   array   0.18181819      1
87234   processing      0.18181819      1
87234   cc      0.09090909      3
87234   operator        0.09090909      3
87234   sorted  0.09090909      3
87234   faster  0.09090909      3
87234   unsorted        0.09090909      3
87234   copyandswap     0.09090909      3
87234   idiom   0.09090909      3
179736  python  0.03716814      1
179736  use     0.015929203     2
179736  dictionary      0.014159292     3
179736  string  0.014159292     3
179736  javascript      0.014159292     3
179736  django  0.0123893805    6
179736  get     0.010619469     7
179736  mysql   0.010619469     7
179736  nodejs  0.010619469     7
179736  query   0.0088495575    10
179736  remove  0.0088495575    10
179736  turn    0.0088495575    10
179736  find    0.0088495575    10
179736  way     0.0088495575    10
Time taken: 7.043 seconds, Fetched: 142 row(s)
```

Source : https://medium.com/analytics-vidhya/tf-idf-term-frequency-technique-easiest-explanation-for-text-classification-in-nlp-with-code-8ca3912e58c3
https://www.youtube.com/watch?v=vZAXpvHhQow
https://courses.cs.ut.ee/MTAT.08.036/2017_fall/uploads/Main/L4_Pig_2017.pdf