Student ID:	20210305
Student name:	Prasad Govardhankar
Student email	prasad.govardhankar2@mail.dcu.ie
Programme	Masters in Computing
Chosen major:	Data Analytics
Module Code	CA675 Cloud Technologies
Date of Submission	20-11-2020

Data Acquisition:

We are need to get the **top 200,000 posts by viewCount** from the Stack Exchange site and it only allow us to download 50.000 records at a time.

To obtain it we need to run at least 4-5 queries in stack exchange site. At first, we need to figure out the lower and upper limit which gives 200000 posts. After multiple attempts I figured out the value greater than 36684 in “ViewCount” gives us 200002 records, hence we can say that by putting lower limit 36684 we can safely get 200000 data records. Also, From the last few records fetched from each query, we can obtain the upper bound in “ViewCount” to use for the next query.

```
select count(*) from posts where posts.ViewCount > 36684
```

This gives – 200002

Further, we can only obtain 50,000 records at a time, we can break down the whole range of “ViewCount” greater than 36684” into at least 4 parts. And each of which has 50,000 records. In order to do that, we arrange them in a descending order.

For TOP 25%

```
select top 50000 * from posts where posts.ViewCount > 36684 ORDER BY posts.ViewCount DESC
```

For Next 25%

```
select top 50000 * from posts where posts.ViewCount <= 112210 ORDER BY posts.ViewCount
DESC
```

For Next 25%

```
select top 50000 * from posts where posts.ViewCount <= 66058 ORDER
BY posts.ViewCount DESC
```

And for remaining 25%

```
select top 50000 * from posts where posts.ViewCount <= 47163 ORDER
BY posts.ViewCount DESC
```

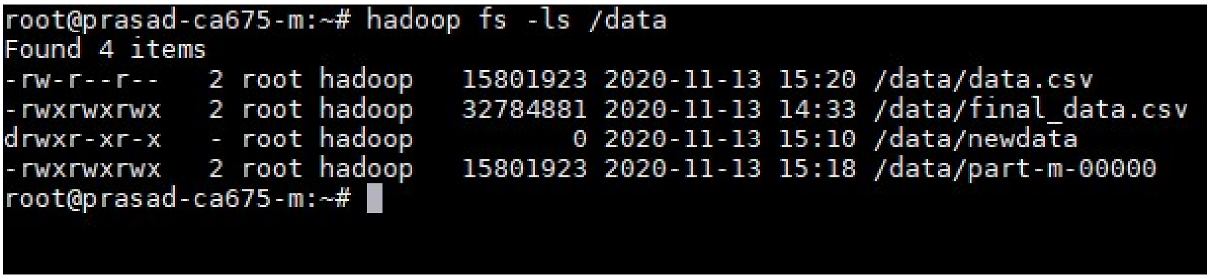
Data Preparation:

Once we have downloaded all 4 CSV files from above queries, we will use python for data preparation. For example, merging for 4 files into **final_data.csv** and rest of the cleaning is done in PIG.

Code can be viewed on - https://github.com/prasad1825/cloud-ass-1/tree/main/Code

Data Cleaning using PIG:

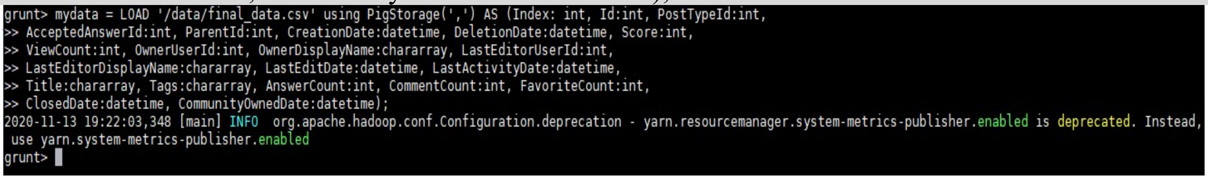
1. We copy **final_data.csv** in root directory into our dataproc cluster using SFTP.
2. Then we copy it inside of Hadoop dir from root directory of dataproc cluster using **hadoop fs -put final_data.csv /data.**



```
root@prasad-ca675-m:~# hadoop fs -ls /data
Found 4 items
-rw-r--r--   2 root hadoop   15801923 2020-11-13 15:20 /data/data.csv
-rwxrwxrwx   2 root hadoop   32784881 2020-11-13 14:33 /data/final_data.csv
drwxr-xr-x   - root hadoop          0 2020-11-13 15:10 /data/newdata
-rwxrwxrwx   2 root hadoop   15801923 2020-11-13 15:18 /data/part-m-00000
root@prasad-ca675-m:~# 
```

3. Using below command I loaded dataset in PIG, specifying each data type.

```
mydata = LOAD '/data/final_data.csv' using PigStorage(',') AS (Index: int, Id:int, PostTypeId:int, AcceptedAnswerId:int, ParentId:int, CreationDate:datetime, DeletionDate:datetime, Score:int, ViewCount:int, OwnerUserId:int, OwnerDisplayName:chararray, LastEditorUserId:int, LastEditorDisplayName:chararray, LastEditDate:datetime, LastActivityDate:datetime, Title:chararray, Tags:chararray, AnswerCount:int, CommentCount:int, FavoriteCount:int, ClosedDate:datetime, CommunityOwnedDate:datetime);
```



```
grunt> mydata = LOAD '/data/final_data.csv' using PigStorage(',') AS (Index: int, Id:int, PostTypeId:int,
>> AcceptedAnswerId:int, ParentId:int, CreationDate:datetime, DeletionDate:datetime, Score:int,
>> ViewCount:int, OwnerUserId:int, OwnerDisplayName:chararray, LastEditorUserId:int,
>> LastEditorDisplayName:chararray, LastEditDate:datetime, LastActivityDate:datetime,
>> Title:chararray, Tags:chararray, AnswerCount:int, CommentCount:int, FavoriteCount:int,
>> ClosedDate:datetime, CommunityOwnedDate:datetime);
2020-11-13 19:22:03,348 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead,
 use yarn.system-metrics-publisher.enabled
grunt> 
```

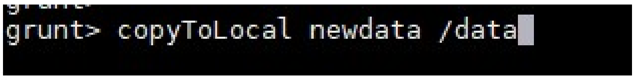
4. Created new table using required columns.

```
A = FOREACH mydata GENERATE Id, Score, ViewCount, OwnerUserId, OwnerDisplayName, Title, Tags;
```

5. Saved cleaned data into new folder

```
STORE A INTO '/data/newdata' using PigStorage(',');
```

6. Copy the newdata folder into /data directory of hadoop



```
grunt> copyToLocal newdata /data
```

7. Now we can see our cleaned dataset “part-m-00000” in the /data/newdata directory. Next, we copy our cleaned dataset into data.csv allow HIVE to access it.
8. hadoop fs -cp /data/part-m-00000 /data/data.csv

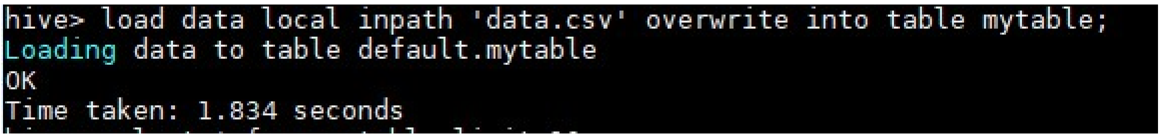
Querying using HIVE:

1. We need to create the empty table(mytable) to fill in, using our dataset. Then we can import our dataset ‘data.csv’ and overwrite the table.

```
create external table if not exists mytable(Id int, Score int, ViewCount int, OwnerUserId int, OwnerDisplayName string, Title string, Tags string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
```

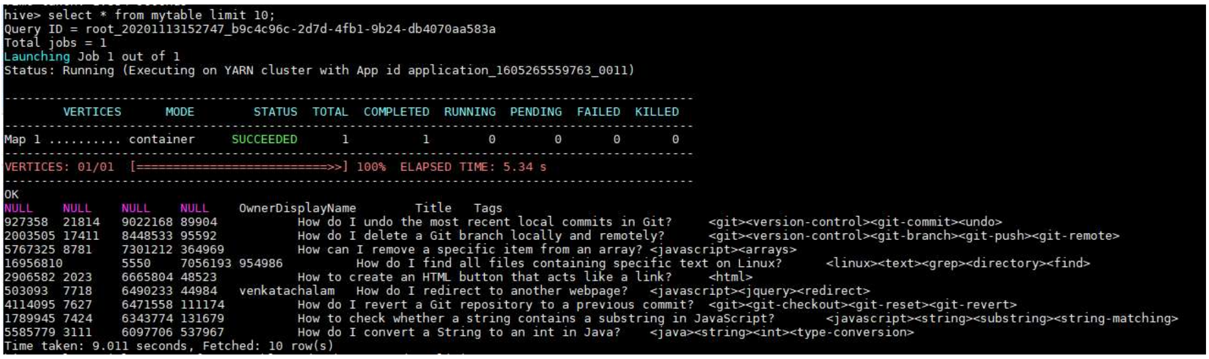
2. Then load data.cvs into our table(Mytable)using below command:

```
load data local inpath '/data/data.csv' overwrite into table mytable
```



```
hive> load data local inpath 'data.csv' overwrite into table mytable;
Loading data to table default.mytable
OK
Time taken: 1.834 seconds
```

```
select * from mytable limit 10;
```



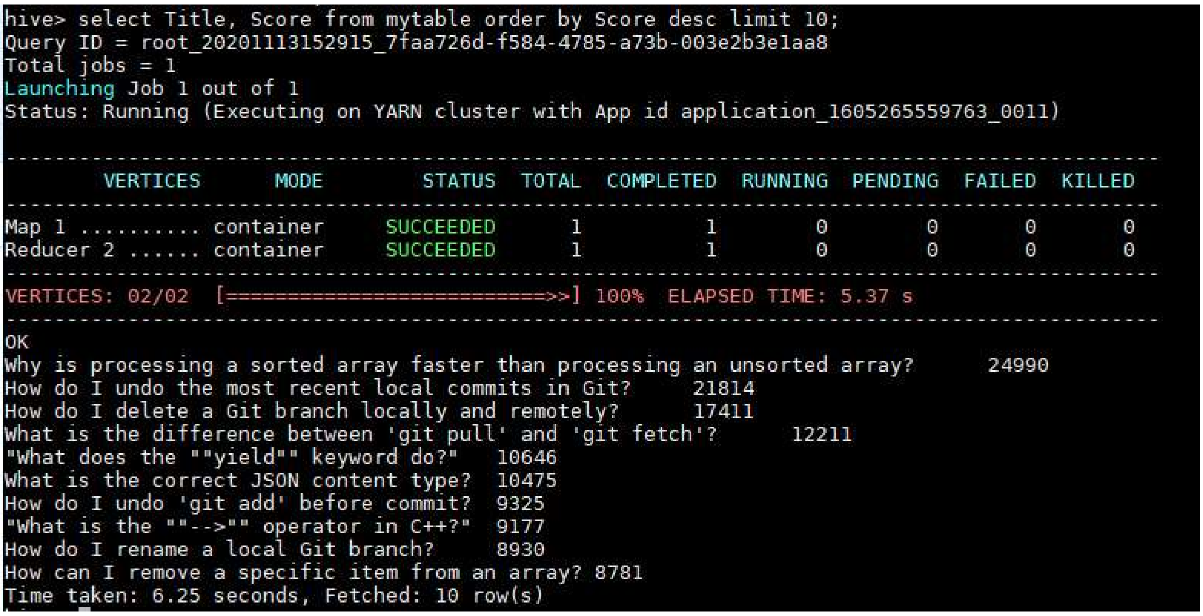
```
hive> select * from mytable limit 10;
Query ID = root_20201113152747_b9c4c96c-2d7d-4fb1-9b24-db4070aa583a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605265559763_0011)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 5.34 s
----------------------------------------------------------------------------------------------
OK
NULL    NULL    NULL    NULL    OwnerDisplayName        Title   Tags
927358  21814   9022168 89904           How do I undo the most recent local commits in Git?     <git><version-control><git-commit><undo>
2003505 17411   8448533 95592           How do I delete a Git branch locally and remotely?      <git><version-control><git-branch><git-push><git-remote>
5767325 8781    7301212 364969          How can I remove a specific item from an array? <javascript><arrays>
16956810        5550    7056193 954986          How do I find all files containing specific text on Linux?      <linux><text><grep><directory><find>
2906582 2023    6665804 48523           How to create an HTML button that acts like a link?     <html>
503093  7718    6490233 44984   venkatachalam   How do I redirect to another webpage?   <javascript><jquery><redirect>
4114095 7627    6471558 111174          How do I revert a Git repository to a previous commit?  <git><git-checkout><git-reset><git-revert>
1789945 7424    6343774 131679          How to check whether a string contains a substring in JavaScript?       <javascript><string><substring><string-matching>
5585779 3111    6097706 537967          How do I convert a String to an int in Java?    <java><string><int><type-conversion>
Time taken: 9.011 seconds, Fetched: 10 row(s)
```

Now we can proceed with assignment questions:

1. The top 10 posts by score

- We can find top 10 posts using **title** and **score** in dataset we downloaded from StackExchange in oder by the **Score**.

select Title, Score from **mytable** order by Score desc limit 10;

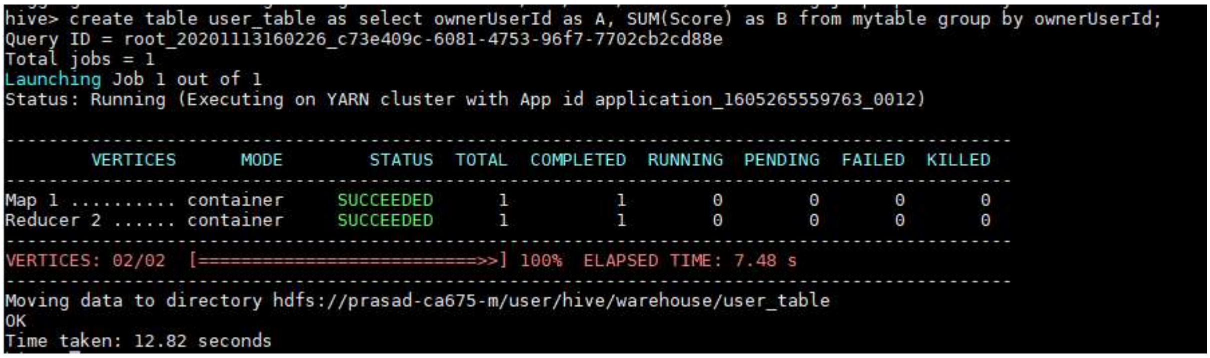


```
hive> select Title, Score from mytable order by Score desc limit 10;
Query ID = root_20201113152915_7faa726d-f584-4785-a73b-003e2b3e1aa8
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605265559763_0011)

----------------------------------------------------------------------------------------------
        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: 5.37 s
----------------------------------------------------------------------------------------------
OK
Why is processing a sorted array faster than processing an unsorted array?      24990
How do I undo the most recent local commits in Git?     21814
How do I delete a Git branch locally and remotely?      17411
What is the difference between 'git pull' and 'git fetch'?      12211
"What does the ""yield"" keyword do?"   10646
What is the correct JSON content type?  10475
How do I undo 'git add' before commit?  9325
"What is the ""-->"" operator in C++?"  9177
How do I rename a local Git branch?     8930
How can I remove a specific item from an array? 8781
Time taken: 6.25 seconds, Fetched: 10 row(s)
```

2. The top 10 users by score

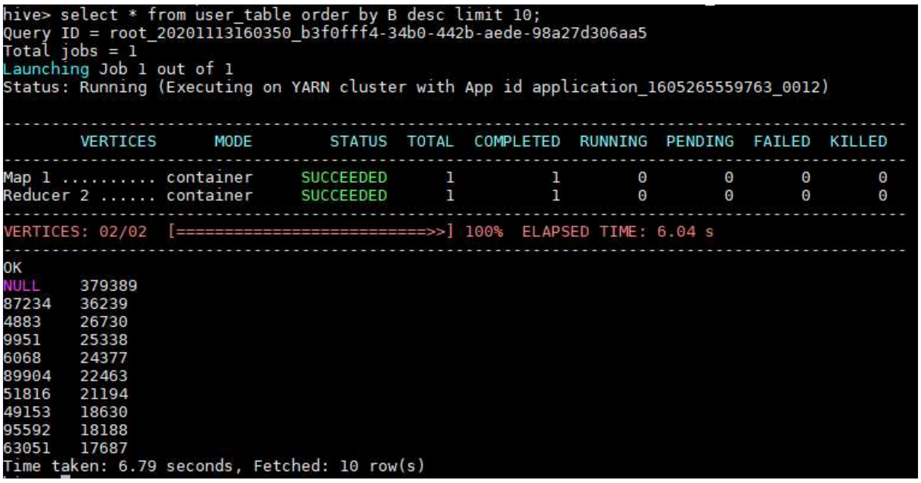
There are two ways to find out 10 users by score either using **UserID** or **UserName**. At first, we get top 10 users by score using **UserId** and in the second time we get it using **UserName**. Inorder to sort users by the total score we used aggregate functions such as SUM() & group by(). Therefore, it makes more sense to create a temporary tables which has two fields –A: UserID and B: the output from SUM(score) & group by(UserID).

1. create table user_table as select ownerUserId as A, SUM(Score) as B from mytable group by ownerUserId;



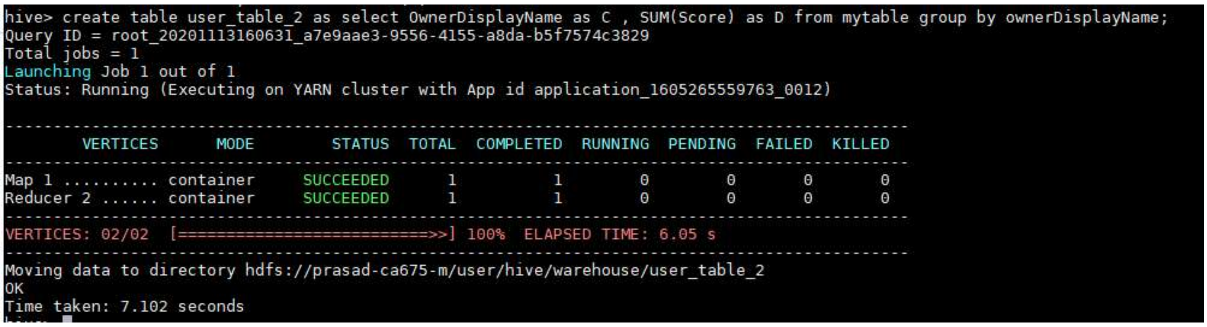
```
hive> create table user_table as select ownerUserId as A, SUM(Score) as B from mytable group by ownerUserId;
Query ID = root_20201113160226_c73e409c-6081-4753-96f7-7702cb2cd88e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605265559763_0012)

----------------------------------------------------------------------------------------------
        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: 7.48 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://prasad-ca675-m/user/hive/warehouse/user_table
OK
Time taken: 12.82 seconds
```

select * from user_table order by B desc limit 10;



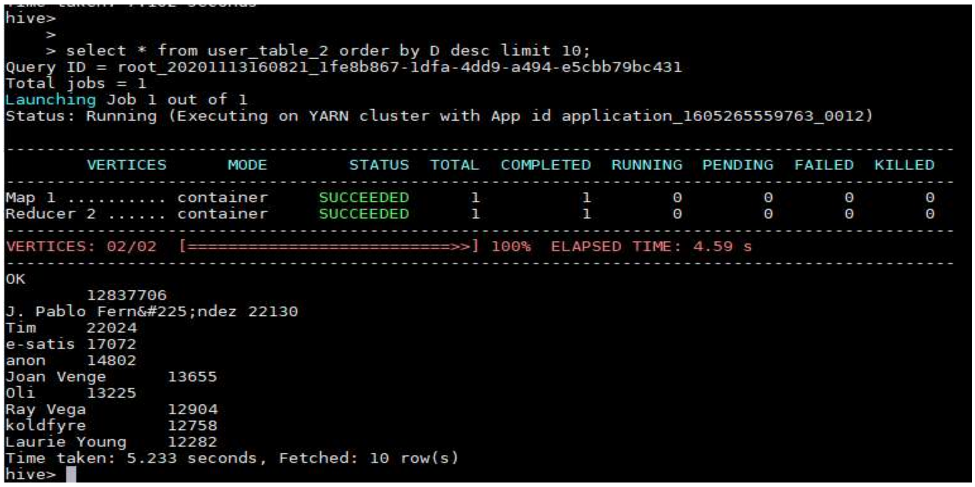
```
hive> select * from user_table order by B desc limit 10;
Query ID = root_20201113160350_b3f0fff4-34b0-442b-aede-98a27d306aa5
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605265559763_0012)

----------------------------------------------------------------------------------------------
        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: 6.04 s
----------------------------------------------------------------------------------------------
OK
NULL    379389
87234   36239
4883    26730
9951    25338
6068    24377
89904   22463
51816   21194
49153   18630
95592   18188
63051   17687
Time taken: 6.79 seconds, Fetched: 10 row(s)
```

2. create table user_table_2 as select OwnerDisplayName as C , SUM(Score) as D from mytable group by ownerDisplayName;



```
hive> create table user_table_2 as select OwnerDisplayName as C , SUM(Score) as D from mytable group by ownerDisplayName;
Query ID = root_20201113160631_a7e9aae3-9556-4155-a8da-b5f7574c3829
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605265559763_0012)

----------------------------------------------------------------------------------------------
        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: 6.05 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://prasad-ca675-m/user/hive/warehouse/user_table_2
OK
Time taken: 7.102 seconds
```

select * from user_table_2 order by D desc limit 10;

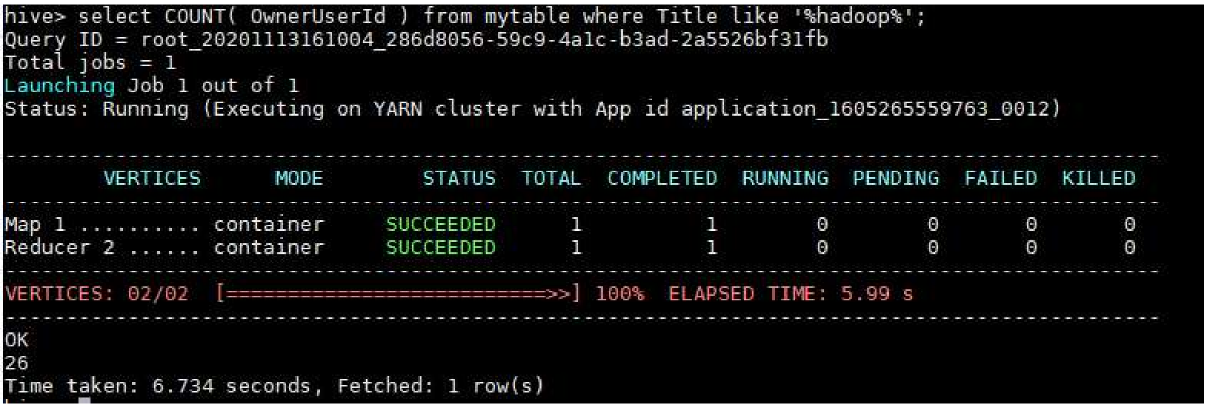


```
hive>
    >
    > select * from user_table_2 order by D desc limit 10;
Query ID = root_20201113160821_1fe8b867-1dfa-4dd9-a494-e5cbb79bc431
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605265559763_0012)

----------------------------------------------------------------------------------------------
        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: 4.59 s
----------------------------------------------------------------------------------------------
OK
        12837706
J. Pablo Fern&#225;ndez 22130
Tim     22024
e-satis 17072
anon    14802
Joan Venge      13655
Oli     13225
Ray Vega        12904
koldfyre        12758
Laurie Young    12282
Time taken: 5.233 seconds, Fetched: 10 row(s)
hive>
```

3. The number of distinct users, who used the word ‘hadoop’ in one of their posts.

In order to find how many users posted about HADOOP in Stack Exchange website. Thus another aggregate function COUNT( )is used with respect to word “hadoop”. Similarly, you can also find out by using “Hadoop”.

select COUNT(OwnerUserId) from mytable where Title like '%hadoop%';



```
hive> select COUNT( OwnerUserId ) from mytable where Title like '%hadoop%';
Query ID = root_20201113161004_286d8056-59c9-4a1c-b3ad-2a5526bf31fb
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605265559763_0012)

----------------------------------------------------------------------------------------------
        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: 5.99 s
----------------------------------------------------------------------------------------------
OK
26
Time taken: 6.734 seconds, Fetched: 1 row(s)
```

4. Calculate the per-user TF-IDF with HIVE:

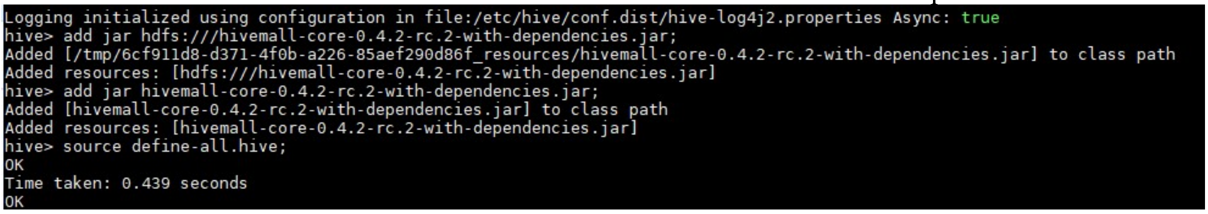
Find Top 10 terms used for each of the top 10 users by post score:

We need to install Apache Hivemall extension in order to use hivemall UDF's to find out TF-IDF.

1. Download the following two installation files and place into the dataproc directory and then copy it to Hadoop.

- define-all.hive
- hivemall-core-0.4.2-rc.2-with-dependencies.jar

2. load all Hivemall functions and define macros used in the TF-IDF computation.

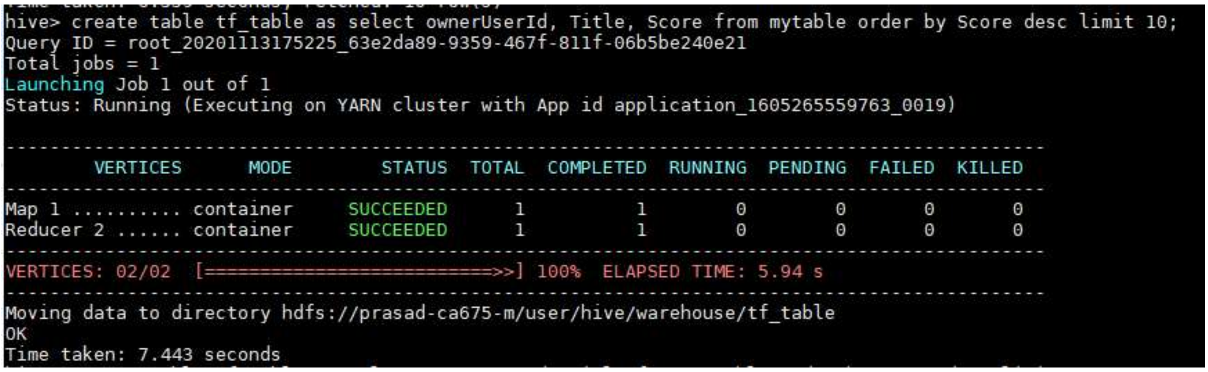


```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: true
hive> add jar hdfs:///hivemall-core-0.4.2-rc.2-with-dependencies.jar;
Added [/tmp/6cf911d8-d371-4f0b-a226-85aef290d86f_resources/hivemall-core-0.4.2-rc.2-with-dependencies.jar] to class path
Added resources: [hdfs:///hivemall-core-0.4.2-rc.2-with-dependencies.jar]
hive> add jar hivemall-core-0.4.2-rc.2-with-dependencies.jar;
Added [hivemall-core-0.4.2-rc.2-with-dependencies.jar] to class path
Added resources: [hivemall-core-0.4.2-rc.2-with-dependencies.jar]
hive> source define-all.hive;
OK
Time taken: 0.439 seconds
OK
```

- 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);

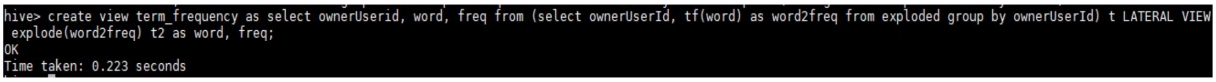
Create a table To calculate TF-IDF, preparing a relation consists of (docid,word) tuples and do TF-IDF calculation for each doc id/word pair**.**

create table tf_table as select ownerUserId, Title from mytable order by Score desc limit 10;



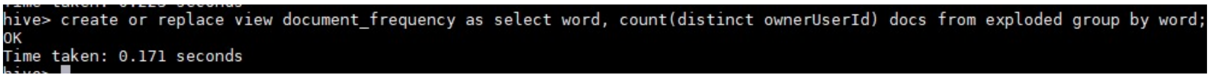
```
hive> create table tf_table as select ownerUserId, Title, Score from mytable order by Score desc limit 10;
Query ID = root_20201113175225_63e2da89-9359-467f-811f-06b5be240e21
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605265559763_0019)

----------------------------------------------------------------------------------------------
        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: 5.94 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://prasad-ca675-m/user/hive/warehouse/tf_table
OK
Time taken: 7.443 seconds
```

- create view exploded as select ownerUserId, word from tf_table LATERAL VIEW explode(split(Title, True)) t as word where not is_stopword(word);
- create view term_frequency as select ownerUserid, word, freq from (select ownerUserId, tf(word) as word2freq from exploded group by ownerUserId) t LATERAL VIEW explode(word2freq) t2 as word, freq;



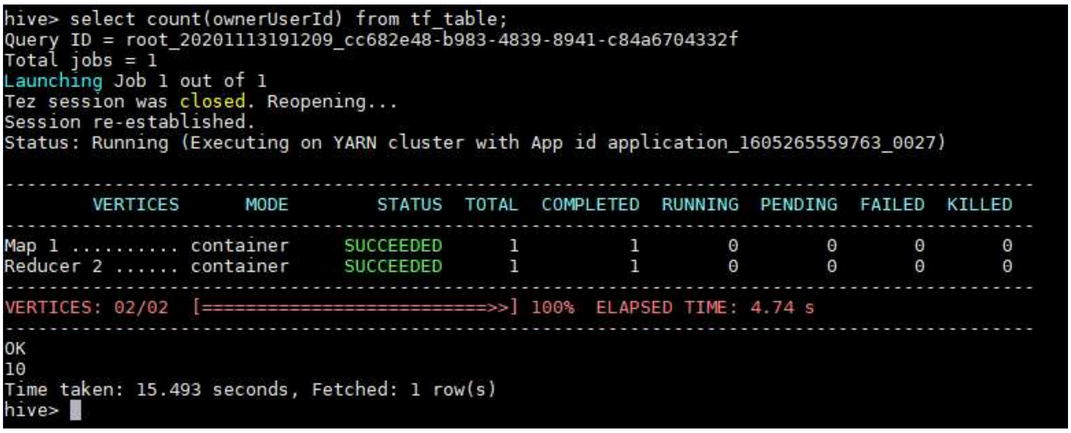
```
hive> create view term_frequency as select ownerUserid, word, freq from (select ownerUserId, tf(word) as word2freq from exploded group by ownerUserId) t LATERAL VIEW
 explode(word2freq) t2 as word, freq;
OK
Time taken: 0.223 seconds
```

- create or replace view document_frequency as select word, count(distinct ownerUserId) docs from exploded group by word;



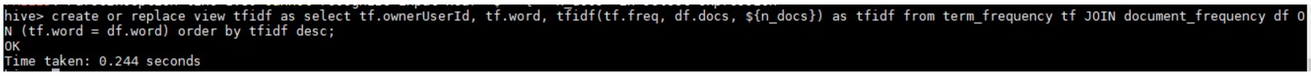
```
hive> create or replace view document_frequency as select word, count(distinct ownerUserId) docs from exploded group by word;
OK
Time taken: 0.171 seconds
```

- select count(ownerUserId) from tf_table;
- set hivevar:n_docs=10;



```
hive> select count(ownerUserId) from tf_table;
Query ID = root_20201113191209_cc682e48-b983-4839-8941-c84a6704332f
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1605265559763_0027)

----------------------------------------------------------------------------------------------
        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: 4.74 s
----------------------------------------------------------------------------------------------
OK
10
Time taken: 15.493 seconds, Fetched: 1 row(s)
hive>
```

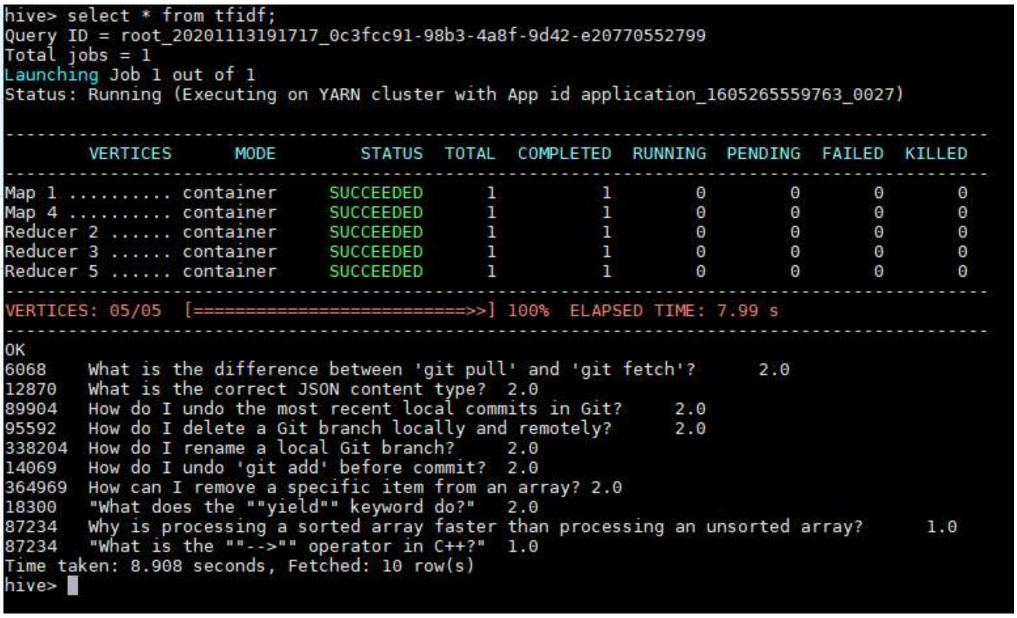
- create or replace view tfidf as select tf.ownerUserId, tf.word, tfidf(tf.freq, df.docs, ${n_docs}) as tfidf from term_frequency tf JOIN document_frequency df ON (tf.word = df.word) order by tfidf desc;



```
hive> create or replace view tfidf as select tf.ownerUserId, tf.word, tfidf(tf.freq, df.docs, ${n_docs}) as tfidf from term_frequency tf JOIN document_frequency df O
N (tf.word = df.word) order by tfidf desc;
OK
Time taken: 0.244 seconds
```

Now we can get the result(w.r.t **userID** & **terms** used)

- select * from tfidf;



```
hive> select * from tfidf;
Query ID = root_20201113191717_0c3fcc91-98b3-4a8f-9d42-e20770552799
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1605265559763_0027)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 4 .......... 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
Reducer 5 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 7.99 s
----------------------------------------------------------------------------------------------
OK
6068    What is the difference between 'git pull' and 'git fetch'?      2.0
12870   What is the correct JSON content type?  2.0
89904   How do I undo the most recent local commits in Git?     2.0
95592   How do I delete a Git branch locally and remotely?      2.0
338204  How do I rename a local Git branch?     2.0
14069   How do I undo 'git add' before commit?  2.0
364969  How can I remove a specific item from an array? 2.0
18300   "What does the ""yield"" keyword do?"   2.0
87234   Why is processing a sorted array faster than processing an unsorted array?      1.0
87234   "What is the ""-->"" operator in C++?"  1.0
Time taken: 8.908 seconds, Fetched: 10 row(s)
hive>
```

Following technologies has been used in while doing this Assignment:

1. **Stack Exchange:** Stack Exchange is a network of question-and-answer websites on topics in diverse fields, each site covering a specific topic, where questions, answers, and users are subject to a reputation award process. The reputation system allows the sites to be self-moderating."Stack Exchange Data Explorer (SEDE) https://data.stackexchange.com/stackoverflow/query/new.
2. **Dataproc:** Dataproc is a fast, easy-to-use, fully managed cloud service for running Apache Spark and Apache Hadoop clusters in a simpler, more cost-efficient way. https://cloud.google.com/dataproc.
3. **Apache HiveMall:** Apache Hivemall is a scalable machine learning library that runs on Apache Hive, Apache Spark, and Apache Pig. https://hivemall.apache.org/
4. **Github:** GitHub is a code hosting platform for version control and collaboration between developers. Refer to my repository - https://github.com/prasad1825/cloud-ass-1.
5. **MobaXterm:** It provides all the important remote network tools (SSH, RDP, X11, SFTP, FTP, Telnet, Rlogin, ...) to Windows desktop