



Name :- Hashmeetsingh Obhan

Student ID :- 21262793

Email ID :- hashmeetsingh.obhan2@mail.dcu.ie

Link for the Git Repository:-

<https://github.com/Hashmeetsingh/CloudTechnologies>

Description of the dataset:-

The data in the dataset comes from a social media app like Facebook or Instagram. It has several key columns, such as Id, Body, Score, UserName, and Tags, that provide detailed information on the user.

Task 1:- Get data from Stack Exchange (Data Acquisition/Collection)

- To acquire the data I used the Data Explorer feature of the StackExchange system using following link to run the below 4 queries.

LINK:- <https://data.stackexchange.com/stackoverflow/query/new>

Technology used:- Used SQL like queries to fetch the data from Data Explorer feature of the StackExchange system. In the below 4 queries I have fetched mostly all the columns of POST table on the basis of ViewCount (ORDER BY ViewCount) in the descending order. I have used the in-built function of SQL like ROW_NUMBER() as I need to fetch 50000 records at a time which made it easier to implement the task and filtered the query in the WHERE clause by RowNumber itself.

Queries:-

SELECT * FROM (SELECT ROW_NUMBER() OVER (ORDER BY VIEWCOUNT DESC) AS RowNumber, Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, DeletionDate, Score, ViewCount, OwnerUserId, OwnerDisplayName, LastEditorUserId, LastEditorDisplayName, LastEditDate, LastActivityDate, Title, Tags, AnswerCount, CommentCount, FavoriteCount, ClosedDate, CommunityOwnedDate, ContentLicense FROM POSTS WHERE VIEWCOUNT IS NOT NULL) AS POSTSTABLE WHERE RowNumber BETWEEN 1 AND 50001

The screenshot shows the Stack Exchange Data Explorer interface. The SQL query is as follows:

```
SELECT * FROM (SELECT ROW_NUMBER() OVER (ORDER BY VIEWCOUNT DESC) AS RowNumber,
Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, DeletionDate, Score, ViewCount, OwnerUserId,
OwnerDisplayName, LastEditorUserId, LastEditorDisplayName, LastEditDate, LastActivityDate, Title, Tags,
AnswerCount, CommentCount, FavoriteCount, ClosedDate, CommunityOwnedDate, ContentLicense FROM
POSTS WHERE VIEWCOUNT IS NOT NULL) AS POSTSTABLE WHERE RowNumber BETWEEN 1 AND 50001
```

The results table shows the first row of data:

RowNumber	Id	PostTypeId	AcceptedAnswerId	ParentId	CreationDate	DeletionDate	Score	ViewCount	OwnerUserId	OwnerDisplay
1	927358	1	927386		2009-05-29 18:09:14		23348	10062790	89904	

Figure:-1. 50k records of Posts table

```
SELECT * FROM (SELECT ROW_NUMBER() OVER (ORDER BY VIEWCOUNT DESC) AS RowNumber,
Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, DeletionDate, Score, ViewCount, OwnerUserId,
OwnerDisplayName, LastEditorUserId, LastEditorDisplayName, LastEditDate, LastActivityDate, Title, Tags,
AnswerCount, CommentCount, FavoriteCount, ClosedDate, CommunityOwnedDate, ContentLicense FROM
POSTS WHERE VIEWCOUNT IS NOT NULL) AS POSTSTABLE WHERE RowNumber BETWEEN 50001
AND 100000
```

The screenshot shows a SQL query editor with the following query:

```
10 ParentId,
11 CreationDate,
12 DeletionDate,
13 Score,
14 ViewCount,
15 OwnerUserId,
16 OwnerDisplayName,
17 LastEditorUserId,
18 LastEditorDisplayName,
19 LastEditDate,
20 LastActivityDate,
21 Title,
22 Tags,
23 AnswerCount,
24 CommentCount,
25 FavoriteCount,
26 ClosedDate,
27 CommunityOwnedDate,
28 ContentLicense
29 FROM POSTS WHERE VIEWCOUNT IS NOT NULL
30 ) AS POSTSTABLE
31 WHERE RowNumber BETWEEN 50001 AND 100000
```

The query is executed, and the results are displayed in a table. The table has the following columns: RowNumber, Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, DeletionDate, Score, ViewCount, OwnerUserId, OwnerDisplayName. The first row of results is:

RowNumber	Id	PostTypeId	AcceptedAnswerId	ParentId	CreationDate	DeletionDate	Score	ViewCount	OwnerUserId	OwnerDisplayName
50001	287404	1	287586		2008-11-13 16:25:07		32	127752	25371	lengtche

Figure:-2. 50k records of Posts table

```
SELECT * FROM (SELECT ROW_NUMBER() OVER (ORDER BY VIEWCOUNT DESC) AS RowNumber,
Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, DeletionDate, Score, ViewCount, OwnerUserId,
OwnerDisplayName, LastEditorUserId, LastEditorDisplayName, LastEditDate, LastActivityDate, Title, Tags,
AnswerCount, CommentCount, FavoriteCount, ClosedDate, CommunityOwnedDate, ContentLicense FROM
POSTS WHERE VIEWCOUNT IS NOT NULL) AS POSTSTABLE WHERE RowNumber BETWEEN 100001
AND 150000
```

The screenshot shows a SQL query editor with the following query:

```
24 CommentCount,
25 FavoriteCount,
26 ClosedDate,
27 CommunityOwnedDate,
28 ContentLicense
29 FROM POSTS WHERE VIEWCOUNT IS NOT NULL
30 ) AS POSTSTABLE
31 WHERE RowNumber BETWEEN 100001 AND 150000
```

The query is executed, and the results are displayed in a table. The table has the following columns: RowNumber, Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, DeletionDate, Score, ViewCount, OwnerUserId, OwnerDisplayName. The first row of results is:

RowNumber	Id	PostTypeId	AcceptedAnswerId	ParentId	CreationDate	DeletionDate	Score	ViewCount	OwnerUserId	OwnerDisplayName
100001	8799660	1	8801409		2012-01-10 06:57:19		31	74784	1129903	

Figure:-3. 50k records of Posts table

```
SELECT * FROM (SELECT ROW_NUMBER() OVER (ORDER BY VIEWCOUNT DESC) AS RowNumber,
Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, DeletionDate, Score, ViewCount, OwnerUserId,
OwnerDisplayName, LastEditorUserId, LastEditorDisplayName, LastEditDate, LastActivityDate, Title, Tags,
AnswerCount, CommentCount, FavoriteCount, ClosedDate, CommunityOwnedDate, ContentLicense FROM
POSTS WHERE VIEWCOUNT IS NOT NULL) AS POSTSTABLE WHERE RowNumber BETWEEN 150001
AND 200000
```

permalink [hide sidebar >>](#)

[Run Query](#) [Cancel](#) Options: ☐ Text-only results ☐ Include execution plan

Switch to meta site |

[Results](#) [Messages](#) [Download CSV](#)

RowNumber	Id	PostTypeId	AcceptedAnswerId	ParentId	CreationDate	DeletionDate	Score	ViewCount	OwnerUserId	OwnerDisplayName
150001	12130653	1	12130885		2012-08-26 13:49:18		36	53346	185041	

Figure:-4. 50k records of Posts table

- Following the execution of the above four queries, I downloaded the CSV files containing 50000 records each, and then combined them into a single file with 200000 records. After that I cleaned “PostBodyData” and “PostTitle” columns of that file to get the optimized result.

Technology used:- I used Python to clean the data as it is quite easy to clean the dataset using python and reduces complexity as well. (6) (7) (8)

```
In [1]: import pandas as pd

In [2]: df1 = pd.read_csv("gs://dataproc-staging-us-central1-359314101846-x5guqtzg/data_hash/QueryResults1.csv")
df2 = pd.read_csv("gs://dataproc-staging-us-central1-359314101846-x5guqtzg/data_hash/QueryResults2.csv")
df3 = pd.read_csv("gs://dataproc-staging-us-central1-359314101846-x5guqtzg/data_hash/QueryResults3.csv")
df4 = pd.read_csv("gs://dataproc-staging-us-central1-359314101846-x5guqtzg/data_hash/QueryResults4.csv")

In [10]: df1 = df1.append(df2)
df1 = df1.append(df3)
df1 = df1.append(df4)
len(df1)

Out[10]: 200000

In [13]: df1['PostBodyData'] = df1['PostBodyData'].str.replace(r'<[>]*>', "", regex=True) # Removing Html tags
df1['PostBodyData'] = df1['PostBodyData'].str.replace(r'([^\w])', " ", regex=True) # Removing Punctuations
df1['PostBodyData'] = df1['PostBodyData'].str.replace(r'\n', " ", regex=True) # Removing New Line
df1['PostBodyData'] = df1['PostBodyData'].str.replace(r'\d+', " ", regex=True) # Removing Digits
```

Figure:-5. Merging 4 dataset files into 1 single file

The screenshot shows a Jupyter Notebook with three code cells. The first two cells use pandas' `str.replace` method to clean 'PostBodyData' and 'PostTitle' columns by removing HTML tags, punctuation, new lines, and digits. The third cell displays the first two rows of the cleaned dataset.

```
In [13]: df1['PostBodyData'] = df1['PostBodyData'].str.replace(r'<[<>]*>', "", regex=True) # Removing Html tags
df1['PostBodyData'] = df1['PostBodyData'].str.replace(r'([^\w])', " ", regex=True) # Removing Punctuations
df1['PostBodyData'] = df1['PostBodyData'].str.replace(r'\n', " ", regex=True) # Removing New Line
df1['PostBodyData'] = df1['PostBodyData'].str.replace(r'\d+', " ", regex=True) # Removing Digits

In [14]: df1['PostTitle'] = df1['PostTitle'].str.replace(r'<[<>]*>', "", regex=True) # Removing Html tags
df1['PostTitle'] = df1['PostTitle'].str.replace(r'([^\w])', " ", regex=True) # Removing Punctuations
df1['PostTitle'] = df1['PostTitle'].str.replace(r'\n', " ", regex=True) # Removing New Line
df1['PostTitle'] = df1['PostTitle'].str.replace(r'\d+', " ", regex=True) # Removing Digits

In [15]: df1
Out[15]:
```

Id	PostScore	PostViewCount	PostBodyData	OwnerPostIdentifier	LastEditedPostDate	LastActivityPostDate	PostTitle	PostTags	PostAnsCount
NaN	23348	10062790	I accidentally committed the wrong files to Git...	89904	30-06-2021 5:07	05-10-2021 13:26	How do I undo the most recent local commits in...	<git><version-control><git-commit><undo>	98
NaN	18514	9285139	I want to delete a branch both locally and rem...	95592	06-10-2021 22:38	06-10-2021 22:38	How do I delete a Git branch locally and remot...	<git><version-control><git-branch><git-push><en	41

Figure:-6. Cleaning “PostBodyData” and “PostTitle” columns of the dataset

- I performed all the tasks on Google Cloud Platform (GCP) environment. Firstly, I created an instance on GCP having 1 namenode and 2 workers (2 datanodes).
- Then I uploaded the CSV file having 200000 records on GCP cloud storage as shown in below image.

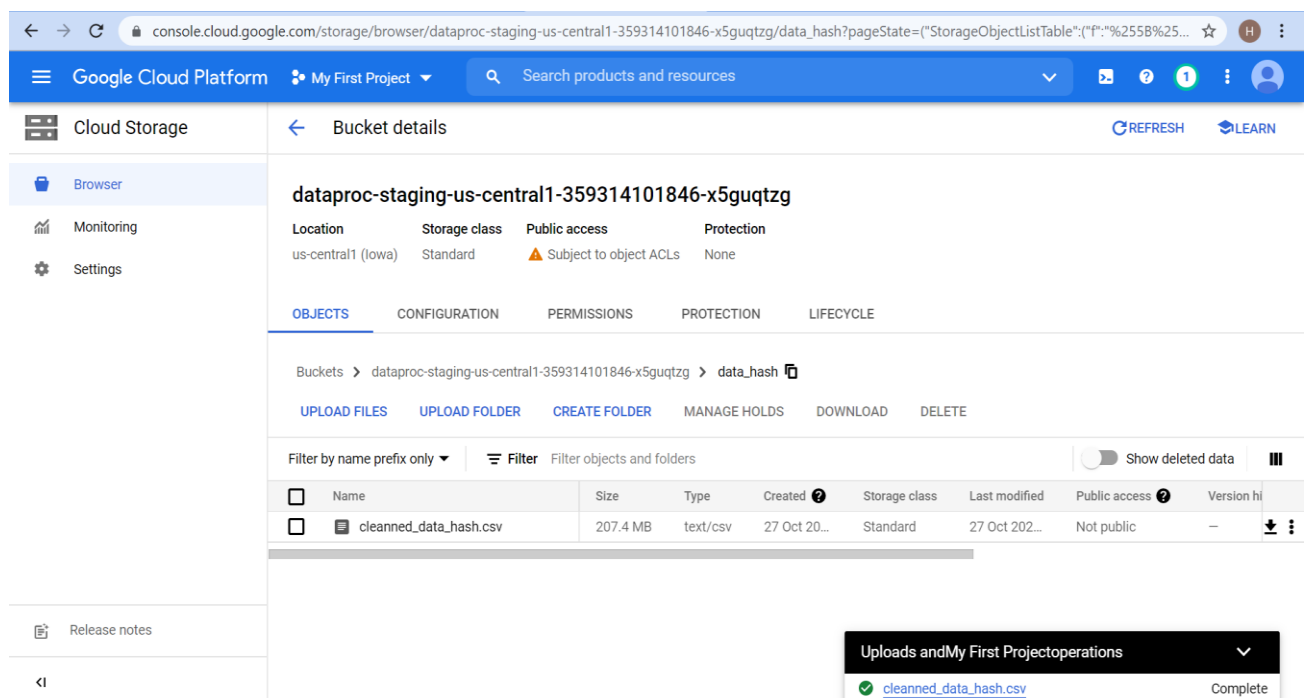


Figure:-7. Successfully uploaded the cleaned data file to GCP cloud

Task 2:- Load data into chosen cloud technology (Hive)

- Once uploading the file on GCP is completed I created a table on HIVE, which is pre-installed on GCP. Then I loaded the data in that table using the query. After that I created a View which has a similar structure as that of table where in I typecast few columns.

Technology used:- To import the data into the table, I utilized HIVE. This is owing to the fact that the HIVE is easier to implement than Pig and Mapreduce, requiring fewer development resources. Another reason I chose HIVE is that I am well familiar with SQL, which made it much easier to use. Similarly, I utilized the references described above to learn about linux commands and their functions, which I then implemented locally because JAVA, HADOOP, and HIVE were already installed on GCP, so I didn't have to do it again. I tried on AWS but got stuck in the middle and couldn't figure out how to fix the problem, then I went to GCP and everything worked fine. (1)(2)(3)(5)

```

hashmeetsingh_obhan2@hashsingh09-m: ~ - Google Chrome
ssh.cloud.google.com/projects/citric-replica-326917/zones/us-central1-a/instances/hashsingh09-m?authuser=0&hl=en_GB&projectNumber=359314101846&useAdminProxy=true&troubleshoot400...
hashmeetsingh_obhan2@hashsingh09-m:~$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: true
hive> CREATE TABLE IF NOT EXISTS HashStackExchangePosts
  > (Identifier int,
  > PostTypeIdentifier tinyint,
  > AcceptedAnswerIdentifier int,
  > ParentIdentifier int,
  > PostCreatedDate timestamp,
  > PostDeletedDate timestamp,
  > PostScore int,
  > PostViewCount int,
  > PostBodyData string,
  > OwnerPostIdentifier int,
  > OwnerPostName varchar(40),
  > LastEditedPostUserId int,
  > LastEditedPostName varchar(40),
  > LastEditedPostDate timestamp,
  > LastActivityPostDate timestamp,
  > PostTitle varchar (250),
  > PostTags varchar (250),
  > PostAnsCount int,
  > CommentedPostCount int,
  > FavoritePostCount int,
  > ClosedPostDate timestamp,
  > OwnedPostDate timestamp,
  > ContentPostLicense varchar (12))
  > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde';
OK
Time taken: 1.336 seconds
hive> LOAD DATA INPATH 'gs://dataproc-staging-us-central1-359314101846-x5guqtzg/data_hash/cleanned_data_hash.csv' INTO TABLE HashStackExchangePosts;
Loading data to table default.hashstackexchangeposts
OK
Time taken: 2.731 seconds

```

Figure:-8. Successfully loaded the data into table using HIVE

Task 3:- Run the Query data using Hive

Technology used:- Similarly like Task 2, I have used HIVE to execute the all the task 3 queries as they are similar to SQL like query language and easy to implement.

3.1. The top 10 posts by score

- To get the result I executed the below query on HIVE (GCP) where it returns the Id, Title and Score from the Posts table on the basis of Score in the descending order and fetched only 10 records so I have set the limit as 10.
- SELECT Identifier, PostTitle, PostScore from HashStackExchangeView ORDER BY PostScore DESC LIMIT 10;

```

hive> SELECT Identifier, PostTitle, PostScore from HashStackExchangeView ORDER BY PostScore DESC LIMIT 10;
Query ID = hashmeetsingh_obhan2_20211026164602_128ec265-f822-4e43-92f6-37a706d8de29
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_1635264620812_0004)

-----
VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    5         5         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 02/02  [=====] 100% ELAPSED TIME: 21.67 s
-----
OK
11227809 Why is processing a sorted array faster than processing an unsorted array? 25933
927358 How do I undo the most recent local commits in Git? 23348
2003505 How do I delete a Git branch locally and remotely? 18514
292357 What is the difference between 'git pull' and 'git fetch'? 12834
231767 What does the "yield" keyword do? 11551
477816 What is the correct JSON content type? 10921
348170 How do I undo 'git add' before commit? 10079
5767325 How can I remove a specific item from an array? 9931
6591213 How do I rename a local Git branch? 9792
1642028 What is the "---->" operator in C/C++? 9560
Time taken: 31.438 seconds, Fetched: 10 row(s)

```

Figure:-9. Successfully executed the top 10 posts by score

3.2. The top 10 users by post score

- To get the result I executed the below query on HIVE (GCP) where it returns the user details like OwnerUserId, OwnerDisplayName and Score from the Posts table on the basis of Posts Score where the limit is set to 10 to fetch only 10 records.
 - SELECT OwnerPostIdentifier, OwnerPostName, sum(PostScore) as PostScore from HashStackExchangeView GROUP BY OwnerPostIdentifier, OwnerPostName ORDER BY PostScore DESC LIMIT 10;

```
hive> SELECT OwnerPostIdentifier,OwnerPostName,sum(PostScore) as PostScore from HashStackExchangeView GROUP BY OwnerPostIdentifier, OwnerPostName ORDER BY PostScore
DESC LIMIT 10
> ;
Query ID = hashmeetsingh_obhan2_20211026170055_b66d3402-a3b5-4fef-903b-6f2efe0828df
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_1635264620812_0005)
```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1	container	SUCCEEDED	5	5	0	0	0	0	0
Reducer 2	container	SUCCEEDED	1	1	0	0	0	0	0
Reducer 3	container	SUCCEEDED	1	1	0	0	0	0	0

```
VERTICES: 03/03 [=====>>>] 100% ELAPSED TIME: 24.27 s
OK
87234 GManNickG 37672
4883 readonly 28817
9951 e-satis 26878
6068 pupeno 25944
89904 Hamza Yerlikaya 24024
51816 Joan Venge 23763
49153 Ali 20203
179736 TIMEX 19603
95592 Matthew Rankin 19479
63051 flybywire 19362
Time taken: 34.873 seconds, Fetched: 10 row(s)
```

Figure:-10. Successfully executed the top 10 users by post score

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

- Similarly like above 2 queries, I executed the below query on HIVE (GCP) to obtain the result. Below query returns the number of distinct users which contains the word “cloud” either in their Body or Title column.
 - SELECT COUNT(DISTINCT OwnerPostIdentifier) as TotalDistinctUsers FROM HashStackExchangeView WHERE PostTitle LIKE '% cloud %' OR PostBodyData LIKE '% cloud %';

```
hive> SELECT COUNT(DISTINCT OwnerPostIdentifier) as TotalDistinctUsers FROM HashStackExchangeView WHERE PostTitle LIKE '% cloud %' OR PostBodyData LIKE '% cloud %';
Query ID = hashmeetsingh_obhan2_20211026171011_7364cfc4-8390-4cfc-bf13-15f2016f8cd8
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_1635264620812_0006)
```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1	container	SUCCEEDED	5	5	0	0	0	0	0
Reducer 2	container	SUCCEEDED	1	1	0	0	0	0	0
Reducer 3	container	SUCCEEDED	1	1	0	0	0	0	0

```
VERTICES: 03/03 [=====>>>] 100% ELAPSED TIME: 23.64 s
OK
248
Time taken: 33.976 seconds, Fetched: 1 row(s)
```

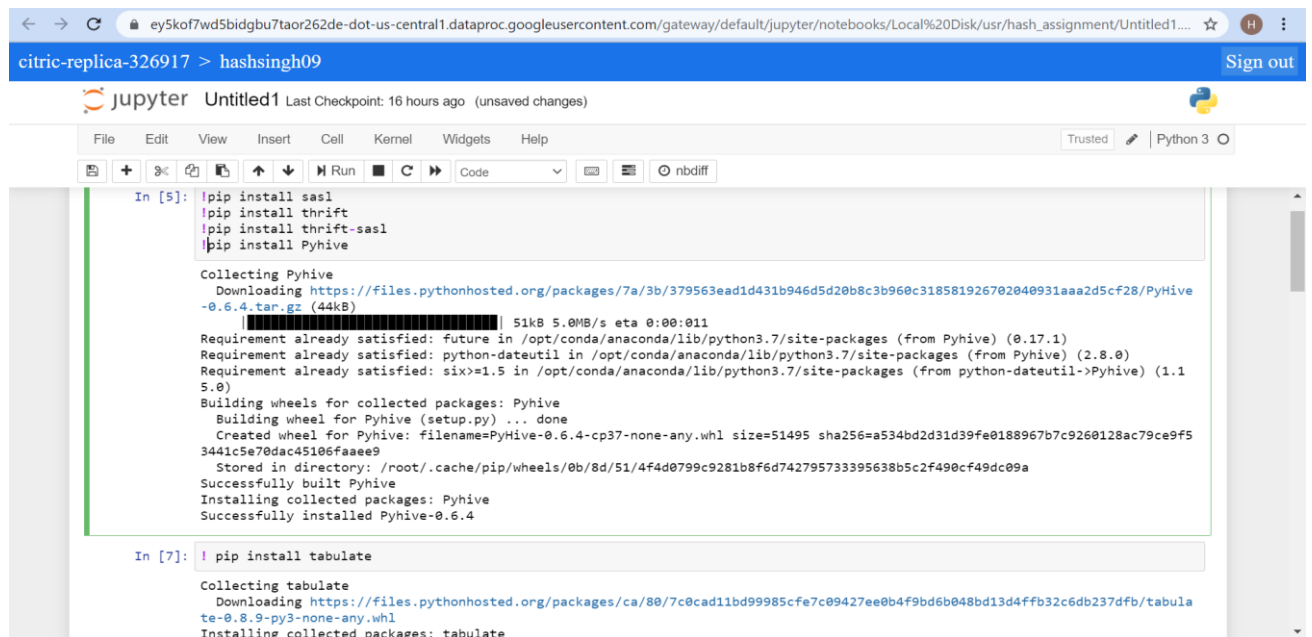
Figure:-11. Number of distinct users who used the word “cloud” in their posts Body/Title column

Task 4:- Calculate the per-user TF-IDF of the top 10 terms for each of the top 10 users.

Technology used:- This time I have used Python to find the top 10 terms for each of the 10 users as it was easy to develop and implement the code in python as compared to HIVE and Pig. Also, the code is much more efficient using Python and able to get the optimum result.

Explanation of the code Snippet:-

I started by installing the necessary libraries and packages. Then I created a connection to the HIVE database by passing the required authorization parameters to the connection string. Then, depending on their post score, I discovered the top ten user details. After that, I looked for OwnerPostIdentifier in the details I had retrieved earlier for 10 users. Last but not least, I used the sklearn library to find each user's TF IDF. Refer to Figures 9-10 and 11 for an example of how to find the TF IDF of the top 10 terms for each of the 10 users. (4)



```

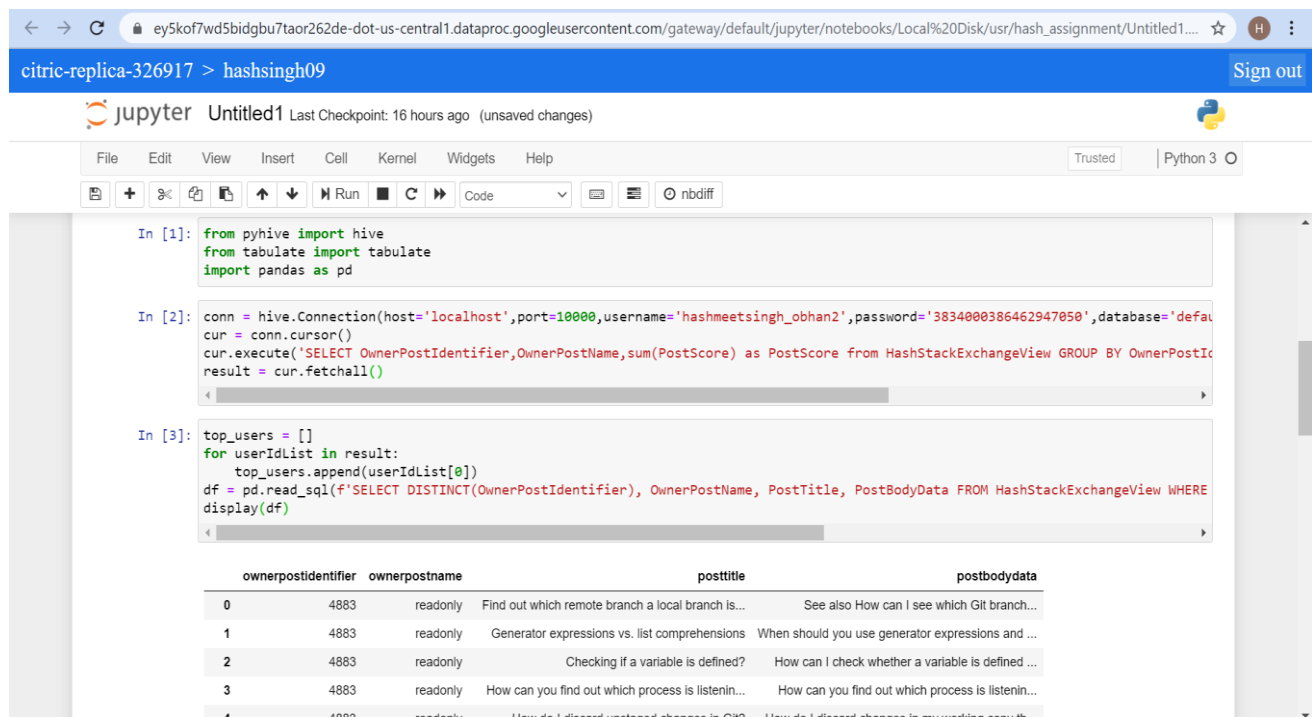
In [5]: !pip install sasl
!pip install thrift
!pip install thrift-sasl
!pip install Pyhive

Collecting Pyhive
  Downloading https://files.pythonhosted.org/packages/7a/3b/379563ead1d431b946d5d20b8c3b960c318581926702040931aaa2d5cf28/PyHive-0.6.4.tar.gz (44kB)
    [#####] 51kB 5.0MB/s eta 0:00:011
Requirement already satisfied: future in /opt/conda/anaconda/lib/python3.7/site-packages (from Pyhive) (0.17.1)
Requirement already satisfied: python-dateutil in /opt/conda/anaconda/lib/python3.7/site-packages (from Pyhive) (2.8.0)
Requirement already satisfied: six>=1.5 in /opt/conda/anaconda/lib/python3.7/site-packages (from python-dateutil->Pyhive) (1.15.0)
Building wheels for collected packages: Pyhive
  Building wheel for Pyhive (setup.py) ... done
  Created wheel for Pyhive: filename=PyHive-0.6.4-cp37-none-any.whl size=51495 sha256=a534bd2d31d39fe0188967b7c9260128ac79ce9f53441c5e70dac45106faae9
  Stored in directory: /root/.cache/pip/wheels/0b/8d/51/4f4d0799c9281b8fd74279573395638b5c2f490cf49dc09a
Successfully built Pyhive
Installing collected packages: Pyhive
Successfully installed Pyhive-0.6.4

In [7]: ! pip install tabulate

Collecting tabulate
  Downloading https://files.pythonhosted.org/packages/ca/80/7c0cad11bd99985cfe7c09427ee0b4f9bd6b048bd13d4ffb32c6db237dfb/tabulate-0.8.9-py3-none-any.whl
Installing collected packages: tabulate
  
```

Figure:-12. Installation of necessary library and packages needed to execute the code



```

In [1]: from pyhive import hive
from tabulate import tabulate
import pandas as pd

In [2]: conn = hive.Connection(host='localhost',port=10000,username='hashmeetsingh_obhan2',password='3834000386462947050',database='default')
cur = conn.cursor()
cur.execute('SELECT OwnerPostIdentifier,OwnerPostName,sum(PostScore) as PostScore from HashStackExchangeView GROUP BY OwnerPostIdentifier')
result = cur.fetchall()

In [3]: top_users = []
for userIdList in result:
    top_users.append(userIdList[0])
df = pd.read_sql(f'SELECT DISTINCT(OwnerPostIdentifier), OwnerPostName, PostTitle, PostBodyData FROM HashStackExchangeView WHERE OwnerPostIdentifier IN ({",".join(str(x) for x in top_users)})')
display(df)
  
```

	ownerpostidentifier	ownerpostname	posttitle	postbodydata
0	4883	readonly	Find out which remote branch a local branch is...	See also How can I see which Git branch...
1	4883	readonly	Generator expressions vs. list comprehensions	When should you use generator expressions and ...
2	4883	readonly	Checking if a variable is defined?	How can I check whether a variable is defined ...
3	4883	readonly	How can you find out which process is listenin...	How can you find out which process is listenin...
4	4883	readonly	How do I discard unstaged changes in Git?	How do I discard changes in my working copy th...

Figure:-13. Established the connection with HIVE Database and found the top 10 users

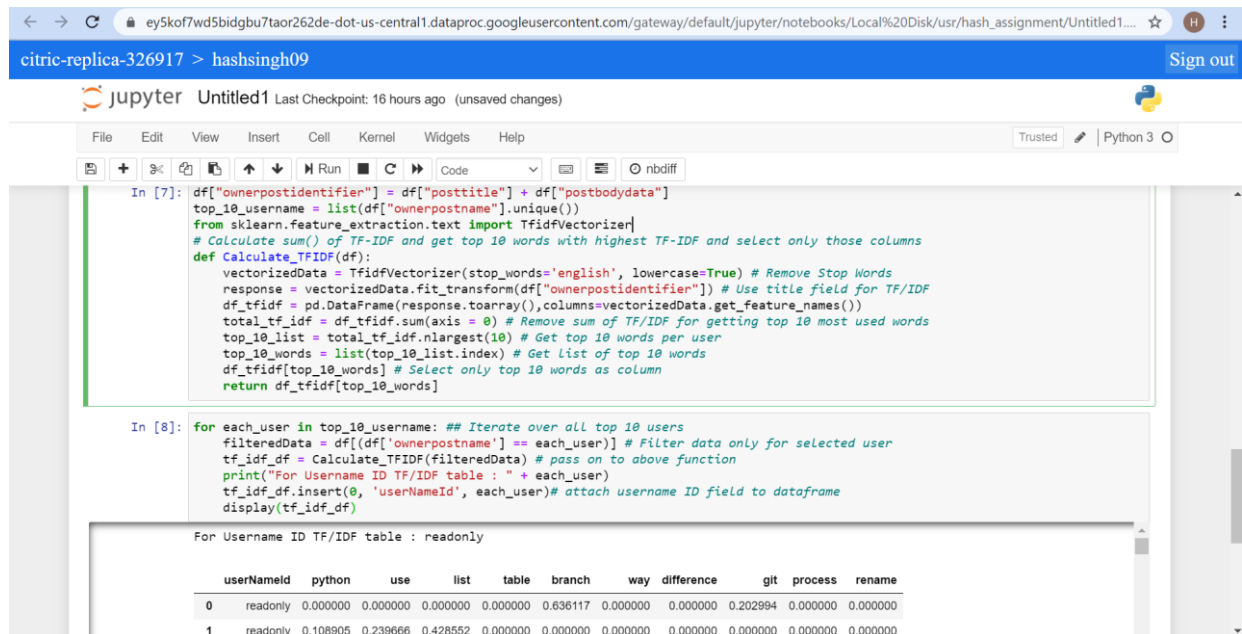


Figure:-14. Successfully executed the top 10 terms for each of the 10 users

APPENDIX

References:-

1. <https://phoenixnap.com/kb/install-hive-on-ubuntu>
2. <https://phoenixnap.com/kb/install-hadoop-ubuntu>
3. <https://www.guru99.com/file-permissions.html>
4. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
5. <https://www.projectpro.io/article/mapreduce-vs-pig-vs-hive/163>
6. <https://stackoverflow.com/questions/45999415/removing-html-tags-in-pandas>
7. <https://www.pythondaddy.com/python/how-to-remove-punctuation-from-a-dataframe-in-pandas-and-python/>
8. <https://stackoverflow.com/questions/41719259/how-to-remove-numbers-from-string-terms-in-a-pandas-dataframe>