|  |  |
| --- | --- |
| **Title** | Assignment 01 - Data Analysis |
| **Name** | Siddarth Shantinath Patil |
| **Student ID Number** | 20210993 |
| **E-mail ID** | [siddarth.patil2@mail.dcu.ie](mailto:siddarth.patil2@mail.dcu.ie) |
| **Programme of Study** | MSc in Computing with Data Analytics Major |
| **Module Code** | CA675 |
| **Date of Submission** | 22-Nov-2020 |
| **GitHub Link** | [siddarth-patil/Coud-Tech-Assginment-01-Data-Analysis: DCU Masters | CA675 | Cloud Technologies (github.com)](https://github.com/siddarth-patil/Coud-Tech-Assginment-01-Data-Analysis) |

**INTRODUCTION:**

The objective of this assignment it to perform data analysis on data which can be considered big data (200,000 records) and try to find the answers to the tasks given. Using the data obtained from StackExchange, I loaded, transformed and stored the data using PIG, further to find solutions to tasks I made use of HIVE and finally the last task was to find the TF-IDF of the data obtained in one of the tasks from HIVE and to perform this I made use of HIVEMALL.

The dataset consists of twenty-three columns in total which was further reduced to eight columns after carefully comparing it with our requirements. This was done using PIG queries. On the stored data, HIVE queries were used to answer three questions: finding top 10 posts by scores, finding top 10 users by post scores and finding number of distinct users who used the word “Hadoop” in their posts. Finally, HIVEMALL was used to find TF-IDF on the data which contained all the posts made by top 10 users who were found in the second task of HIVE queries. All the codes with output screenshots and explanations can be found at the GitHub Repository

**DATA ACQUISITION:**

The objective is to collect the top 200,000 records by score from StackExchange. StackExchange has a limit of 50,000 records per query. And therefore, to collect the data required for the assignment I made use of 5 queries. The code of the quires are as follows:

**For the top 44918 posts:**

select top 50000 \* from posts where posts.ViewCount > 121000 ORDER BY posts.ViewCount

**For the next 47817 posts:**

select top 50000 \* from posts where posts.ViewCount <= 121000

AND posts.ViewCount >70000 ORDER BY posts.ViewCount

**For the next 47066 posts:**

select top 50000 \* from posts where posts.ViewCount <= 70000

AND posts.ViewCount > 50000 ORDER BY posts.ViewCount

**For the next 41595 posts:**

select top 50000 \* from posts where posts.ViewCount <= 50000

AND posts.ViewCount > 40000 ORDER BY posts.ViewCount

**For the next 18604 posts:**

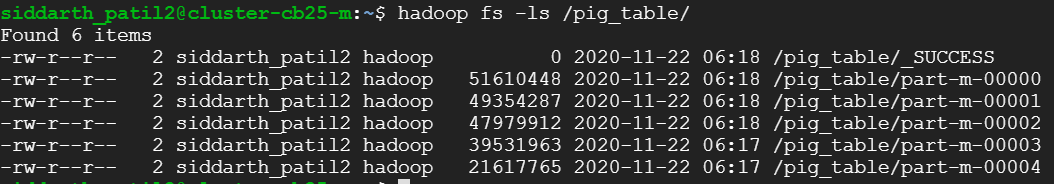
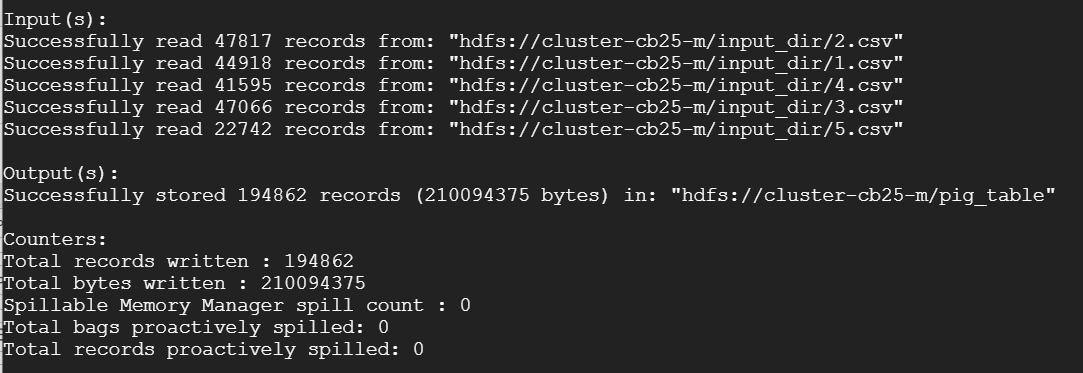
select top 18604 \* from posts where posts.ViewCount <= 40000

AND posts.ViewCount > 36000 ORDER BY posts.ViewCount

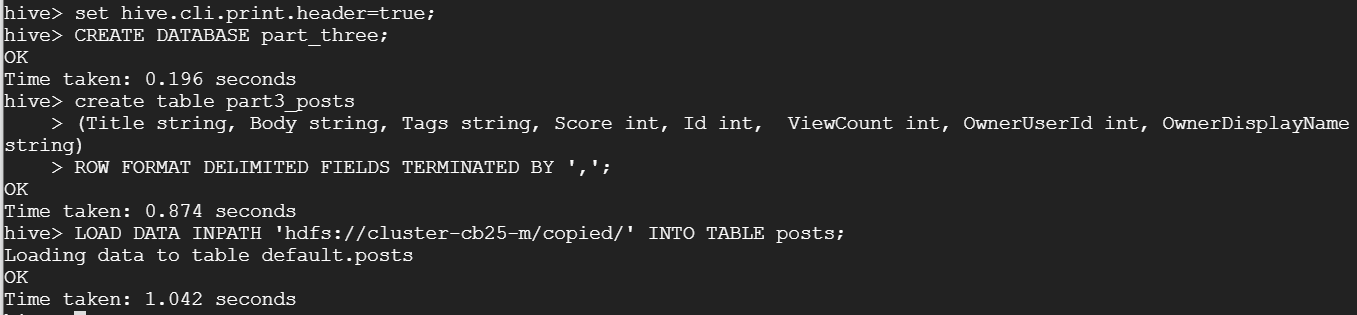
I arrived to the first number 121000 by hit and trail. The logic was to fetch a little less than 50,000 records to make sure that no data was missed. And I continued to fetch the data in similar way till I reached 200,000 records.

**DATA TRANSFORMATION USING PIG:**

Initially, the data acquired was loaded to local machine in GCP using the GUI and later was copied to HSFS. As there are 5 different csv files, the data was loaded in five different relations and then were joined using a UNION function. Later this joined data was transformed using the PIG queries. To transform, first the records where the OwnerUserID or the Id was null were removed and later out of 23 columns only 8 were kept and rest were omitted. Finally, this transformed data was stored in HDFS. The bellow screenshots provide the code and the output. The below codes and their explanation with references and screenshots can be found in the GitHub repository

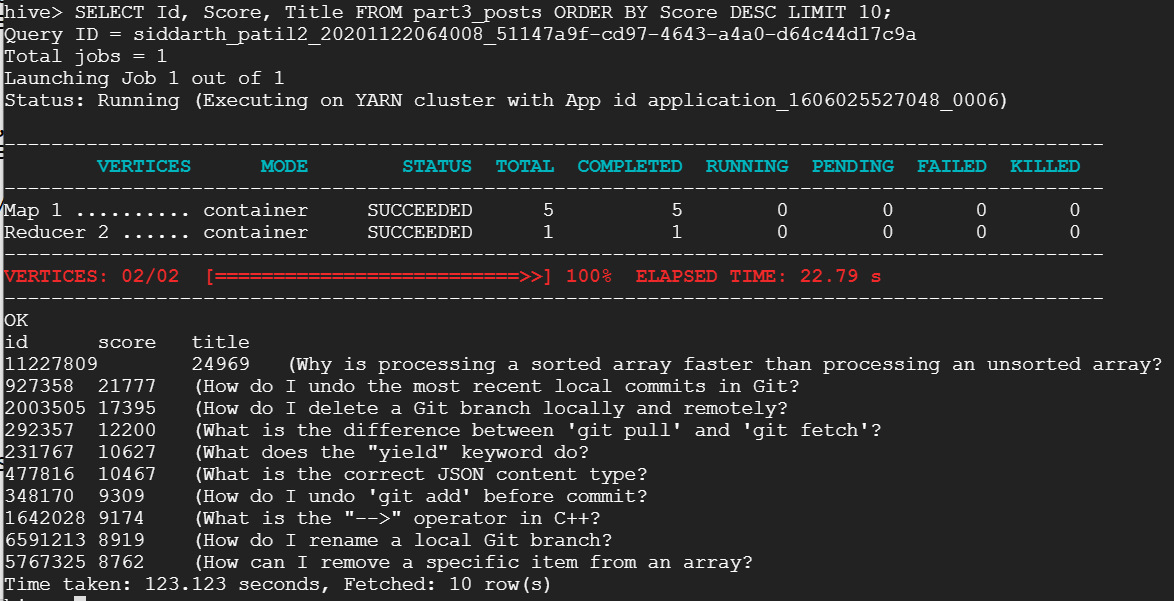


**HIVE LAODING AND QUERY:**

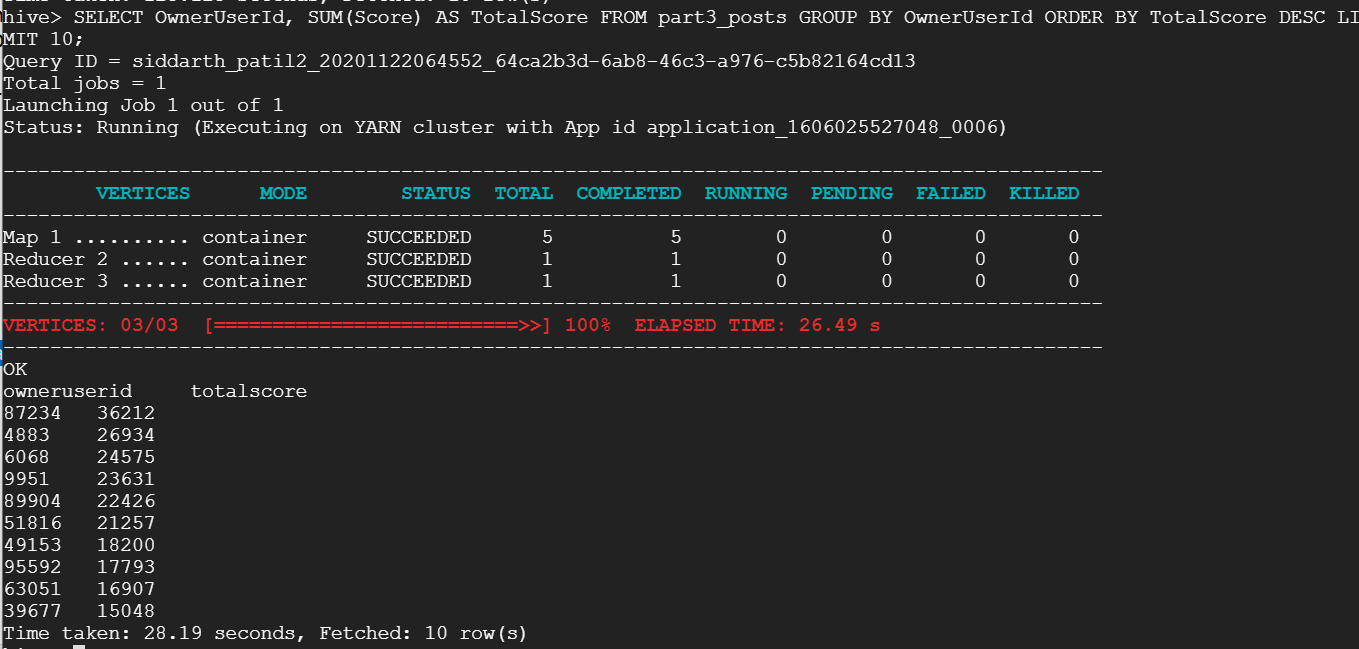
The data which was transformed using PIG in the above step was used and loaded here. Firstly, a Database was created and further a new table inside it. Finally, the data was loaded in the created table. The code with output for these steps can be seen below: 

Further on this table three quires were run for each of the three tasks:

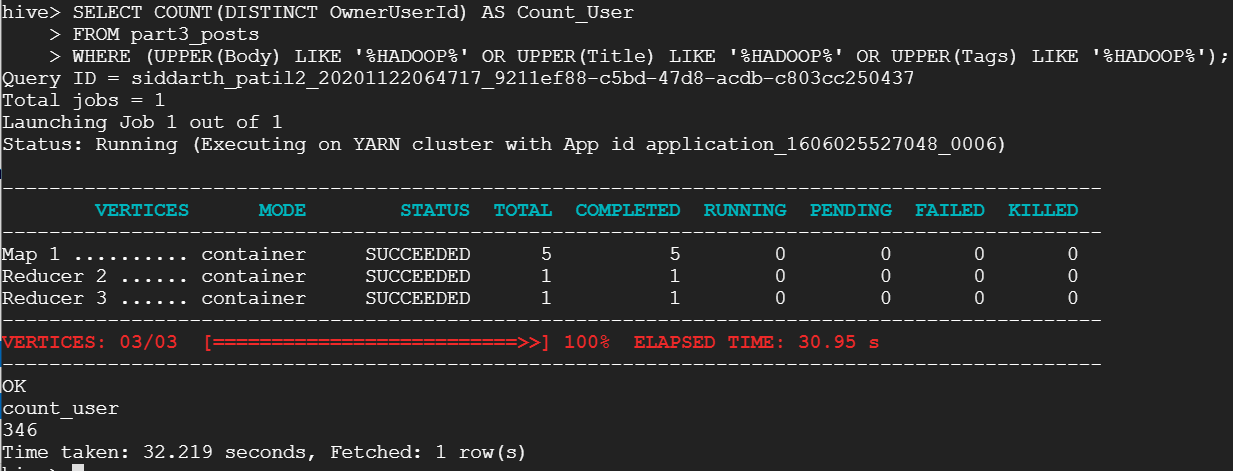
1. Finding top 10 posts by scores.



1. Finding top 10 users by post scores.

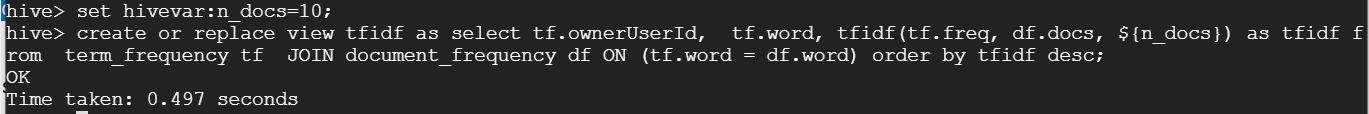
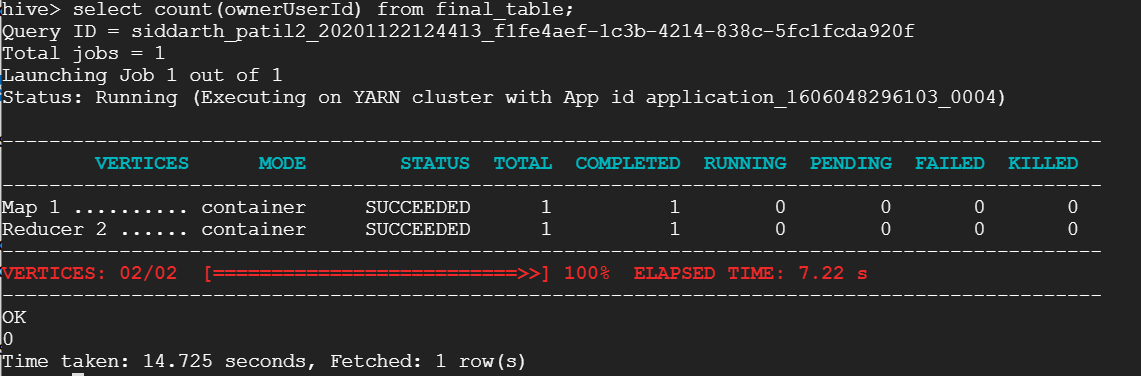
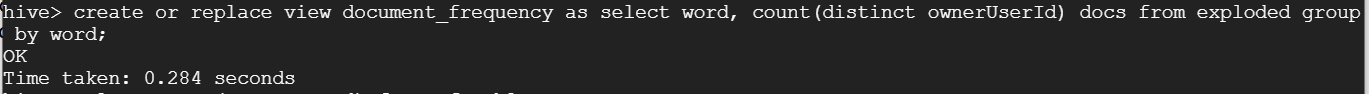
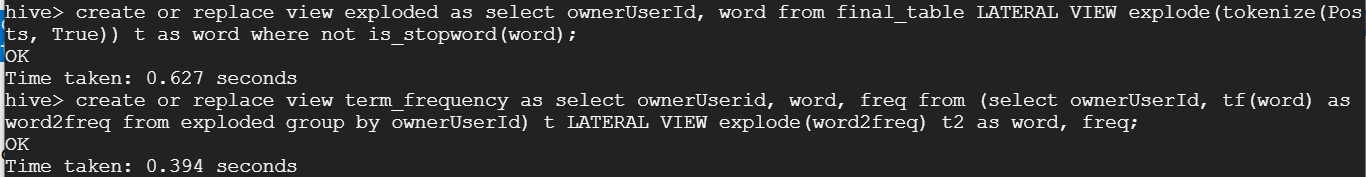
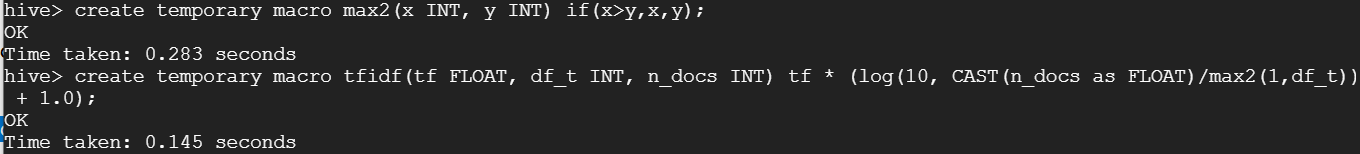
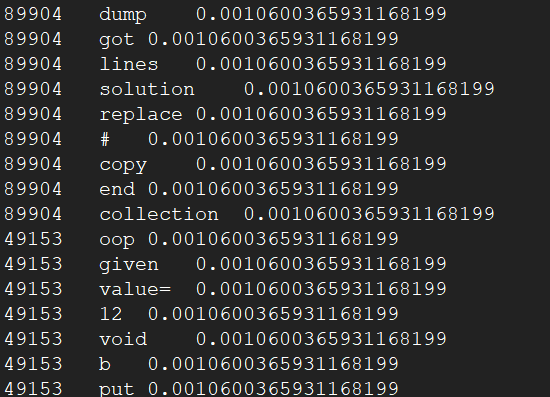


1. Finding number of distinct users who used the word “Hadoop” in their posts



**CALCULATING TF-IDF USING HIVEMALL:**

The data for TF-IDF consists of all the posts made by the top 10 users as calculated in the above steps. For this, the **OwnerUserId** of those 10 users were made use of with JOIN function on the table **part3\_posts** to get and store only the posts made by top 10 users. As the created table consists of all the eight fields it was cleaned and the final table consisted of only 2 columns: one was the OenerUserID and the other was the concatenation of **Body, Title** and **Tags** columns. Finally, dependencies of HIVEMALL were installed in order to use its functionalities. The screenshot of the code and output can be seen below.

****

**REFERENCES:**

1. [TF-IDF Term Weighting · Hivemall User Manual (apache.org)](https://hivemall.incubator.apache.org/userguide/ft_engineering/tfidf.html)
2. [Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/)
3. [Getting Started · Hivemall User Manual (apache.org)](https://hivemall.incubator.apache.org/userguide/getting_started/)
4. [Pig Latin Basics (apache.org)](https://pig.apache.org/docs/r0.17.0/basic.html)