Task 2: Load data into chosen cloud technology

1. Load and merge the data in HDFS:

A. Firstly, upload the 5 downloaded CSV files in the local home directory and check whether the same are present in it using the ls command.

hduser@Dell:/usr/local/hadoop$ ls

B. Then, move the files on Hadoop using the put command and check for their presence.

hduser@Dell:/usr/local/hadoop$ hadoop fs -put QueryResults.csv /

hduser@Dell:/usr/local/hadoop$ hadoop fs -put QueryResults_1.csv /

hduser@Dell:/usr/local/hadoop$ hadoop fs -put QueryResults_2.csv /

hduser@Dell:/usr/local/hadoop$ hadoop fs -put QueryResults_3.csv /

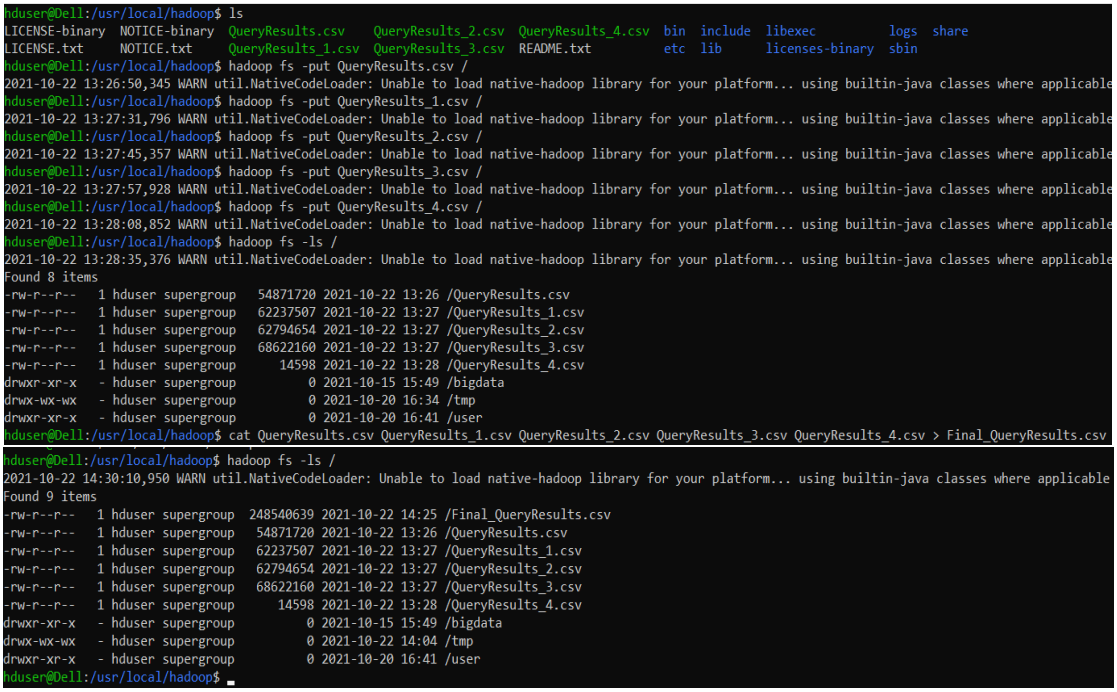
hduser@Dell:/usr/local/hadoop$ hadoop fs -put QueryResults_4.csv /

hduser@Dell:/usr/local/hadoop$ hadoop fs -ls /

C. Lastly, merge all the 5 CSV files into a single CSV file using the cat command and again check whether that single CSV file is present in it.

hduser@Dell:/usr/local/hadoop$ cat QueryResults.csv QueryResults_1.csv QueryResults_2.csv QueryResults_3.csv QueryResults_4.csv > Final_QueryResults.csv

hduser@Dell:/usr/local/hadoop$ hadoop fs -ls /



2. Clean the data in PIG:

A. Firstly, load the data from HDFS to PIG, specifying each data type.

```
grunt>stackdata=LOAD'hdfs://localhost:9870/usr/local/hadoopFinal_QueryResults.csv' USING PigStorage (',') AS (id:int, posttypeid:int, acceptedanswerid:int, parentid:int, creationdate:chararray, deletiondate:chararray, score:int, viewcount:int, body:chararray, owneruserid:int, ownerdisplayname:chararray, lasteditoruserId:int, lasteditordisplayname:chararray, lasteditdate:chararray, lastactivitydate:chararray, title:chararray, tags:chararray, answercount:int, commentcount:int, favoritecount:int, closeddate:chararray, communityowneddate:chararray );
```

```
grunt> DESCRIBE stackdata;
```

B. Now, we do not need every field present in the data. Therefore, we generate a new table with the required fields only and also remove the empty records present in the data.

```
grunt> pickCols = FOREACH stackdata GENERATE id, score, viewcount, owneruserid, title, tags, (REPLACE(body,'[\r\n]+',' ')) AS body;
```

```
grunt> DESCRIBE pickCols;
```

C. Further, we need to remove the duplicate records present in the data. So, removing the same using the DISTINCT function.

```
grunt> datadistinct = DISTINCT pickCols;
```

D. Also, the additional records after the 200,000 records count is completed need to be removed. Therefore, removing the same using the LIMIT function and thereafter reverifying the count of the remaining records using the COUNT_STAR function which return the value 200,000!

```
grunt> datalimit = LIMIT datadistinct 200000;
```

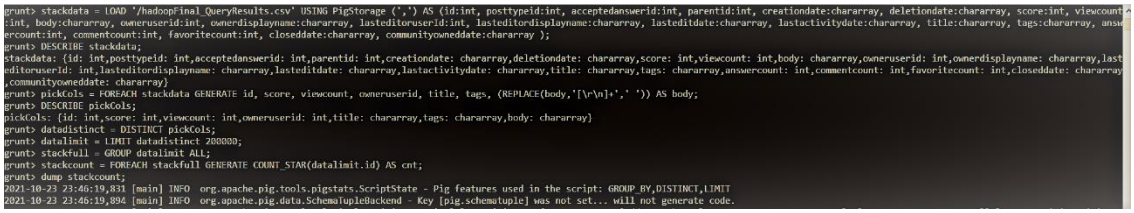
```
grunt> stackfull = GROUP datalimit ALL;
```

```
grunt> stackcount = FOREACH stackfull GENERATE COUNT_STAR(datalimit.id) AS cnt;
```

```
grunt> dump stackcount;
```

E. Lastly, creating a new file folder to save this cleaned up data.

```
grunt> STORE datalimit INTO 'Pig_QueryResults' USING PigStorage(',');
```


grunt> stackdata = LOAD '/hadoopFinal_QueryResults.csv' USING PigStorage (',') AS (id:int, posttypeid:int, acceptedanswerid:int, parentid:int, creationdate:chararray, deletiondate:chararray, score:int, viewcount:int, body:chararray, owneruserid:int, ownerdisplayname:chararray, lasteditoruserId:int, lasteditordisplayname:chararray, lasteditdate:chararray, lastactivitydate:chararray, title:chararray, tags:chararray, answercount:int, commentcount:int, favoritecount:int, closeddate:chararray, communityowneddate:chararray);
grunt> DESCRIBE stackdata;
stackdata: {id: int,posttypeid: int,acceptedanswerid: int,parentid: int,creationdate: chararray,deletiondate: chararray,score: int,viewcount: int,body: chararray,owneruserid: int,ownerdisplayname: chararray,lasteditoruserId: int,lasteditordisplayname: chararray,lasteditdate: chararray,lastactivitydate: chararray,title: chararray,tags: chararray,answercount: int,commentcount: int,favoritecount: int,closeddate: chararray,communityowneddate: chararray}
grunt> pickCols = FOREACH stackdata GENERATE id, score, viewcount, owneruserid, title, tags, (REPLACE(body,'[\r\n]+',' ')) AS body;
grunt> DESCRIBE pickCols;
pickCols: {id: int,score: int,viewcount: int,owneruserid: int,title: chararray,tags: chararray,body: chararray}
grunt> datadistinct = DISTINCT pickCols;
grunt> datalimit = LIMIT datadistinct 200000;
grunt> stackfull = GROUP datalimit ALL;
grunt> stackcount = FOREACH stackfull GENERATE COUNT_STAR(datalimit.id) AS cnt;
grunt> dump stackcount;
2021-10-23 23:46:19,891 [main] INFO org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: GROUP_BY,DISTINCT,LIMIT
2021-10-23 23:46:19,894 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.