# IST769 Homework Submission #5

## Basic Information

Your Name: Thulasi Ram Ruppa Krishnan   
Your SUID: 460746269  
Your Email: truppakr@syr.edu  
Date Due: 4-29-2020   
Homework #: HW5

## Instructions

For each answer, please include your answer as text, and any screenshot(s) which demonstrate your answer was executed. Most importantly, make sure to include evidence your answer is correct. This will most likely be a screenshot. If you had issues, problems, or had to make assumptions include them in your answer.

## Answers:

commands required to complete the task.

1. Upload all the documents in **datasets/text** into a folder called **text** in HDFS. What HDFS command must you run to verify the files are there after they are uploaded? Your answer should consist of the command you typed to complete the task.

To upload the documents in hdfs , the below command is used

**docker-compose exec cloudera bash -c "su -l cloudera"**

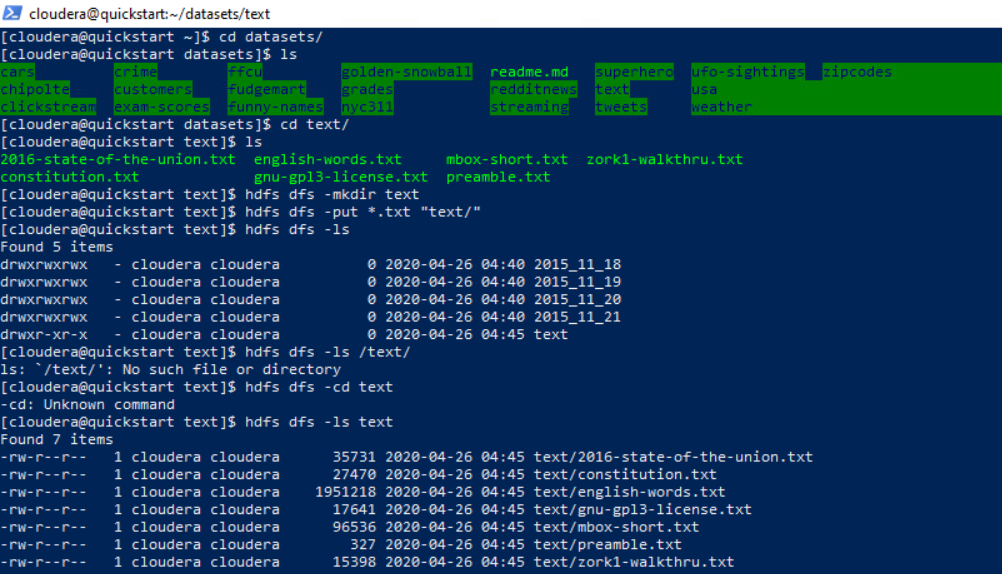
**hdfs dfs -mkdir text**

**hdfs dfs -put \*.txt “text/”**

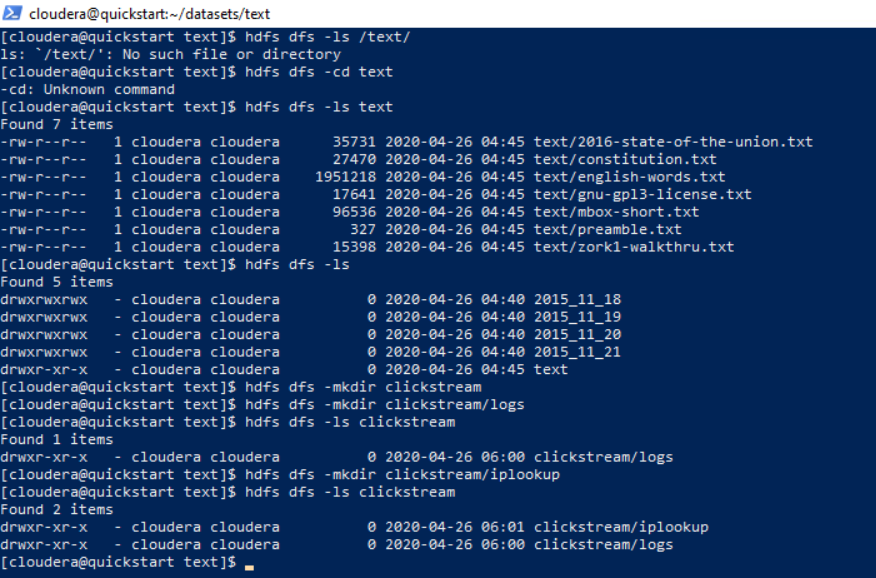
And the following command to check the files if they are uploaded

**hdfs dfs -ls text**

Screenshot of the command line after executing the above commands are attached here



1. In this part you will upload the **clickstream** dataset to HDFS. Specifically, create a **clickstream** folder in HDFS, then create a **logs** and **iplookup** folder inside the clickstream folder. Upload all of the **\*.log** files from the **datasets/clickstream** local folder into **clickstream/logs** in HDFS. Upload the **ip\_lookup.csv** file from the same folder into **clickstream/iplookup** on HDFS. Verify the files are there. Your answer should consist of the commands you typed to complete the task.



1. Use the MapReduce examples:  
   export MREX=/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar   
   to perform a wordcount on the 2016 State of the Union address, saving the output to the HDFS folder **sotu2016**. Write down the commands to complete the task. How many times does the word **are** appear in the 2016 State of the Union address? Describe a process which could be done to make the wordcount more useful?

**Additional process which makes the word count more useful.**

We can output the word count data with counts of individual word in descending order. This will be helpful to look at the top n word frequencies and chart them out. Also, we can take the high frequency words easily for further modeling and investigation

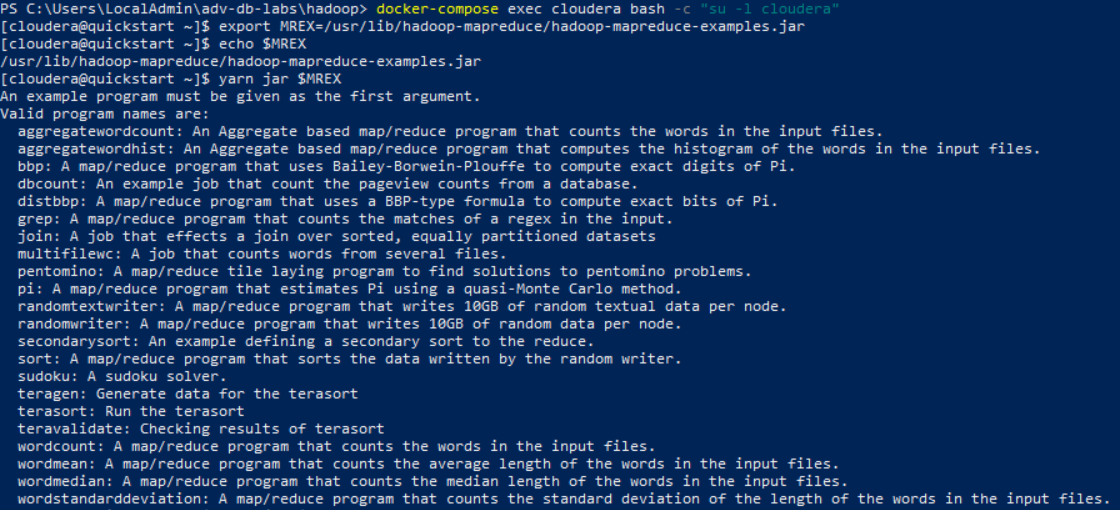
**Command**

**docker-compose exec cloudera bash -c "su -l cloudera"**

**export MREX=/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar**

**yarn jar $MREX**

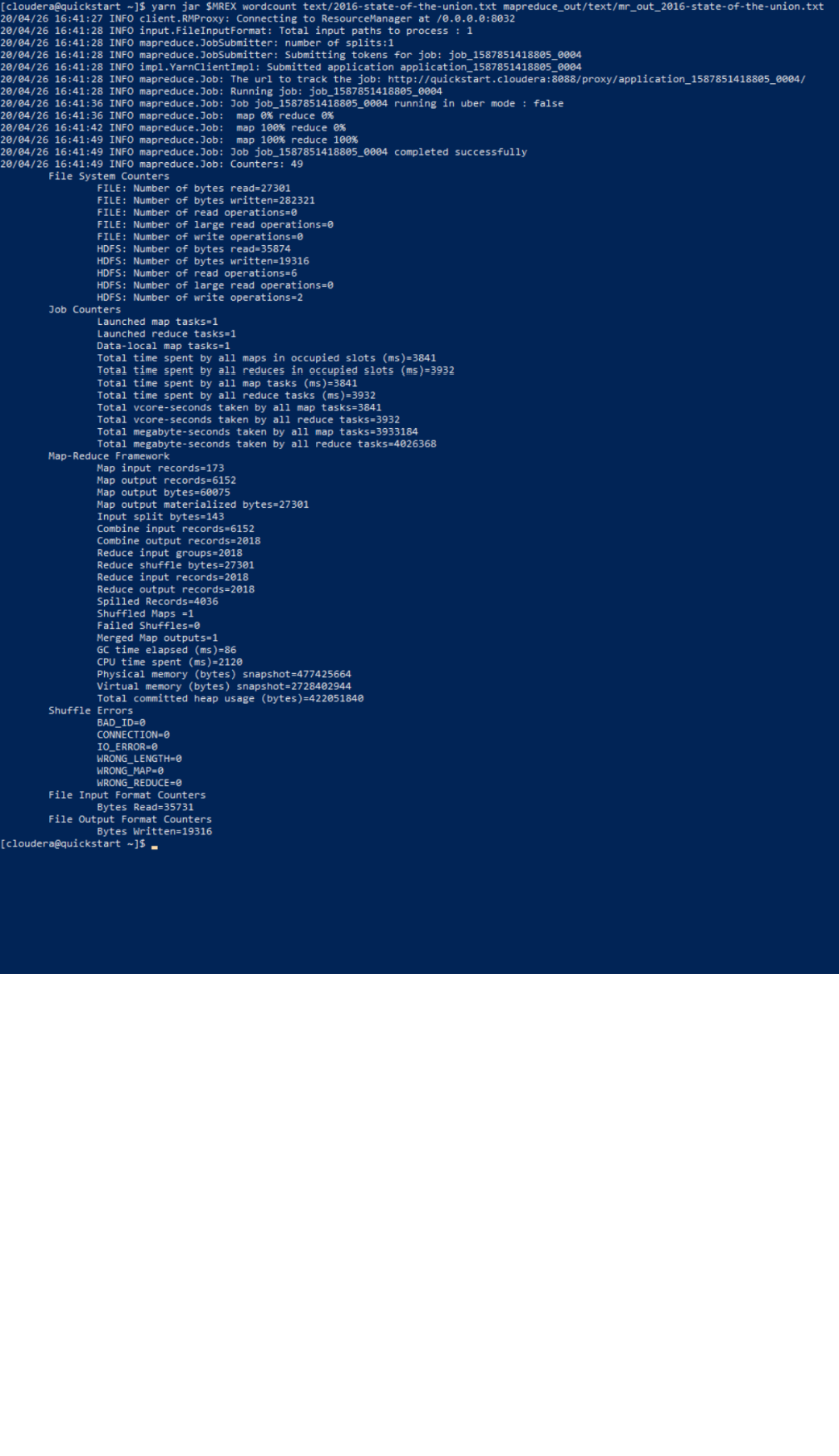
**Result**



**Command**

**yarn jar $MREX wordcount text/2016-state-of-the-union.txt mapreduce\_out/text/mr\_out\_2016-state-of-the-union.txt**

**Result**

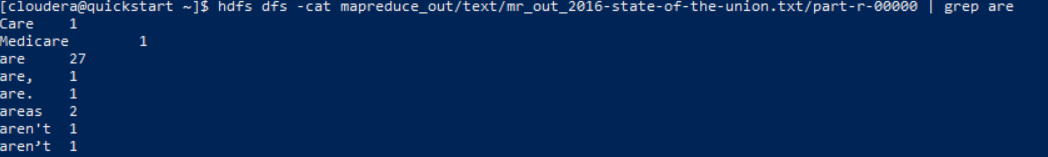


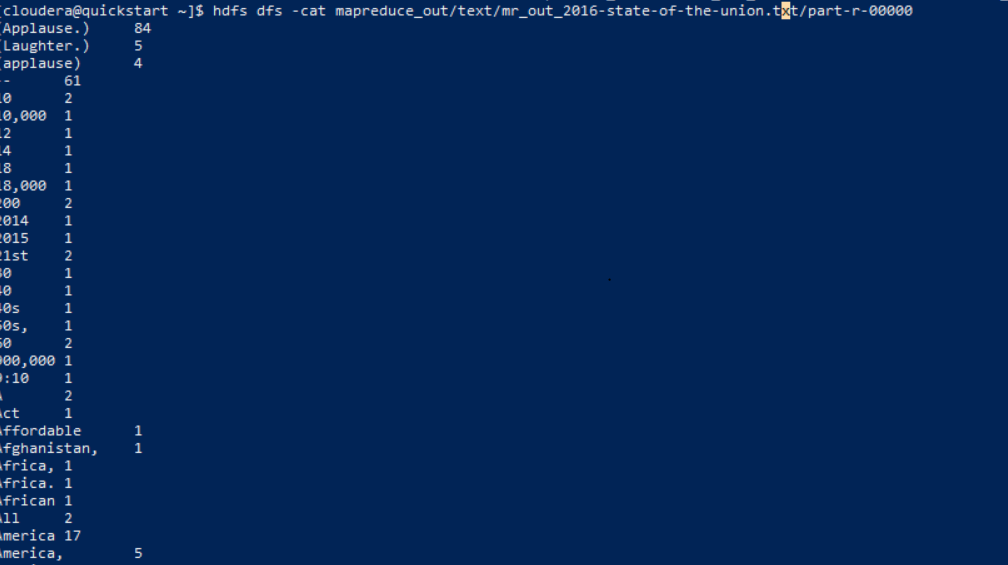
**Command**

**hdfs dfs -cat mapreduce\_out/text/mr\_out\_2016-state-of-the-union.txt/part-r-0000 | grep are**

**hdfs dfs -cat mapreduce\_out/text/mr\_out\_2016-state-of-the-union.txt/part-r-0000**

**Result**





1. Type the following command to import the **fudgemart\_v3** database into the local **mysql** instance on the Hadoop client:  
   mysql -u root -p < ~/datasets/fudgemart/mysql.sql   
   The password is **cloudera**. Write down the commands you used to complete these tasks:  
   Use the **sqoop** utility to verify there are tables in the database by listing them from the **fudgemart\_v3** database. Next write a sqoop command to import Fudgemart products in the ‘Clothing’ department into a HDFS folder **/user/cloudera/fudgemart-clothing**

**Command**

**Window 1**

**docker-compose exec cloudera bash -c "su -l cloudera"**

**sqoop list-databases –connect jdbc:mysql://cloudera/fudgemart\_v3 –username=root –password=cloudera**

**sqoop list-tables –connect jdbc:mysql://cloudera/fudgemart\_v3 –username=root –password=cloudera**

**Window 2**

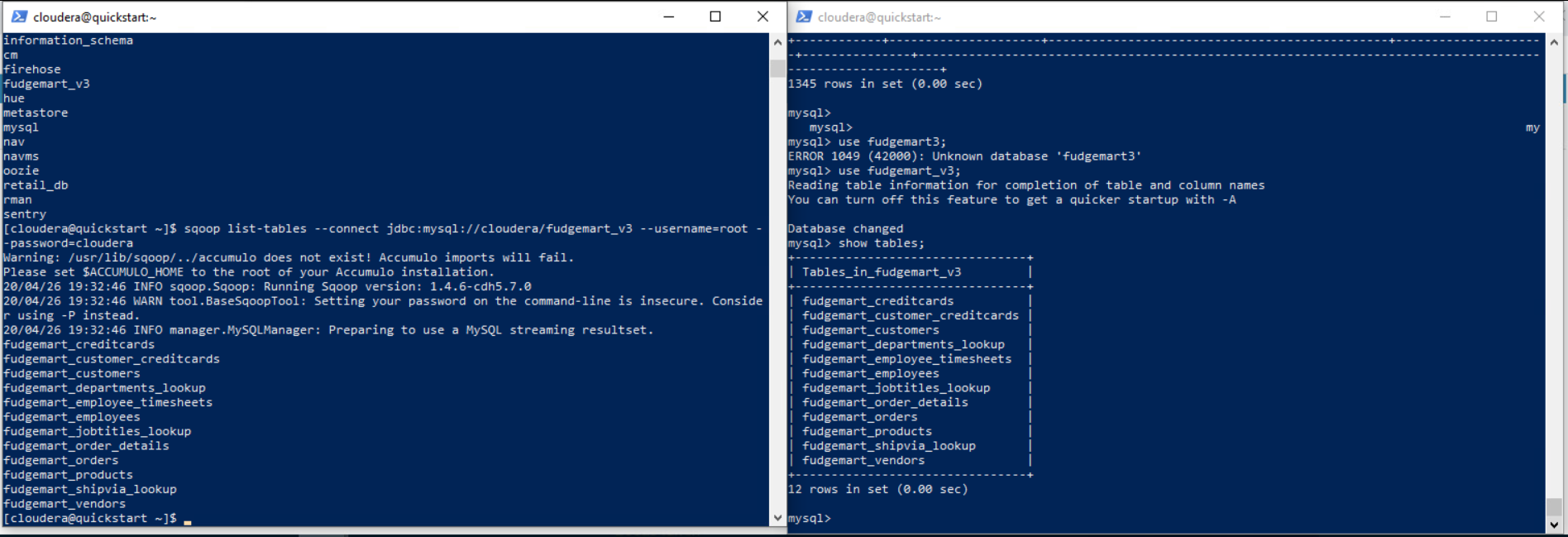
**docker-compose exec cloudera bash -c "su -l cloudera"**

**mysql -u root -p**

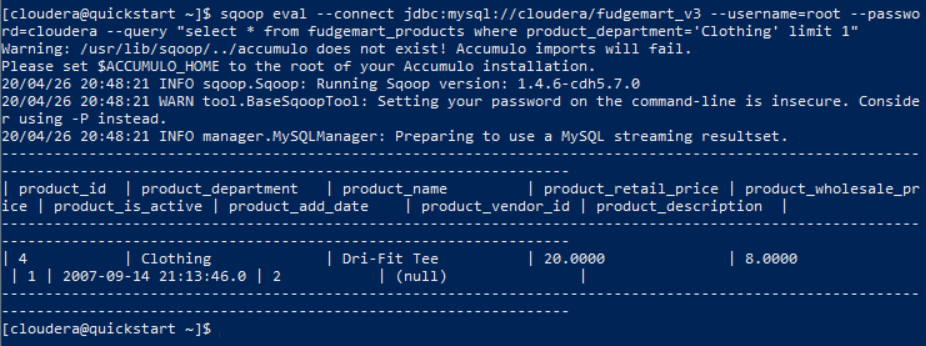
**show databases;**

**use fudgemart\_v3;**

**show tables;**



**sqoop eval –connect jdbc:mysql://cloudera/fudgemart\_v3 –username=root –password=cloudera –query ”select \*from fudgemart\_products where product\_department=’Clothing’ limit 1 ”**



**Sqoop import \**

**--connect jdbc:mysql://cloudera/fudgemart\_v3 \**

**--username=root –password=cloudera \**

**--query “select \* from fudgemart\_products p where p.product\_department=’clothing’ and \$CONDITIONS” \**

**--target-dir /user/cloudera/fudgemart-clothing \**

**--as-textfile**

**--split-by product\_name**





**Sqoop import \**

**--connect jdbc:mysql://cloudera/fudgemart\_v3 \**

**--username=root –password=cloudera \**

**--table fudgemart\_products \**

**--target-dir /data/fudgemart\_v3/fudgemart\_products \**

**--as-textfile**





**hdfs dfs -ls fudgemart-clothing/**

**hdfs dfs -cat fudgemart-clothing/part-m-00001**



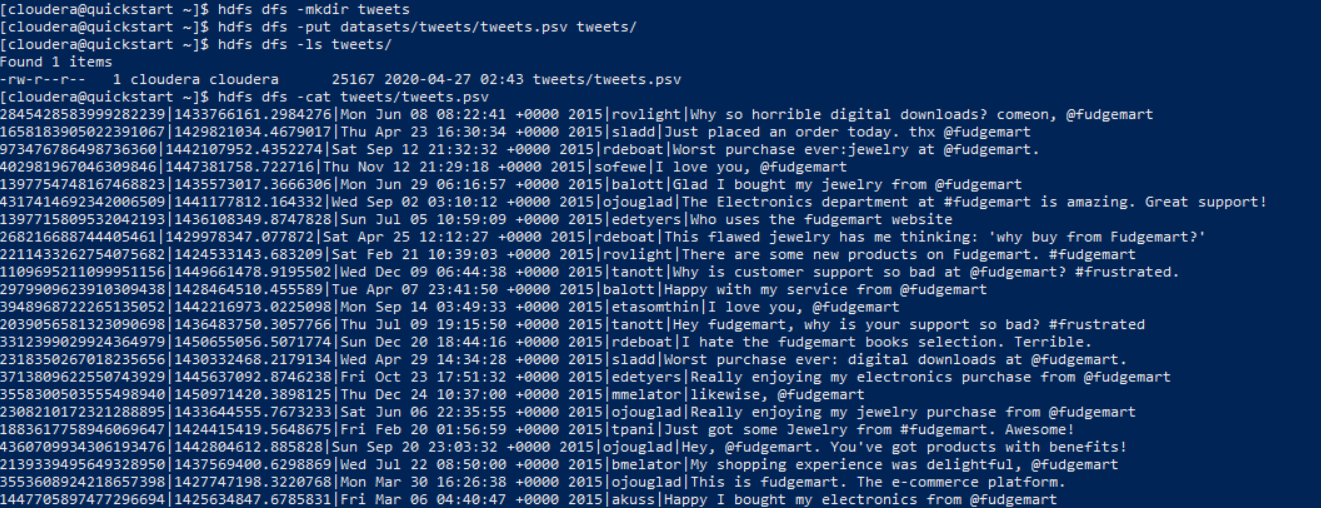
1. Let’s import HDFS data into MySQL. Record each command you type as your solution:
   1. Load **datasets/tweets/tweets.psv** into the HDFS folder **tweets**

**hdfs dfs -mkdir tweets**

**hdfs dfs -put datasets/tweets/tweets.psv tweets/**

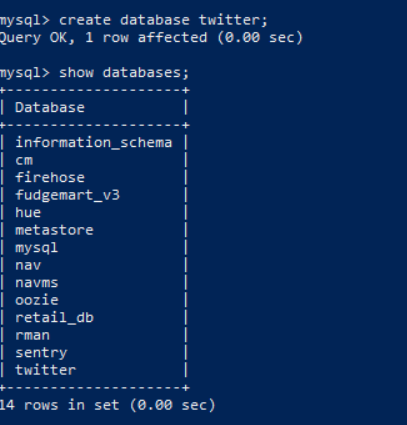
**hdfs dfs -ls tweets/**

**hdfs dfs -cat tweets/tweets.psv**

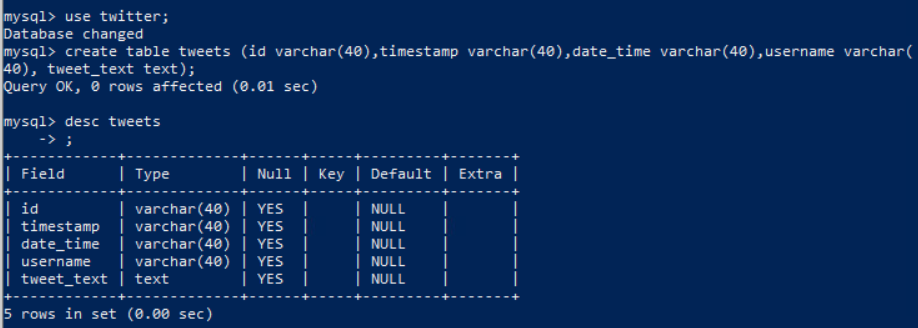


* 1. Login to MySQL: mysql -u root -p The password is **cloudera**. Create a database **twitter**

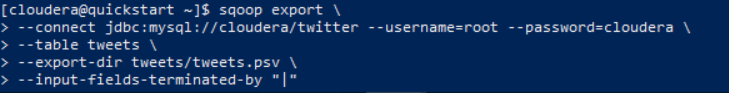
**create database twitter;**



* 1. Create a table called **tweets** inside the database **twitter** the table should have columns for id, timestamp, date time, username, and tweet\_text.



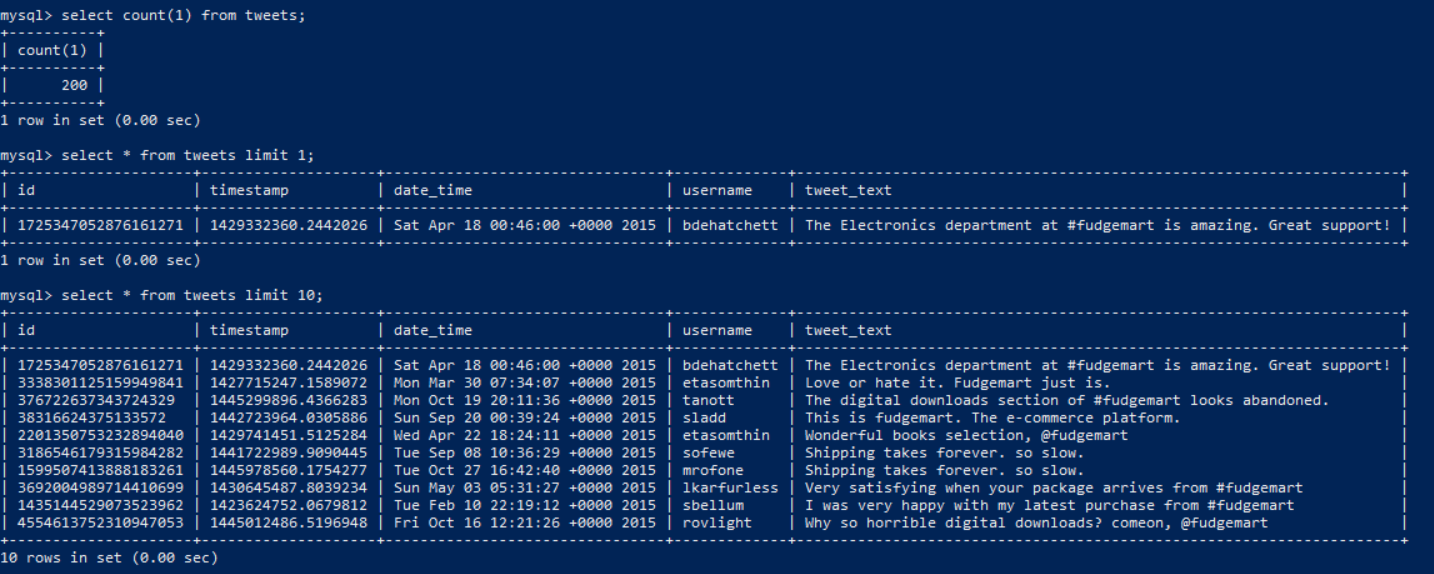
* 1. Export the data from HDFS into the MySQL table.  
     **TIPS:** If your SQOOP job fails it is likely due to the table constraints such as selecting a data type too small for the imported data. This will help you to ensure your chosen data types will work.





**Select count(1) from tweets;**

**Select \* from tweets limit 10;**



# Tear-Down

When you are finished with the homework you should stop the environment:

1. From the terminal window where you typed docker-compose up -d type in the following:  
   docker-compose stop

