1) In this section we are going to use Hive to run a few queries over the Hadoop framework. These instructions assume that you are starting from a working Hadoop installation. It should be sufficient to start your instance and the Hadoop framework on it.

   Hive commands are listed in **Calibri bold font**

   a) Download and install Hive:

      cd
      (this command is there to make sure you start from home directory, on the same level as where hadoop is located)
      wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/apache-hive-2.0.1-bin.tar.gz
      gunzip apache-hive-2.0.1-bin.tar.gz
      tar xvf apache-hive-2.0.1-bin.tar

      set the environment variables (can be automated by adding these lines in ~/.bashrc). If you don't, you will have to set these variables every time you use Hive.
      export HIVE_HOME=/home/ec2-user/apache-hive-2.0.1-bin
      export PATH=$HIVE_HOME/bin:$PATH

      $HADOOP_HOME/bin/hadoop fs -mkdir /tmp
      $HADOOP_HOME/bin/hadoop fs -mkdir /user/hive/warehouse
      (if you get an error here, it means that /user/hive does not exist yet. Fix that by running
      $HADOOP_HOME/bin/hadoop fs -mkdir -p /user/hive/warehouse instead)

      $HADOOP_HOME/bin/hadoop fs -chmod g+w /tmp
      $HADOOP_HOME/bin/hadoop fs -chmod g+w /user/hive/warehouse

      We are going to use Vehicle data (originally from
      http://www.fueleconomy.gov/feg/download.shtml)

      You can get the already unzipped, comma-separated file from here:
      wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/vehicles.csv

      You can take a look at the data file by either
      nano vehicles.csv or
      more vehicles.csv (you can press space to scroll and q or Ctrl-C to break out)

      Note that the first row in the data is the list of column names. What follows after commands that start Hive, is the table that you will create in Hive loading the first 5 columns. Hive is not particularly sensitive about invalid or partial data, hence if we only define the first 5 columns, it will simply load the first 5 columns and ignore the rest. You can see the description of all the columns here (atvtype was added later)
      http://www.fueleconomy.gov/feg/ws/index.shtml#vehicle

Create the ec2-user directory on the HDFS side (absolute path commands should work anywhere and not just in Hadoop directory as bin/hadoop does). Here, we are creating the user "home" directory on the HDFS side.

hadoop fs -mkdir /user/ec2-user/

Run hive (from the hive directory because of the first command below):
cd $HIVE_HOME
$HIVE_HOME/bin/schematool -initSchema -dbType derby
(NOTE: This command initializes the database metastore. If you need to restart/reformat or see errors related to meta store, run rm -rf metastore_db/ and then repeat the above initSchema command)
bin/hive

You can now create a table by pasting this into the Hive terminal:

```
CREATE TABLE VehicleData (
barrels08 FLOAT, barrelsA08 FLOAT,
charge120 FLOAT, charge240 FLOAT,
city08 FLOAT)
ROW FORMAT DELIMITED FIELDS
TERMINATED BY ',' STORED AS TEXTFILE;
```


You can load the data (from the local file system, not HDFS) using:


```
LOAD DATA LOCAL INPATH '/home/ec2-user/vehicles.csv'
OVERWRITE INTO TABLE VehicleData;
```


(NOTE: If you downloaded vehicles.csv file into the hive directory, you have to change file name to /home/ec2-user/apache-hive-2.0.1-bin/vehicles.csv instead)

Verify that your table had successfully loaded by running
SELECT COUNT(*) FROM VehicleData;
(Copy the query output and report how many rows you got as an answer.)

Run a couple of HiveQL queries to verify that everything is working properly:

SELECT MIN(barrels08), AVG(barrels08), MAX(barrels08) FROM VehicleData;
(copy the output from that query)

SELECT (barrels08/city08) FROM VehicleData;
(you do not need to report the output from that query, but report "Time taken")

Next, we are going to output three of the columns into a separate file (as a way to transform data for further manipulation that you may be interested in)

```
INSERT OVERWRITE DIRECTORY 'ThreeColExtract'
SELECT barrels08, city08, charge120
```

FROM VehicleData;

You can now exit Hive by running **exit;**

And verify that the new output file has been created (the file will be called 000000_0) The file would be created in HDFS in user home directory (/user/ec2-user/ThreeColExtract)

Report the size of the newly created file.

Next, you should go back to the Hive terminal, create a new table that is going to load 8 columns instead of 5 in our example (i.e. create and load a new table that defines 8 columns by including columns city08U,cityA08,cityA08U) and use Hive to generate a new output file containing only the city08U and cityA08U columns from the vehicles.csv file. Report the size of that output file as well.

Submit a single document containing your written answers. Be sure that this document contains your name and "CSC 555 Assignment 2" at the top.