# **Execution Script:**

Copy the excel files (4) and paste it onto the VMWare’s desktop (blue screen)

Move them to a folder for easier access e.g. /home/training using the File Browser explorer

Open a terminal session, start Hive

Type: **sudo service hive-server2 start**

**Type: hive**

On Hue Browser:

Click **Query Editor**, under Database: click ‘default’

Copy paste and execute the create tables commands one by one for the 3 tables first:

Geolocation

Trucks

Truck\_mg

1. CREATE TABLE geolocation ( truckid string, driverid string, event string, latitude DOUBLE, longitude DOUBLE, city string, state string, velocity BIGINT, event\_ind BIGINT, idling\_ind BIGINT ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE TBLPROPERTIES ("skip.header.line.count"="1");
2. CREATE TABLE trucks(driverid string, truckid string, model string, jun13\_miles bigint, jun13\_gas bigint, may13\_miles bigint, may13\_gas bigint, apr13\_miles bigint, apr13\_gas bigint, mar13\_miles bigint, mar13\_gas bigint, feb13\_miles bigint, feb13\_gas bigint, jan13\_miles bigint, jan13\_gas bigint, dec12\_miles bigint, dec12\_gas bigint, nov12\_miles bigint, nov12\_gas bigint, oct12\_miles bigint, oct12\_gas bigint, sep12\_miles bigint, sep12\_gas bigint, aug12\_miles bigint, aug12\_gas bigint, jul12\_miles bigint, jul12\_gas bigint, jun12\_miles bigint, jun12\_gas bigint,may12\_miles bigint, may12\_gas bigint, apr12\_miles bigint, apr12\_gas bigint, mar12\_miles bigint, mar12\_gas bigint, feb12\_miles bigint, feb12\_gas bigint, jan12\_miles bigint, jan12\_gas bigint, dec11\_miles bigint, dec11\_gas bigint, nov11\_miles bigint, nov11\_gas bigint, oct11\_miles bigint, oct11\_gas bigint, sep11\_miles bigint, sep11\_gas bigint, aug11\_miles bigint, aug11\_gas bigint, jul11\_miles bigint, jul11\_gas bigint, jun11\_miles bigint, jun11\_gas bigint, may11\_miles bigint, may11\_gas bigint, apr11\_miles bigint, apr11\_gas bigint, mar11\_miles bigint, mar11\_gas bigint, feb11\_miles bigint, feb11\_gas bigint, jan11\_miles bigint, jan11\_gas bigint, dec10\_miles bigint, dec10\_gas bigint, nov10\_miles bigint, nov10\_gas bigint, oct10\_miles bigint, oct10\_gas bigint, sep10\_miles bigint, sep10\_gas bigint, aug10\_miles bigint, aug10\_gas bigint, jul10\_miles bigint, jul10\_gas bigint, jun10\_miles bigint, jun10\_gas bigint, may10\_miles bigint, may10\_gas bigint, apr10\_miles bigint, apr10\_gas bigint, mar10\_miles bigint, mar10\_gas bigint, feb10\_miles bigint, feb10\_gas bigint, jan10\_miles bigint, jan10\_gas bigint, dec09\_miles bigint, dec09\_gas bigint, nov09\_miles bigint, nov09\_gas bigint, oct09\_miles bigint, oct09\_gas bigint, sep09\_miles bigint, sep09\_gas bigint, aug09\_miles bigint, aug09\_gas bigint, jul09\_miles bigint, jul09\_gas bigint, jun09\_miles bigint, jun09\_gas bigint, may09\_miles bigint, may09\_gas bigint, apr09\_miles bigint, apr09\_gas bigint, mar09\_miles bigint, mar09\_gas bigint, feb09\_miles bigint, feb09\_gas bigint, jan09\_miles bigint, jan09\_gas bigint) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE TBLPROPERTIES ("skip.header.line.count"="1");
3. CREATE TABLE trucks\_mg(driverid string, truckid string, model string, Tdate string, miles bigint, gas bigint ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE TBLPROPERTIES ("skip.header.line.count"="1");

**Loading Data into the tables:**

On hive prompt:

load data local inpath '/home/training/geolocation.csv' into table geolocation;

load data local inpath '/home/training/trucks.csv' into table trucks;

load data local inpath '/home/training/trucks\_mg.csv' into table trucks\_mg;

**Check to see if the data has been loaded into tables:**

select \* from geolocation limit 5;

select \* from trucks limit 1;

select \* from trucks\_mg limit 10;

**create all the other tables**

1. CREATE TABLE truck\_mileage AS SELECT truckid, driverid, rdate, miles, gas, miles / gas mpg FROM trucks LATERAL VIEW stack(54, 'jun13',jun13\_miles,jun13\_gas,'may13',may13\_miles,may13\_gas,'apr13',apr13\_miles,ap r13\_gas,'mar13',mar13\_miles,mar13\_gas,'feb13',feb13\_miles,feb13\_gas,'jan13',jan13\_ miles,jan13\_gas,'dec12',dec12\_miles,dec12\_gas,'nov12',nov12\_miles,nov12\_gas,'oct12' ,oct12\_miles,oct12\_gas,'sep12',sep12\_miles,sep12\_gas,'aug12',aug12\_miles,aug12\_gas, 'jul12',jul12\_miles,jul12\_gas,'jun12',jun12\_miles,jun12\_gas,'may12',may12\_miles,may1 2\_gas,'apr12',apr12\_miles,apr12\_gas,'mar12',mar12\_miles,mar12\_gas,'feb12',feb12\_mi les,feb12\_gas,'jan12',jan12\_miles,jan12\_gas,'dec11',dec11\_miles,dec11\_gas,'nov11',no v11\_miles,nov11\_gas,'oct11',oct11\_miles,oct11\_gas,'sep11',sep11\_miles,sep11\_gas,'au g11',aug11\_miles,aug11\_gas,'jul11',jul11\_miles,jul11\_gas,'jun11',jun11\_miles,jun11\_gas ,'may11',may11\_miles,may11\_gas,'apr11',apr11\_miles,apr11\_gas,'mar11',mar11\_miles, mar11\_gas,'feb11',feb11\_miles,feb11\_gas,'jan11',jan11\_miles,jan11\_gas,'dec10',dec10\_ miles,dec10\_gas,'nov10',nov10\_miles,nov10\_gas,'oct10',oct10\_miles,oct10\_gas,'sep10', sep10\_miles,sep10\_gas,'aug10',aug10\_miles,aug10\_gas,'jul10',jul10\_miles,jul10\_gas,'ju n10',jun10\_miles,jun10\_gas,'may10',may10\_miles,may10\_gas,'apr10',apr10\_miles,apr1 0\_gas,'mar10',mar10\_miles,mar10\_gas,'feb10',feb10\_miles,feb10\_gas,'jan10',jan10\_mil es,jan10\_gas,'dec09',dec09\_miles,dec09\_gas,'nov09',nov09\_miles,nov09\_gas,'oct09',oc t09\_miles,oct09\_gas,'sep09',sep09\_miles,sep09\_gas,'aug09',aug09\_miles,aug09\_gas,'jul 09',jul09\_miles,jul09\_gas,'jun09',jun09\_miles,jun09\_gas,'may09',may09\_miles,may09\_g as,'apr09',apr09\_miles,apr09\_gas,'mar09',mar09\_miles,mar09\_gas,'feb09',feb09\_miles, feb09\_gas,'jan09',jan09\_miles,jan09\_gas ) dummyalias AS rdate, miles, gas;
2. CREATE TABLE avg\_mileage AS SELECT truckid, avg(mpg) avgmpg FROM truck\_mileage GROUP BY truckid;
3. CREATE TABLE DriverMileage AS SELECT driverid, sum(miles) totmiles FROM truck\_mileage GROUP BY driverid;
4. a) CREATE TABLE riskfactor (driverid string, events bigint, totmiles bigint, riskfactor float) ;

check if data has been populated in them using above select \* syntax.

**Create new table california\_population:**

1. CREATE TABLE california\_population ( city string, population BIGINT, land\_area DOUBLE, population\_sq\_mile DOUBLE, city\_class string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE TBLPROPERTIES ("skip.header.line.count"="1");

**Load data into this new table:**

load data local inpath '/home/training/Use\_California\_Population.csv' into table california\_population;

Via Hue browser:File Browser look into /user/hive/warehouse/ if you see all your tables created.

**Use the below PIG’s script to populate “riskfactor” table:**

**Open a new terminal instance, type PIG**

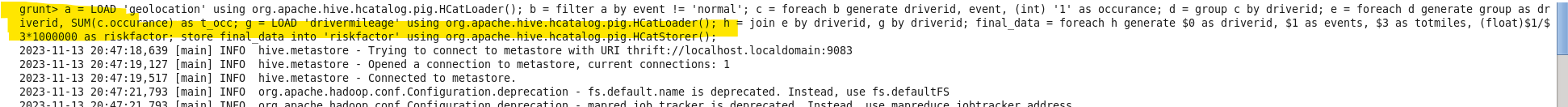
**A screenshot of a computer

Description automatically generated**

Once done, it shows a prompt **grunt>**

**Copy the following in one go and paste it on the grunt> prompt.**

a = LOAD 'geolocation' using org.apache.hive.hcatalog.pig.HCatLoader(); b = filter a by event != 'normal'; c = foreach b generate driverid, event, (int) '1' as occurance; d = group c by driverid; e = foreach d generate group as driverid, SUM(c.occurance) as t\_occ; g = LOAD 'drivermileage' using org.apache.hive.hcatalog.pig.HCatLoader(); h = join e by driverid, g by driverid; final\_data = foreach h generate $0 as driverid, $1 as events, $3 as totmiles, (float)$1/$3\*1000000 as riskfactor; store final\_data into 'riskfactor' using org.apache.hive.hcatalog.pig.HCatStorer();



This takes a while to run and you’ll see the last message as follows:

