ECS 165a HW4 Writeup
By Shyam Pinnipati and Nguyen Ngo

For this project, we chose to use these tables for our queries:

```
CREATE TABLE eia_co2_transportation_2014
(
  msn text,
  yyyymm bigint,
  value bigint,
  column_order bigint,
  description text,
  unit text
);

CREATE TABLE eia_mkwh_2014
(
  msn text,
  yyyymm bigint,
  value bigint,
  column_order bigint,
  description text,
  unit text
);

CREATE TABLE eia_co2_electric_2014
(
  msn text,
  yyyymm bigint,
  value bigint,
  column_order bigint,
  description text,
  unit text
);

CREATE TABLE dayv2pub
(
  houseid bigint,
  personid bigint,
  frsthm bigint,
  outoftwn bigint,
  ontd_p1 bigint,
  ontd_p2 bigint,
  ontd_p3 bigint,
```

ontd_p4 bigint,
ontd_p5 bigint,
ontd_p6 bigint,
ontd_p7 bigint,
ontd_p8 bigint,
ontd_p9 bigint,
ontd_p10 bigint,
ontd_p11 bigint,
ontd_p12 bigint,
ontd_p13 bigint,
ontd_p14 bigint,
ontd_p15 bigint,
tdcaseid double precision,
hh_hisp bigint,
hh_race bigint,
driver bigint,
r_sex bigint,
worker bigint,
drvrcnt bigint,
hhfaminc bigint,
hhsize bigint,
hhvehcnt bigint,
numadlt bigint,
flag100 bigint,
lif_cyc bigint,
trippurp text,
awayhome bigint,
cdivmsar bigint,
census_d bigint,
census_r bigint,
drop_prk bigint,
drvr_flg bigint,
educ bigint,
endtime bigint,
hh_ontd bigint,
hhmemdrv bigint,
hhresp bigint,
hhstate text,
hhstfips bigint,
intstate bigint,
msacat bigint,
msasize bigint,
nonhhcnt bigint,

numontrp bigint,
paytoll bigint,
prmact bigint,
proxy bigint,
psgr_flg bigint,
r_age bigint,
rail bigint,
strttime bigint,
tracc1 bigint,
tracc2 bigint,
tracc3 bigint,
tracc4 bigint,
tracc5 bigint,
tracctm bigint,
travday bigint,
tregr1 bigint,
tregr2 bigint,
tregr3 bigint,
tregr4 bigint,
tregr5 bigint,
tregrtm bigint,
trpaccmp bigint,
trphhacc bigint,
trphhveh bigint,
trptrans bigint,
trvl_min bigint,
trvlcmin bigint,
trwaittm bigint,
urban bigint,
urbansize bigint,
urbrur bigint,
useintst bigint,
usepubtr bigint,
vehid bigint,
whodrove bigint,
whyfrom bigint,
whyto bigint,
whytrp1s bigint,
wrkcount bigint,
dweltime bigint,
whytrp90 bigint,
tdtrpnum bigint,
tdwknd bigint,

```
  tdaydate bigint,
  trpmiles double precision,
  wttrdfin double precision,
  vmt_mile double precision,
  pubtrans bigint,
  homeown bigint,
  hometype bigint,
  hbhur text,
  htresdn bigint,
  hthtnrnt bigint,
  htppopdn bigint,
  hteempdn bigint,
  hbresdn bigint,
  hbhtnrnt bigint,
  hbppopdn bigint,
  gasprice double precision,
  vehtype bigint,
  hh_cbsa text,
  hhc_msa text
)

CREATE TABLE hhv2pub
(
  houseid bigint,
  varstrat bigint,
  wthhfin double precision,
  drvrcnt bigint,
  cdivmsar bigint,
  census_d bigint,
  census_r bigint,
  hh_hisp bigint,
  hh_race bigint,
  hhfaminc bigint,
  hhrelatd bigint,
  hhresp bigint,
  hhsize bigint,
  hhstate text,
  hhstfips bigint,
  hhvehcnt bigint,
  homeown bigint,
  hometype bigint,
  msacat bigint,
  msasize bigint,
```

```
  numadlt bigint,
  rail bigint,
  resp_cnt bigint,
  scresp bigint,
  travday bigint,
  urban bigint,
  urbansize bigint,
  urbrur bigint,
  wrkcount bigint,
  tdaydate bigint,
  flag100 bigint,
  lif_cyc bigint,
  cnttdhh bigint,
  hbhur text,
  htresdn bigint,
  hthtnrnt bigint,
  htppopdn bigint,
  hteempdn bigint,
  hbresdn bigint,
  hbhtnrnt bigint,
  hbppopdn bigint,
  hh_cbsa text,
  hhc_msa text
)

CREATE TABLE vehv2pub
(
  houseid bigint,
  wthhfin double precision,
  vehid bigint,
  drvrcnt bigint,
  hhfaminc bigint,
  hhsize bigint,
  hhvehcnt bigint,
  numadlt bigint,
  flag100 bigint,
  cdivmsar bigint,
  census_d bigint,
  census_r bigint,
  hhstate text,
  hhstfips bigint,
  hybrid bigint,
  makecode text,
```

modlcode text,
msacat bigint,
msasize bigint,
od_read bigint,
rail bigint,
travday bigint,
urban bigint,
urbansize bigint,
urbrur bigint,
vehcomm bigint,
vehownmo double precision,
vehyear bigint,
whomain bigint,
wrkcount bigint,
tdaydate bigint,
vehage bigint,
personid bigint,
hh_hisp bigint,
hh_race bigint,
homeown bigint,
hometype bigint,
lif_cyc bigint,
annmiles double precision,
hbhur text,
htresdn bigint,
hthtnrnt bigint,
htppopdn bigint,
hteempdn bigint,
hbresdn bigint,
hbhtnrnt bigint,
hbppopdn bigint,
best_flg bigint,
bestmile double precision,
best_edt bigint,
best_out bigint,
fueltype bigint,
gsyrgal bigint,
gscost double precision,
gstotcst bigint,
epatmpg double precision,
epatmpgf bigint,
eiadmpg double precision,
vehtype bigint,

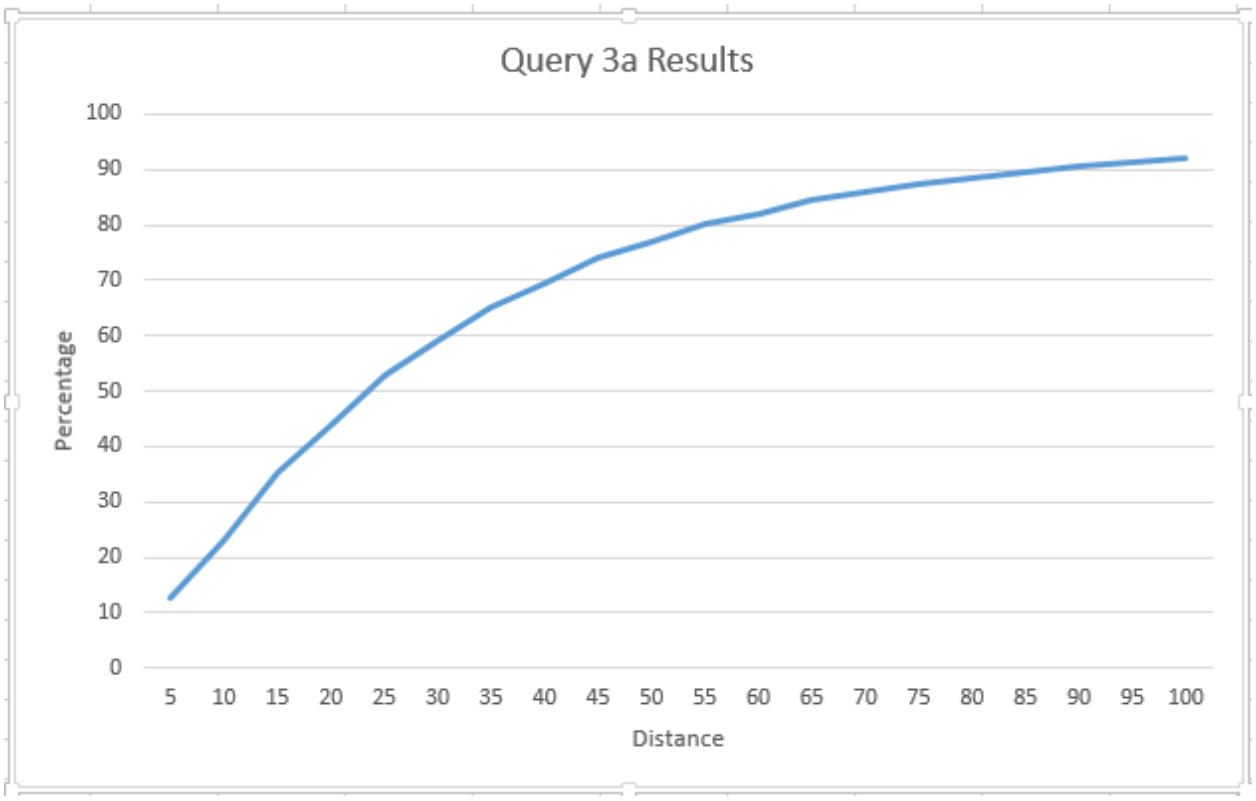
```
  hh_cbsa text,
  hhc_msa text
);
```

We chose to retain all information in the csv files in creating database schemas for the NHTS and EIA data and to create a separate schema for each csv file. The reason why we chose this design is because it allows us more flexibility in creating our queries. In other words, we won't be restricted to which attributes we use because we know that it is already in the database. If we had chosen to select only some attributes in the csv files to create the database, then any future queries we make will be restricted to using that set of attributes. Furthermore, we use the tightest bound that can fit all the data in determining the data type for the attributes. So for any attributes that can be represented as a string, we set its data type to "text", and for any attributes that can be represented as float, we set its data type to "float". Otherwise, we set its data type to "big int".

(For the purpose of completing this assignment, the file PERV2PUB.CSV is not needed, so it is not included in our database schemas.)

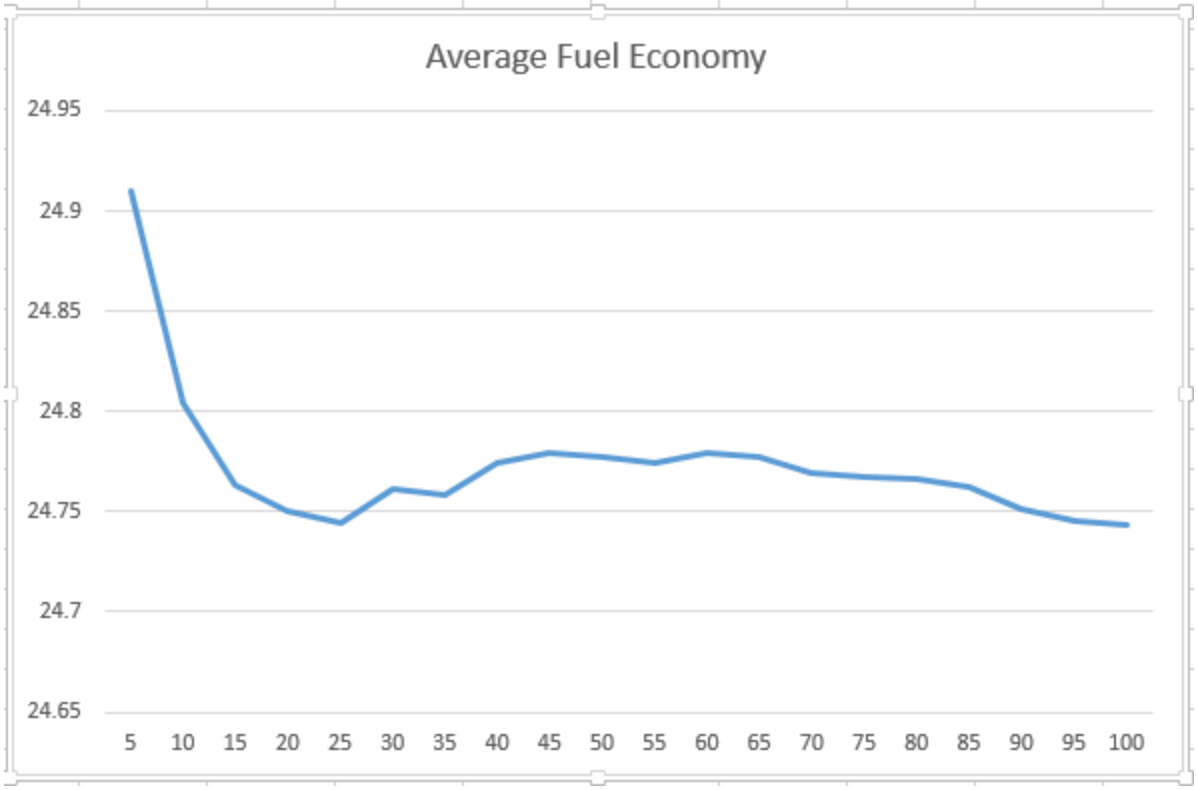
Since our schemas have identical structures to that of their respected csv files, if new datas are available then syncing the database is a simple task. To update the schemas, you can use the UPDATE statement to modify an existing row, the INSERT statement to add a new row, and the DELETE statement to delete a row. If the attribute in the schema is the same as another attribute in a different schema, then we would also have to run the UPDATE statement on the other schema.

These are the results for 3a:



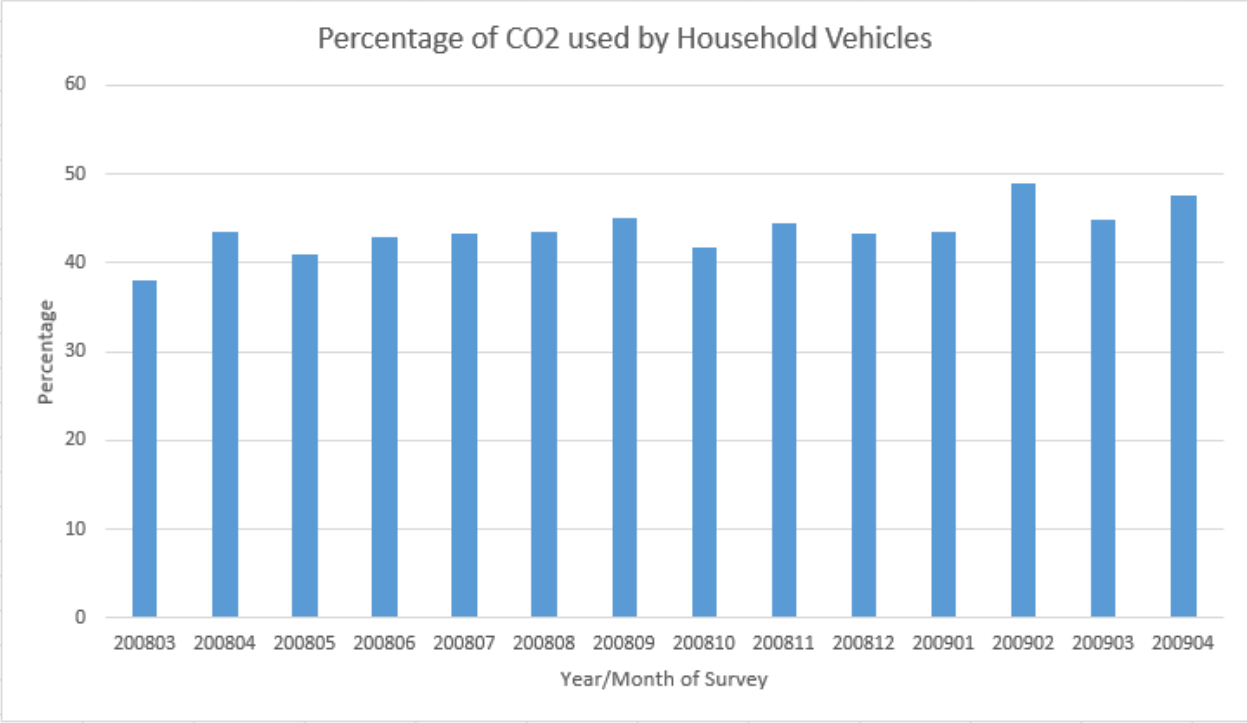
As you can see, the percentage of drivers who drive under the specified distances slowly straightens out as the distance increases. Most drivers did not drive over 100 miles. We assumed that using the houseid, personid, tdaydate (year and month) and travday (day of the week) was enough to differentiate each person's total travel for a specific day. Each tuple is unique to the individual by the tdaydate. If a person travels multiple times per day, the travel miles are accumulated into a single tuple per individual. Duplicates are preserved, but do not have any major effects on the data trends.

These are our results for query 3b:



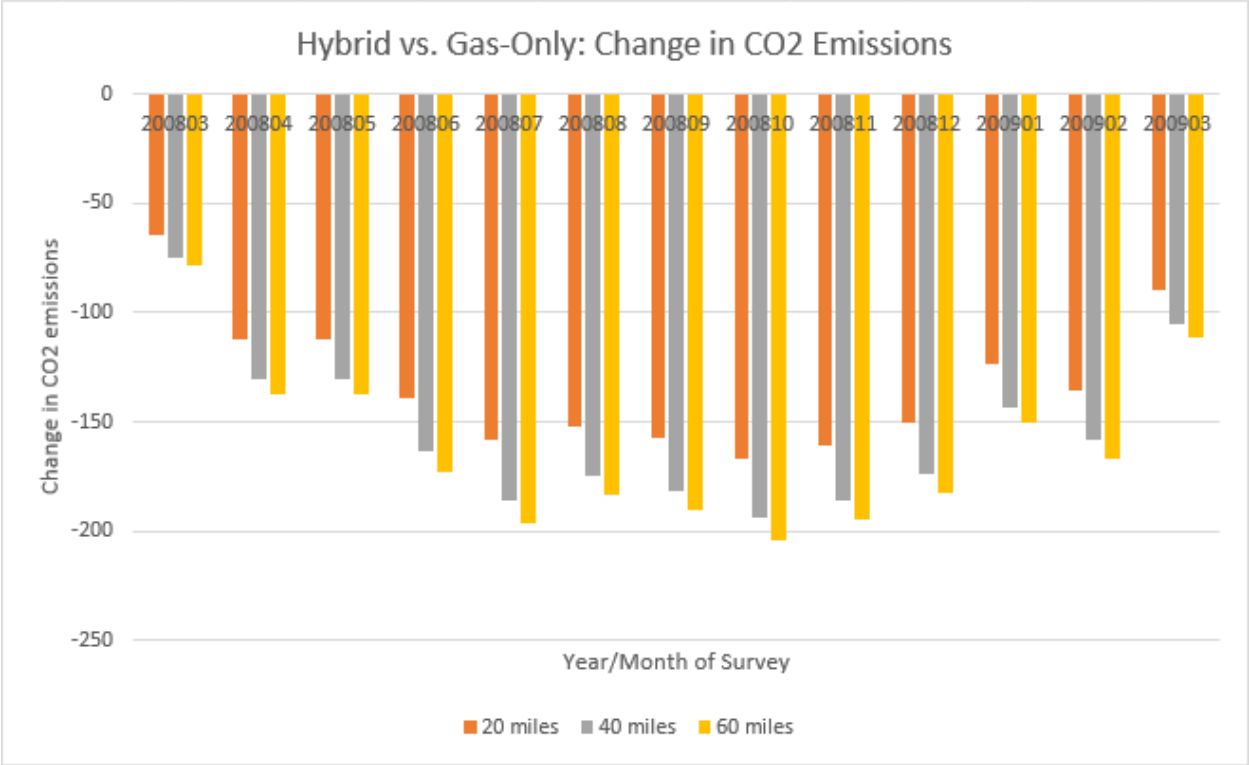
Again, duplicate data is left in, but this doesn't affect the trend. As you can see, the data is mostly in a very tight bound, all around 24.7 to 24.9 miles per gallon. The formula we used to determine the average fuel economy per vehicle was the total miles travelled over the total amount of fuel consumed. To get the gallons per trip, we divided the miles travelled by the EPATMPG of the vehicle.

These are our results for query 3c:



As we travel through the months of the survey, we can see that household vehicles do indeed make up a significant amount of the total carbon dioxide emissions. We came up with these values by finding the amount of carbon dioxide generated by each trip in the survey, scaling the value up to a month's worth, then scaling it once more to match the number of households in the United States. We then divided that amount by the Total Energy Transportation Sector CO2 Emissions from the EIA Transportation CO2 table, and multiplied it by 100 to find the final percentage.

Here are the results of our query for 3d:



Note that this is the difference (not absolute) between the hybrid and combustible emissions. Specifically, combustible emissions subtracted from hybrid emissions.The data overwhelmingly suggests that there is a substantial change in CO2 levels when hybrid vehicles are used instead of traditional gasoline. To find this value, we checked when the travel distance of a vehicle was greater than the given vehicle charge distance (20, 40, and 60 miles respectively). If the distance was greater than the charge distance, we removed the charge distance from the trip miles and calculated the remaining miles' CO2 emissions. However, electricity has its own CO2 emissions from production (via coal/gasoline burning, for example), and we had to calculate those emissions as well using the EIA electric carbon dioxide emissions table and the EIA kilowatt-hour table.

We also accomplished loading data from the MATLAB file into the postgres database. We did so by using the h5py library in python. The h5py lib helped us dive into the file and pull all the necessary information we needed to fill the tables. We created the tables and inserted all the values correctly. However, it is a very slow process. Even though the database contains somewhat less info than the csv files, it still takes a very long time to load 1 million tuples into PostgreSQL.