# Project Overview

For this project we are asked to extract, transform, and load (ETL) several datasets using any necessary alternations (cleaning, joining, filtering, aggregating, etc.) to produce a single clean dataset, from which a final production database will be established.

# Extraction

While heavier-duty processes can be taken to scrape the web for data, we found it more practical to simply download our datasets directly as csv files. These files are then placed in a folder named ‘resources’, where they can be easily accessed during the transformation process.

# Data Sources: In order to gain insight into market trends in the automotive sector, we will be utilizing vehicular data taken from the following reputable sources:

1. **Dataset:** Features and MSRP

**Source:** *Kaggle,* <https://www.kaggle.com/CooperUnion/cardataset>

**Description:** Cars dataset with features including make, model, year, engine, and other properties of the car used to predict its price.

**Format:** CSV file

1. **Dataset:** Vehicle Technical Specs

**Source:** *FuelEconomy.gov Web Services,* <https://www.fueleconomy.gov/feg/ws/index.shtml>

**Description:** Dataset including make, model, year, and many other characteristics including EPA ratings.

**Format:** CSV file

1. **Dataset:** Vehicle Emissions

**Source:** *FuelEconomy.gov Web Services,* <https://www.fueleconomy.gov/feg/ws/index.shtml>

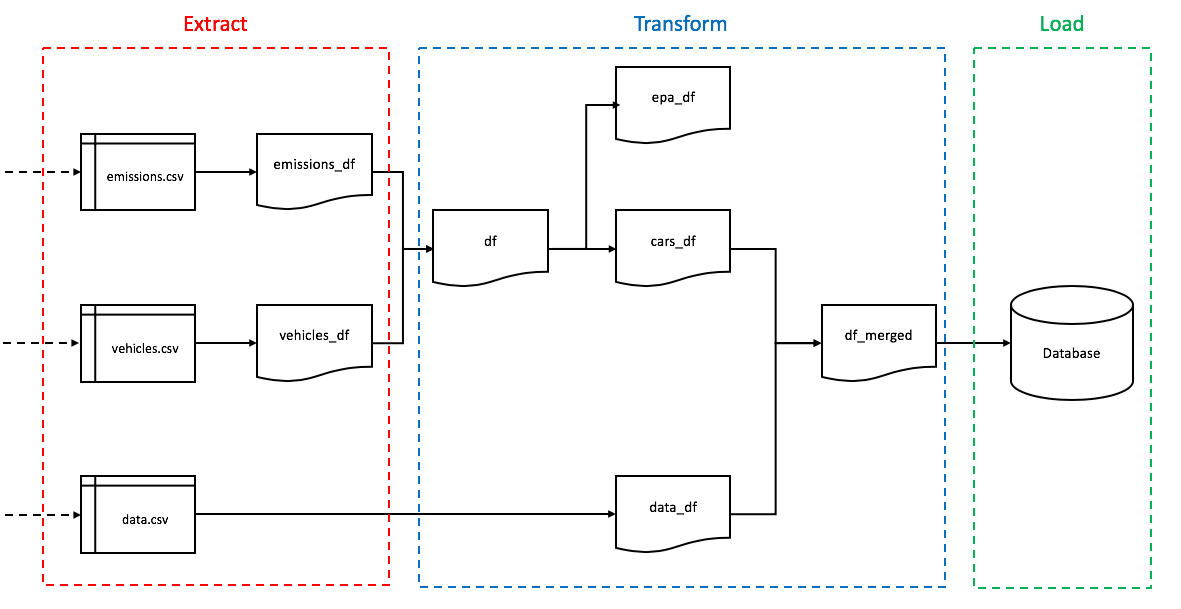
**Description:** Dataset including vehicle record ID, EPA and emissions details.

**Format:** CSV file

Other datasets from the following sites were also considered and reviewed:

1. <https://datasetsearch.research.google.com/>
2. <https://data.world/>
3. <https://www.statista.com/>
4. <https://car2db.com/>
5. <https://www.cars-data.com/>
6. <https://en.wikipedia.org/wiki/List_of_automobile_sales_by_model>

One potential data flow diagram of our ETL process is displayed below:



# Transformation

While our datasets were fairly clean, several steps were taken to clean all datasets, as is demonstrated in the accompanying Jupyter Notebook file *‘pandas\_etl.ipynb’*. Other transformation processes (i.e. joining, filtering, and aggregating) were deemed by us to be easier to performed in SQL following the Load step. An example cleaning and transformation process for the *‘vehicles.csv’* dataset is shown below:

## Vehicles dataset (*vehicles.csv)*

**Step 0. Import dataset**

vehicles\_file = "Resources/vehicles.csv"

vehicles\_df = pd.read\_csv(vehicles\_file)

vehicles\_df.head()

**Step 1. Rename column headers**



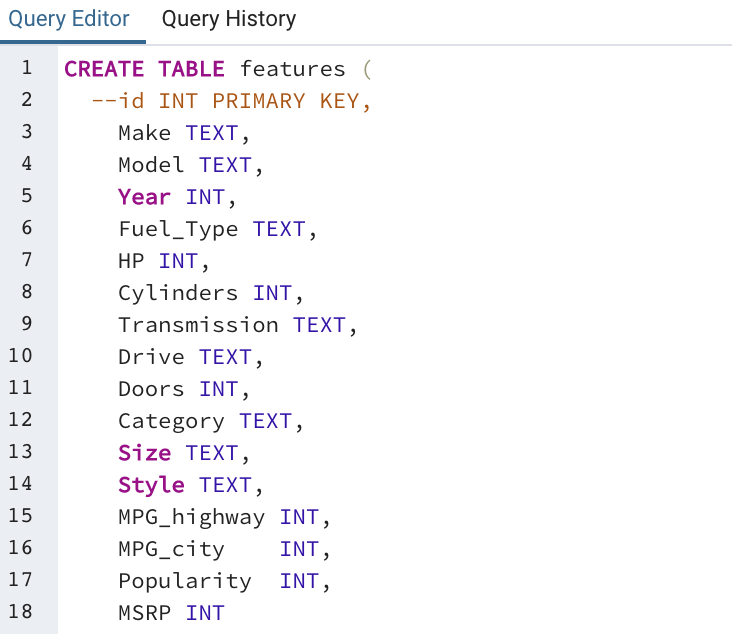
Step 2. Rearrange columns, create copy and set index to column *‘id’*



# Loading

For our purposes we will be using PostgreSQL for our database. PostgreSQL (or Postres for short) is an open-source relational database and is useful when standards compliance and extensibility is important for a given dataset.

We can now create an SQL (relational) db and table structure (ex. *‘features’* table):



**Next we create a db connection to the db and table structure described above**

connection\_string = "postgres:XXXXXXXX@localhost:5432/Cars\_Project2"

engine = create\_engine(f'postgresql://{connection\_string}')

**And load dataframes into the SQL database**



As mentioned above, we considered the transformational processes of joining, filtering, and aggregating to be easier done in SQL. We can join ‘vehicles’ and ‘emissions’ tables on id/vehicle\_record\_id and ‘vehicles’ and ‘features’ tables on make and/or model, in order to obtain needed data. See examples below.

