# Project Overview

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

The ETL process makes up a crucial part of the Analytics Paradigm (Fig. 1) employed by data analysts and is an essential step to creating meaningful and accurate analyses.

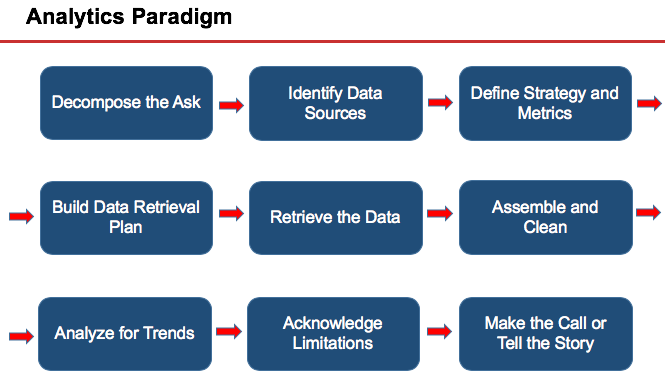


Figure - Analytics Paradigm

# 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. **Car Features and MSRP**,*Kaggle*

<https://www.kaggle.com/CooperUnion/cardataset>

‘resources/data.csv’

1. **Vehicle Records**, *FuelEconomy.gov Web Services*

<https://www.fueleconomy.gov/feg/ws/index.shtml>

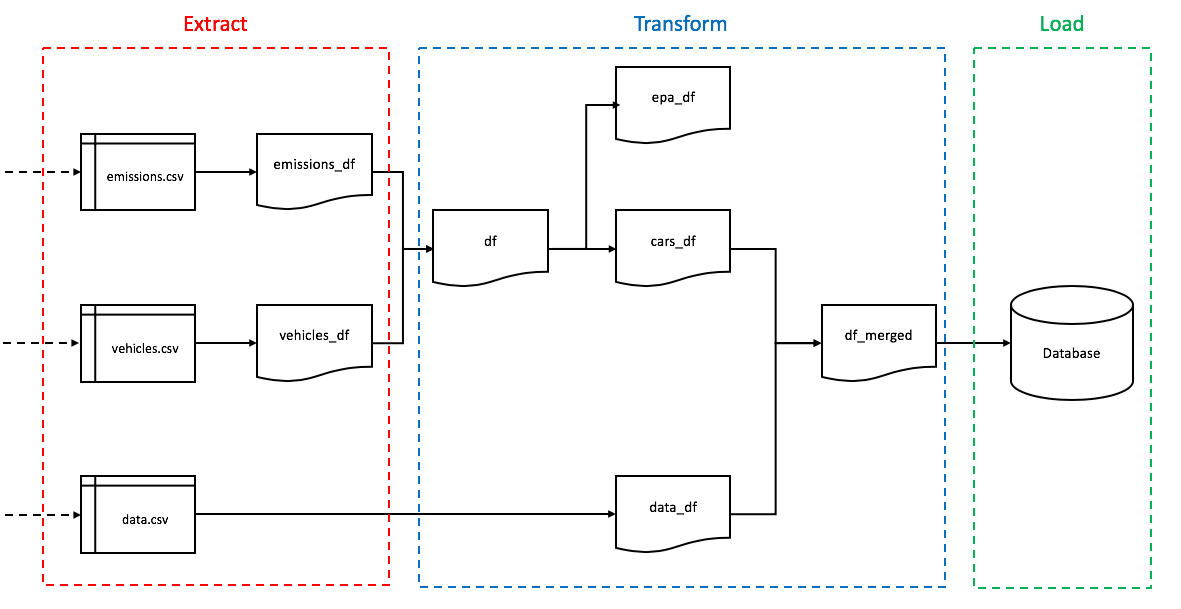
‘resources/vehicles.csv’

1. **Emissions Records**, *FuelEconomy.gov Web Services*

<https://www.fueleconomy.gov/feg/ws/index.shtml>

‘resources/emissions.csv’

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



# Extraction

While heavier-duty processes can be taken to scrape the web for data, we found it much 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.

# Transformation

While our datasets are fairly straightforward and well-maintained, several steps were taken to clean, and then later join, all datasets, as is demonstrated in the accompanying Jupyter Notebook file *‘pandas\_etl.ipynb’*. An example cleaning and transformation process for the *‘vehicles.csv’* dataset is shown below:

## Vehicles dataset (*vehicles.csv)*

Step 0. Import dataset

Step 1. Rename column headers



Step 2. Rearrange columns and create copy

Step 3. Set index to column *‘id’*



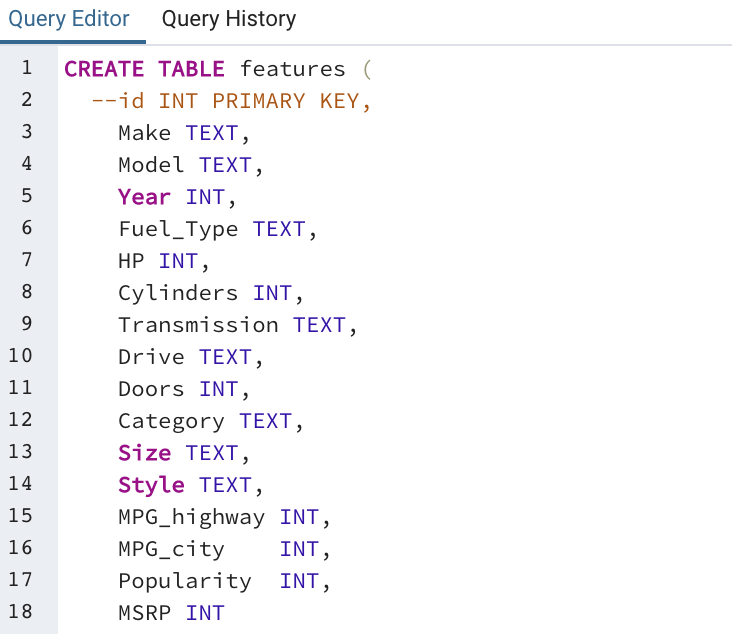
# Loading

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.

Creating a database connection and engine:



We now create a table for *‘features’* in Postgres:



This process was repeated for the ‘vehicles’ and ‘emissions’.

Populating the tables back in Jupyter Notebook:



Finally, we can then join both the ‘vehicles’ and ‘emissions’ tables and then the ‘vehicles’ and ‘features’ tables, as shown below:

