**Extract: your original data sources and how the data was formatted (CSV, JSON, pgAdmin 4, etc)**

The three original data sources are: the PetFinder API, “Most Popular Dog Breeds of 2018 (2019)” article from the *American Kennel Club* website, and the “Most Pet-Friendly Cities” article from the *Wallet Hub* website. The PetFinder API calls return nested dictionaries in JSON format. The “Most Popular Dog Breeds of 2018 (2019)” and “Most Pet-Friendly Cities” articles are formatted in HTML.

**Transform: what data cleaning or transformation was required**

To extract the information from the American Kennel Club article, we utilized the Pandas module. First, we loaded the URL into a Python variable. This variable was called into the “read\_html” Pandas function. This function returned the HTML data into a Python list. We converted the Python list to a Python DataFrame, renamed the column headers, and added a column “Rank Year” to display when the ranking information was pulled. This will allow for year over year comparisons.

To extract the information from the Wallet Hub article, we navigated to the URL in the Chrome browser and used Chrome’s Inspect tool to view the source HTML code. Next, the HTML code was copied and pasted into a HTML file in the Visual Studio Code program. The OS, Beautiful Soup, and CSV Python modules were used to extract the table information. The HMTL file was read in with the OS module and stored in a variable. A Beautiful Soup object was created from this variable, and the Beautiful Soup “find” function was used to locate the table in the file. A function was then used to write all the table row data into a CSV file. This CSV file was read into a Pandas DataFrame.

The Python Requests library was used to connect to the PetFinder API. The API call was filtered to solely extract the dog animal type. The breed of the dog and city where it is located were equally important to pull as these variables will allow us join with the aforementioned HTML data. The other variables that were pulled from the API include the following: name of dog, attriubutes of dog (declawed, etc.), status of dog, city and state of dog, gender, age, and primary/secondary breeds.

We used the Pandas DataFrame “Merge” function to combine the PetFinder API and Top Breeds Pandas DataFrames on “Breed Name.” This new DataFrame was then merged with the “Pet Friendly Cities” DataFrame on “City.” This new DataFrame is our final DataFrame. We exported the DataFrame into a CSV file.

**Load: the final database, tables/collections, and why this was chosen.**

There is one final database named “Adoptable Dogs.” This is a MySQL database. The MySQL database was chosen due to the tabular format of the data. There is one table named “PetFinder\_dogs.” This table was populated using the “Table Data Import Wizard” function in MySQL Workbench.

