1. Demonstrate that all the relations in the relational schema are normalized to Boyce–Codd normal form (BCNF).

- The relations in the relational schema are currently in 2NF, since there are no partial dependencies present within our schema. However, there are transitive dependencies present with the Electric Vehicle relation. The first step to normalize our schema to Boyce-Codd normal form is to remove the transitive dependencies.

    - {Municipality, County, Year} -> {Total vehicles, No of EV's)
    - {Total vehicles, No of EV's} -> {% of EV's}

- Once the transitive dependencies have been removed and decomposed the relational schema is normalized to Boyce-Codd normal form.

2. Define the different views (virtual tables) required. For each view list the data and transaction requirements. Give a few examples of queries, in English, to illustrate.

- The view would have virtual tables of Electric vehicles, Fuel vehicles, Community Profile, and GHG. For Electric Vehicles, the user can view total vehicles, number of Evs, and %s of EVs based on Municipality, County, and Year. For Fuel Vehicles, the data would contain the municipality, county, year, fuel vehicle emissions, means of transportation, and miles traveled. For the Community Profile virtual table, the data displayed would be municipality, county, year, means of transportation, square miles, population, median household income, % of population in poverty. For the GHG virtual table, the data displayed would be municipality, county, year, passenger vehicle emissions, and total MTCO2e. There are no transaction requirements for our views.

3. Design a complete set of SQL queries to satisfy the transaction requirements identified in the previous stages, using the relational schema and views defined in tasks 2 and 3 above.

- For the user to view data from a virtual table they can select the information they would like to view from a specific virtual table based on the name of the municipality, county, or year. For example, the user can view the Total number of Vehicles, No of EV's and % of EV's from the Electric Vehicle relation for a certain municipality.
    - SELECT Total number of Vehicles, No of EV's and % of EV's
    - FROM Electric Vehicle
    - WHERE municipality = " *municipality name*";
- To view data from two different tables, for example if you want to view the Community Profile data and the Electric Vehicle data you can use the following command:
    - SELECT * FROM Community Profile
    - INNER JOIN Electric Vehicle
    - ON Community Profile.Year = Electric Vehicle.Year;

Since there are no transaction requirements, these two samples of queries satisfy the database capabilities.