

Vehicle Database

Warvin Hassan

December 4, 2024

Introduction: Climate change is real, and one of the biggest contributors to that are vehicles. For this paper, it is assumed that the reader has knowledge of databases equivalent to an introductory database student. This paper serves as a follow-up to the final project presentation on my vehicle database, done in class, and effectively describes the process of how I created and used a database to illuminate the threat of CO2 emissions from vehicles. The goal is to identify the main culprit of these emissions using mock data, focusing on the specific types of vehicles and their associated emissions. The database schema was designed to capture relevant details while ensuring scalability for future data integration. The relationships between the various entities, such as person, vehicle, and engine, were modeled using normalized tables. This structure allows for efficient data storage and retrieval, while minimizing redundancy. The mock data used in this paper includes variables such as vehicle make, model, engine type, and emissions, all of which contribute to understanding the environmental footprint of different vehicles. By analyzing these data points, we can identify which vehicle types and engine configurations are the most significant contributors to CO2 emissions.

Implementation: For the data collection phase, I leveraged ChatGPT to generate mock data based on common vehicle categories with the following prompt:

“I want 100 entries of raw data generated in the form of a csv file that mocks real data from 2015 until now. I want the columns to be based off of this schema:
Owner information, such as fullname, and address(street1,street2optional,city,state,postal)
Vehicle information such as year, make, model, plateID and Statecode, engine type and the emissions produced based off of miles driven (for electric I want emissions produced from production fossil fuels since they don’t really produce anything on their own). As for the vehicle types (electric, hybrid, and gas) I want data on the battery capacity and range of electric types, battery capacity and fuel efficiency of hybrids and fuel efficiency of gas types. Also, some optional data meaning they don’t all have to have data on these columns: registrationDate,

salesAmount, salesDate.”

Using the AI’s ability to simulate data from a range of automotive sources, I created a diverse dataset that includes vehicle makes, models, engine types, fuel efficiency ratings, and annual CO2 emissions estimates. This mock data aimed to represent various vehicle types, such as gas, electric vehicles, and hybrids, ensuring a wide coverage for the analysis. While the data generated is not representative of real-world figures, it was more so designed to almost accurately reflect the type of information typically found in publicly available automotive databases and governmental reports on emissions. The mock data provided a foundation for building out the database structure and allowed for meaningful queries and analysis of the relationship between vehicle characteristics and emissions outputs. For first normal form, I ensured that all attributes in the database contained atomic values, meaning each field only holds a single value. For example, the vehicle table contains distinct columns for make, model, and year, avoiding multiple values in a single column. For second normal form, I ensured that all non-key attributes in the database were fully functionally dependent on the primary key. For instance, the engine table stores CO2 emissions data that depends directly on the engineID.

Results: This database was queried to generate insights into which vehicle types (electric, hybrid, gas) are most responsible for CO2 emissions. The main objective of this query was to identify vehicles with above-average CO2 emissions, in order to analyze the types of vehicles and engines contributing most to the environmental impact. The query selects relevant vehicle details, including the make, model, state, engine type, and annual CO2 emissions. It joins several tables—vehicle, engine, personvehicle, person, and address—to retrieve the necessary data and applies a sub-query that filters for vehicles whose emissions exceed the average emissions for all engine types. More specially, the columns make and model provide the vehicle’s make and model, helping identify specific types of vehicles contributing to high emissions. State shows the residence for the vehicle owner, offering geographical context. engine_type specifies the engine type for each vehicle (e.g., gas, hybrid, electric), which plays

a critical role in determining which vehicle type is contributing to high emissions. Finally, `annual.CO2_kg` presents the annual CO2 emissions in kilograms, allowing for direct comparison across vehicles. The query returns a list of vehicles that have annual CO2 emissions higher than the average emissions for their respective engine types. The results are sorted in descending order of emissions, allowing for a clear view of which vehicles contribute most to CO2 output. The results clearly show that traditional gasoline-powered vehicles such as the Toyota RAV4 and Toyota Prius emit the most CO2 compared to their electric and hybrid counterparts, which produce significantly lower emissions.

Table 1:

Make	Model	State	Engine_Type	Annual_CO2_kg
Toyota	RAV4	TX	gas	6981.95
Toyota	Prius	OR	gas	5949.63
Honda	Insight	NM	hybrid	2820.43
Honda	CR-V	NY	hybrid	2530.89

The resulted table of 4 rows to answer my question using the query in figure 1. Based off of the data in this table Toyota is the main contributor to annual CO2.

```
1 • SELECT make Make, model Model, state State,  
2     type Engine_Type, emissions Annual_CO2_kg  
3     FROM vehicle AS v  
4     JOIN engine AS e ON v.engineID = e.engineID  
5     JOIN personvehicle AS pv ON v.vehicleID = pv.vehicleID  
6     JOIN person AS p ON pv.personID = p.personID  
7     JOIN address AS a ON p.personID = a.personID  
8     WHERE emissions > (  
9         SELECT AVG(emissions)  
10        FROM engine)  
11     ORDER BY emissions DESC;
```

Figure 1: The query to table 1 that joins respective tables and compares emissions to average emissions.

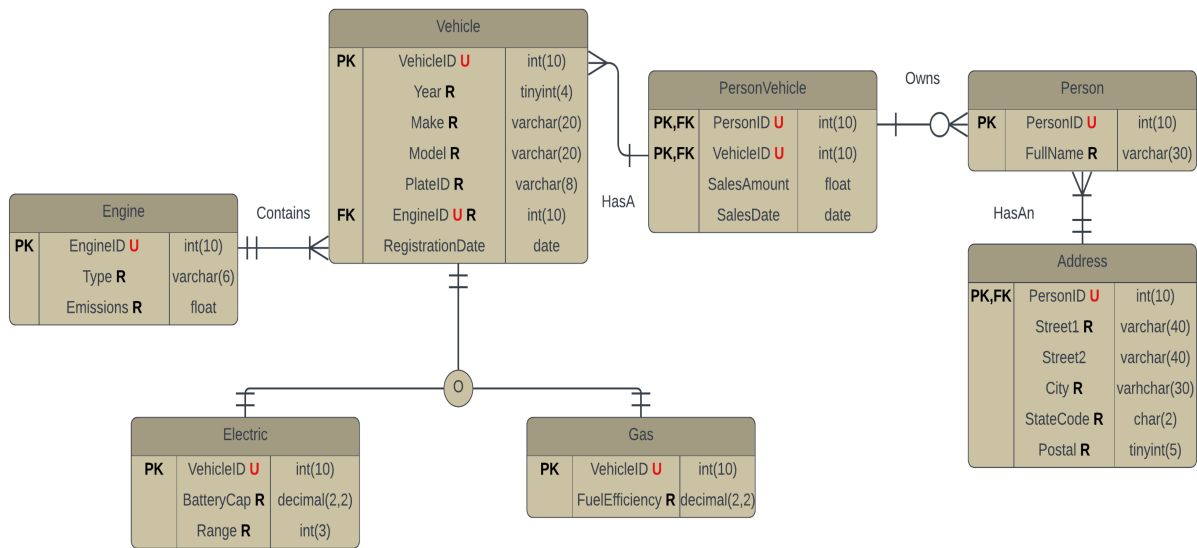


Figure 2: The vehicle table diagram used for the creation of my schema.

```
CREATE TABLE `address` (  
  `personID` smallint unsigned NOT NULL,  
  `street1` varchar(40) NOT NULL,  
  `street2` varchar(40) DEFAULT NULL,  
  `city` varchar(30) NOT NULL,  
  `state` char(2) NOT NULL,  
  `postal` int NOT NULL,  
  PRIMARY KEY (`personID`),  
  CONSTRAINT `personID` FOREIGN KEY (`personID`) REFERENCES `person` (`personID`)  
    ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

Figure 3: Query used to create address table.


```
CREATE TABLE `electric` (  
  `vehicleID` smallint unsigned NOT NULL,  
  `batteryCap` float NOT NULL,  
  `range` smallint NOT NULL,  
  PRIMARY KEY (`vehicleID`),  
  CONSTRAINT `electricID` FOREIGN KEY (`vehicleID`) REFERENCES `vehicle` (`vehicleID`)  
    ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

Figure 4: Query used to create electric subtable.

```
CREATE TABLE `gas` (  
  `vehicleID` smallint unsigned NOT NULL,  
  `fuelEfficiency` float NOT NULL,  
  PRIMARY KEY (`vehicleID`),  
  CONSTRAINT `gasID` FOREIGN KEY (`vehicleID`) REFERENCES `vehicle` (`vehicleID`)  
    ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

Figure 5: Query used to create gas subtable.

```

CREATE TABLE `vehicle` (
  `vehicleID` smallint unsigned NOT NULL AUTO_INCREMENT,
  `year` smallint unsigned NOT NULL,
  `make` varchar(20) NOT NULL,
  `model` varchar(20) NOT NULL,
  `plateID` varchar(8) NOT NULL,
  `engineID` smallint unsigned NOT NULL,
  `registration` date DEFAULT NULL,
  PRIMARY KEY (`vehicleID`),
  UNIQUE KEY `statePlate_idx` (`plateID`),
  KEY `engineID_idx` (`engineID`),
  CONSTRAINT `engineID` FOREIGN KEY (`engineID`) REFERENCES `engine` (`engineID`)
  ON DELETE RESTRICT ON UPDATE CASCADE
);

```

Figure 6: Query used to create vehicle table.

```
CREATE TABLE `engine` (  
  `engineID` smallint unsigned NOT NULL AUTO_INCREMENT,  
  `type` varchar(10) NOT NULL,  
  `emissions` float NOT NULL,  
  PRIMARY KEY (`engineID`)  
);
```

Figure 7: Query used to create engine table.

```
CREATE TABLE `personvehicle` (  
  `personID` smallint unsigned NOT NULL,  
  `vehicleID` smallint unsigned NOT NULL,  
  `saleAmount` float DEFAULT NULL,  
  `saleDate` date DEFAULT NULL,  
  PRIMARY KEY (`personID`,`vehicleID`),  
  KEY `vID_idx` (`vehicleID`),  
  CONSTRAINT `pID` FOREIGN KEY (`personID`) REFERENCES `person` (`personID`)  
    ON DELETE RESTRICT ON UPDATE CASCADE,  
  CONSTRAINT `vID` FOREIGN KEY (`vehicleID`) REFERENCES `vehicle` (`vehicleID`)  
    ON DELETE RESTRICT ON UPDATE CASCADE  
);
```

Figure 8: Query used to create personvehicle table.

```
CREATE TABLE `person` (  
  `personID` smallint unsigned NOT NULL AUTO_INCREMENT,  
  `fullName` varchar(40) NOT NULL,  
  PRIMARY KEY (`personID`)  
);
```

Figure 9: Query used to create person table.