

CSE 111

Automobile

Database

By: Omar Ramos-Correa and Lucas Ortiz-Gonzalez

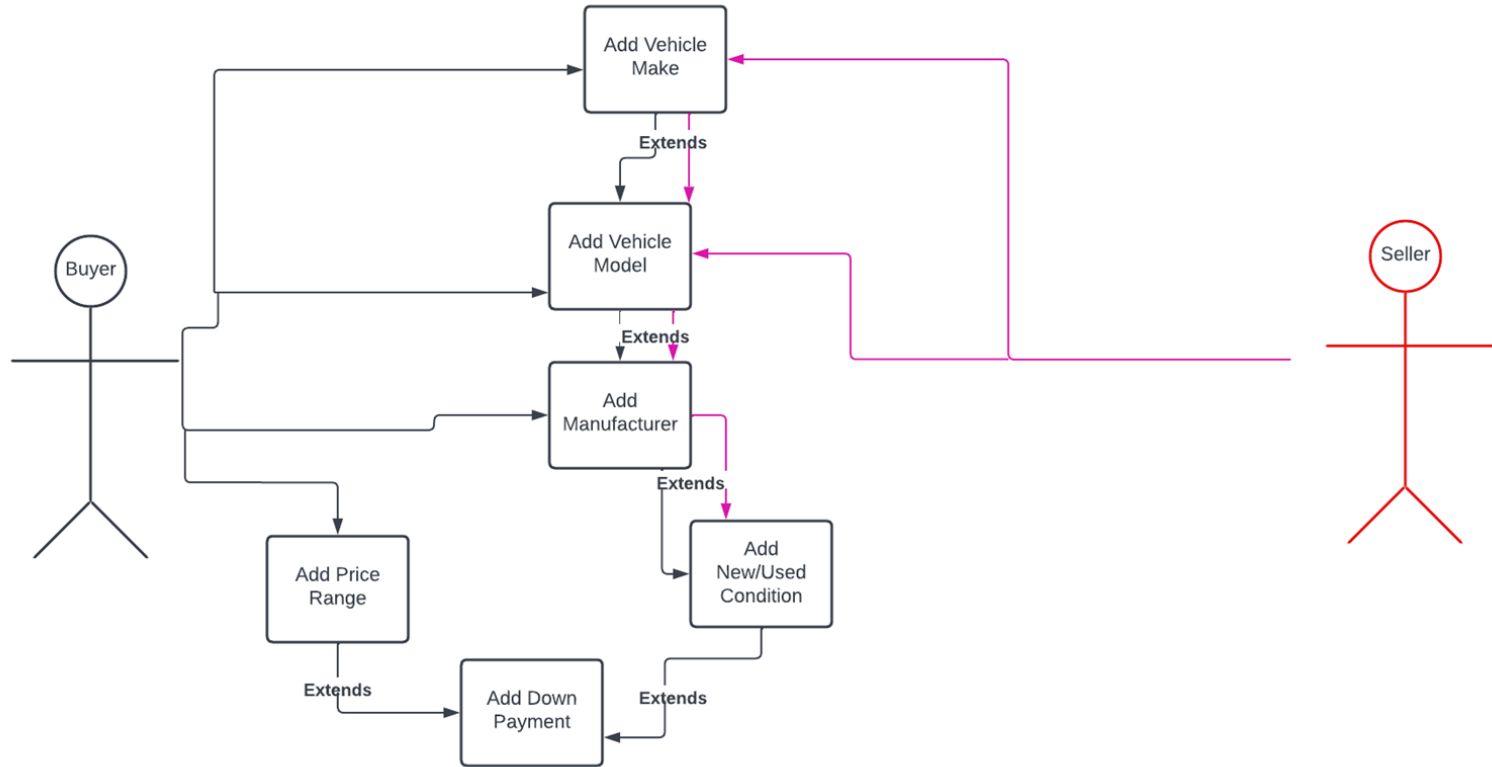


Description

We are going to implement an Automobile Database where a customer can view all the automobiles in said database and make a purchase on the one they like. Sellers will upload automobiles they would like to sell on the database.

- Seller adds vehicle into database
- Buyer can view all vehicles depending on their specifications
- Certain attributes cannot be modified after being created

UML Diagram





Tables

Customer [Person buying the car]

Seller [Person selling the car]

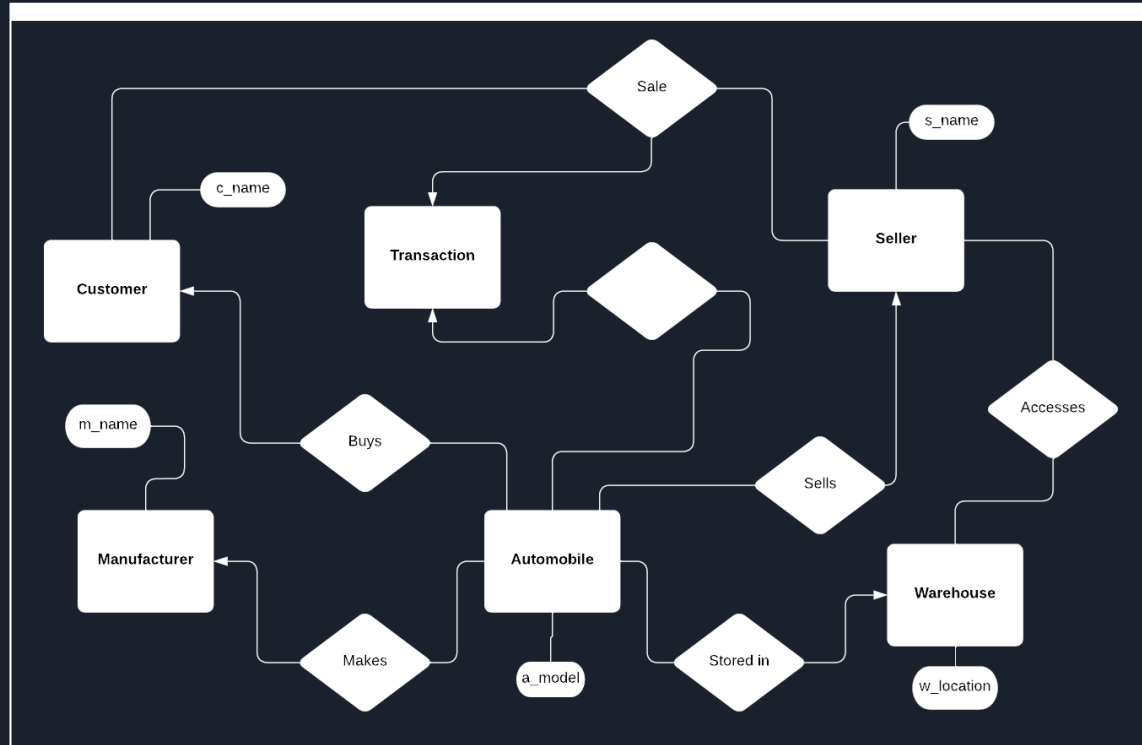
Manufacturer [Company who made the car]

Transaction [Action of car being sold]

Automobile [The car being bought]

Warehouse [Location where car is stored]

E/R Diagram



Schema

Automobile

- 🔑 a_VIN : integer
- ◆ a_make : varchar
- ◆ a_model : varchar
- ◆ a_type : varchar
- ◆ a_year : integer
- ◆ a_condition : varchar
- ◆ a_color : varchar
- ◆ a_price : integer

Customer

- 🔑 c_custkey : integer
- ◇ c_VIN : integer
- ◆ c_lastname : varchar
- ◆ c_firstname : varchar
- ◆ c_phone : string
- ◆ c_city : varchar
- ◆ c_state : varchar
- ◆ c_sellername : varchar

Manufacturer

- ◆ m_mftrkey : integer
- ◆ m_make : varchar
- ◆ m_model : varchar
- ◆ m_type : varchar

Seller

- 🔑 s_sellerkey : integer
- ◆ s_name : varchar
- ◆ s_phone : string
- ◆ s_city : varchar
- ◆ s_state : varchar
- ◆ s_email : varchar

Transactions

- 🔑 t_trkey : integer
- ◆ t_VIN : integer
- ◆ t_custkey : integer
- ◆ t_sellername : varchar
- ◆ t_price : integer
- ◆ t_date : date

Warehouse

- ◆ w_VIN : integer
- ◆ w_sellerkey : integer



Implementation Details

For the project, we used our own database which we named “cars.db” and connected our sqlite queries to a python code via an ubuntu terminal.

We used :

- Python

- Sqlite3

- Visual Studio Code



Github

https://github.com/LOrtizGonzalez/CSE111_FinalProject