Introduction to Data Management and Engineering

PROJECT 2 – E-R Diagrams and Database Programming with Python

Lampros Floudas

f3352126

Q1) Firstly, to design the E-R diagram for the car dealership project, the tables consisting the database needed to be decided.

- ➢ The first table to be drawn, is the **Brands** table, containing all the Brand names of the cars available as well as brand ids as a primary key.
- ➢ The second table to be drawn is the **Models** table. This table contains the model's name using a model ID as a primary key and the brand ID as a foreign key connecting each model to a particular brand.
- ➢ Thirdly, another table drawn is the **Features** table. It contains all available features for each car.
- ➢ As a fourth table, a Dealers table was computed consisting of the Dealer's name, the dealer's ID as a primary key and the region each dealer is in.
- ➢ The fifth and main table to be drawn is the **Vehicles** table. This table contains the VIN as a primary key, the model ID connecting it to the model (and in turn with the brand), the options ID as a connection to the features table and lastly the dealer ID as a foreign key connecting each car to the dealer who has it available.
- ➢ After that, a **Customers** table was created giving each customer a customer ID as a primary key, each customer's Frist name, Last name, Region, email and photo.
- ➢ Lastly, the **Orders** table was computed, consisting of the Order ID as a primary key, and order Date and a customer ID as a foreign key referencing the customers table as well as a vin connecting the order to the car ordered.
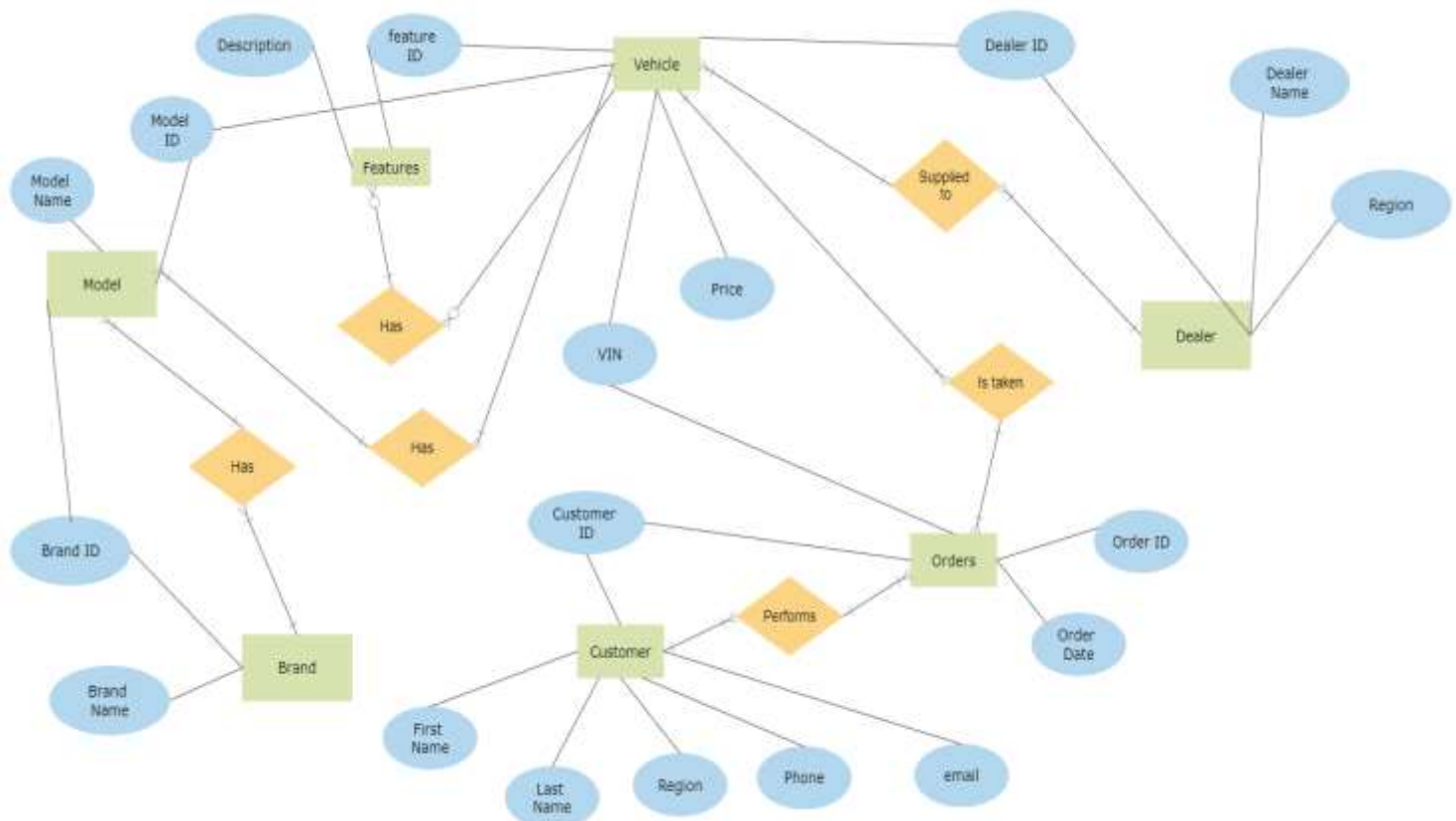

The relations between each table is shown on the E-R Diagram drawn above.

Q2) After deciding on the tables to be used, they needed to be created. Using the CREATE TABLE command, all tables were created on PostgreSQL. For each table, the IDs were created using the command Serial which works as the "AUTO INCREMENT" command, automatically assigning values for the primary key, making sure they are all different to each other. Creating the tables, it was made sure that the phone numbers of each client was a string since a phone number would be a very large integer and there was no point in having it as an integer.

After creating the tables of the data base, they needed to be filled with data. Using INSERT INTO table VALUES (…, …, … ,…) the tables were filled with data. Real brands and models were used for cars while fake names, emails and phones were used for data for the customers and dealers.

The PostgreSQL E-R generated can be seen above.
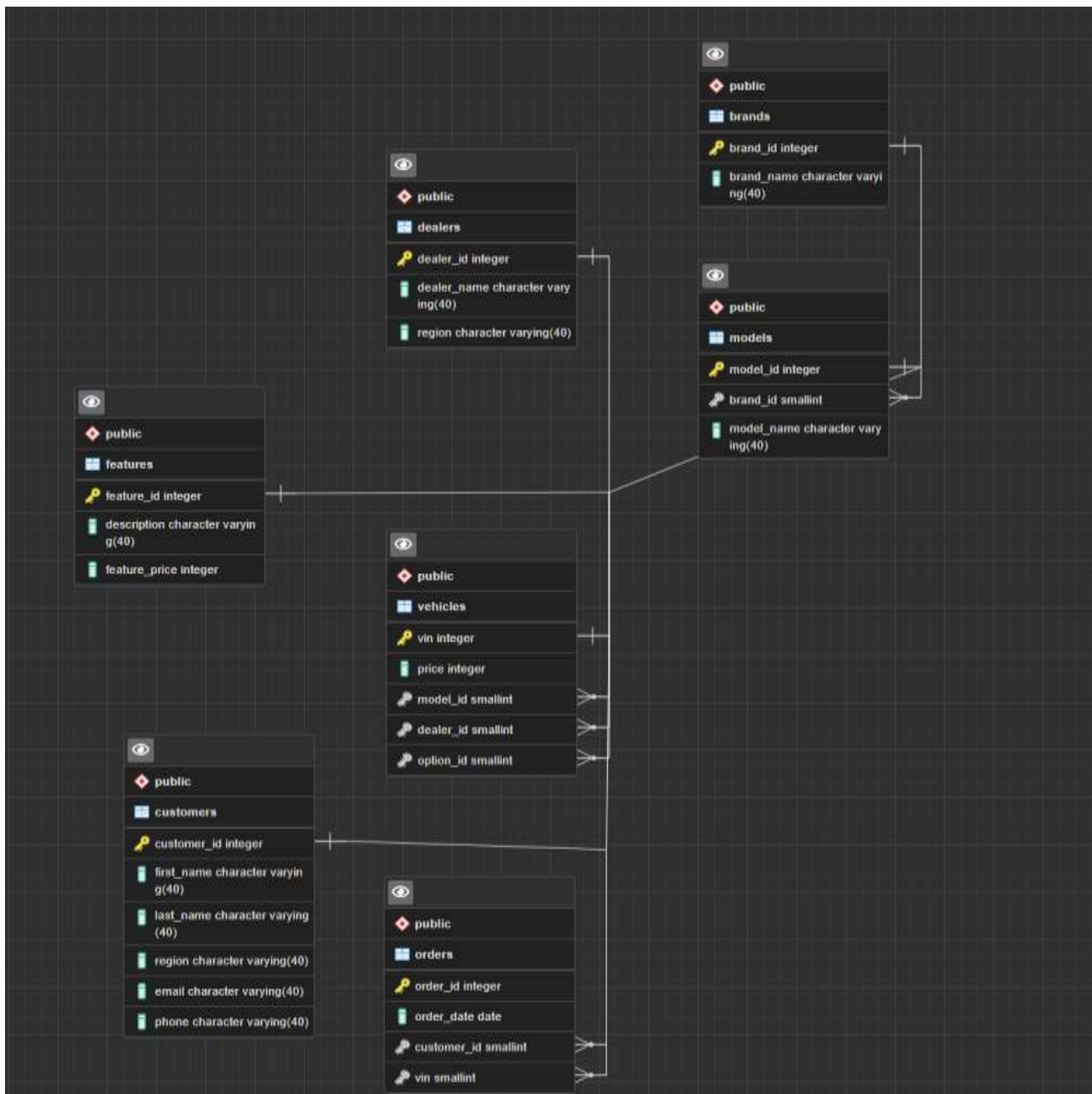
Q3) The indices created for this database were the price of the vehicles, the last and first name of the customers, the brands and the regions of each dealer. The indices were created while taking in mind all the queries that would be used on python where the searches are made without using each table's primary key.

CREATE INDEX priceidx ON vehicles (price);

CREATE INDEX lastnameidx ON customers (last_name);

CREATE INDEX firstnameidx ON customers (first_name);

CREATE INDEX brandidx ON brands (brand_name);

CREATE INDEX regionidx ON dealers (region);

Q4) The last part of the project was done in python. The libraries used in order to establish a connection were sqlalchemy and psycopg2. A simple menu was created where it prompts the user for a choice. The choices given to the user are:

1. Entity Management
2. Entity Search
3. Specific Query

After that, for each choices more choices are given to the user. For that, functions were created with specific commands with connection to the PostgreSQL database.

As far as the Entity Management goes, the user is given the choice of inserting a new customer to the customers table, inserting a new order to the orders table, Deleting a specific customer, deleting an order and updating the price of a vehicle. For inserting a new order, the datetime module was used in order to timestamp the order automatically through python.

For the Entity Search, the user has the choice of searching for each customer's activity, searching for available vehicles and searching for vehicle's orders.

For Specific Queries, the user can either search for a car with up to a certain price and search for dealerships from regions.