Principles of Data Management

Insert Group Name Here

August 2018 - December 2018

Phase 2

Our application software will be in the form of a command line interface that provides a tree of commands to the user. The intended user of this software will be administrators who can access information related to customers, dealers, vehicles, and the relationships between them. This software can also function as an abstraction of the database to developers who need access to a higher level API that can provide information about the relationships between customers, dealers, and vehicles.

Progress Report

Progress of the project during this phase went very smooth overall getting all needed attributes of the phase completed and looked over in a timely manner. Starting first with an agreement on how to do the UI in our case we agree on a terminal based UI, afterwards doing the description of the application and modifying the design. These modifications include splitting brand into its own entity with Country, Reliability and Name attributes that has two has relations with Dealer and Vehicle both being many to many relations. Dealer got more attributes replacing Brands with Name and Phone attributes, Sale got its Amount attribute, removed. Vehicle got its Brand attribute removed and replaced with Model and got a new attribute Mileage. Finally, Customer got its Address attribute expanded upon with Street containing Street Number and Street Name and, the inclusion of State and Zipcode under Address. These changes cause a little bit of trouble for the group on the exact specifics of how everything all works together although nothing major. After the design changes work was started on the actual SQL code and getting the database started. The database will use additional data that expands upon the data found in phase 1, in a large quantity of data and updated with the newly add and updated attributes.

ER and UML Diagrams

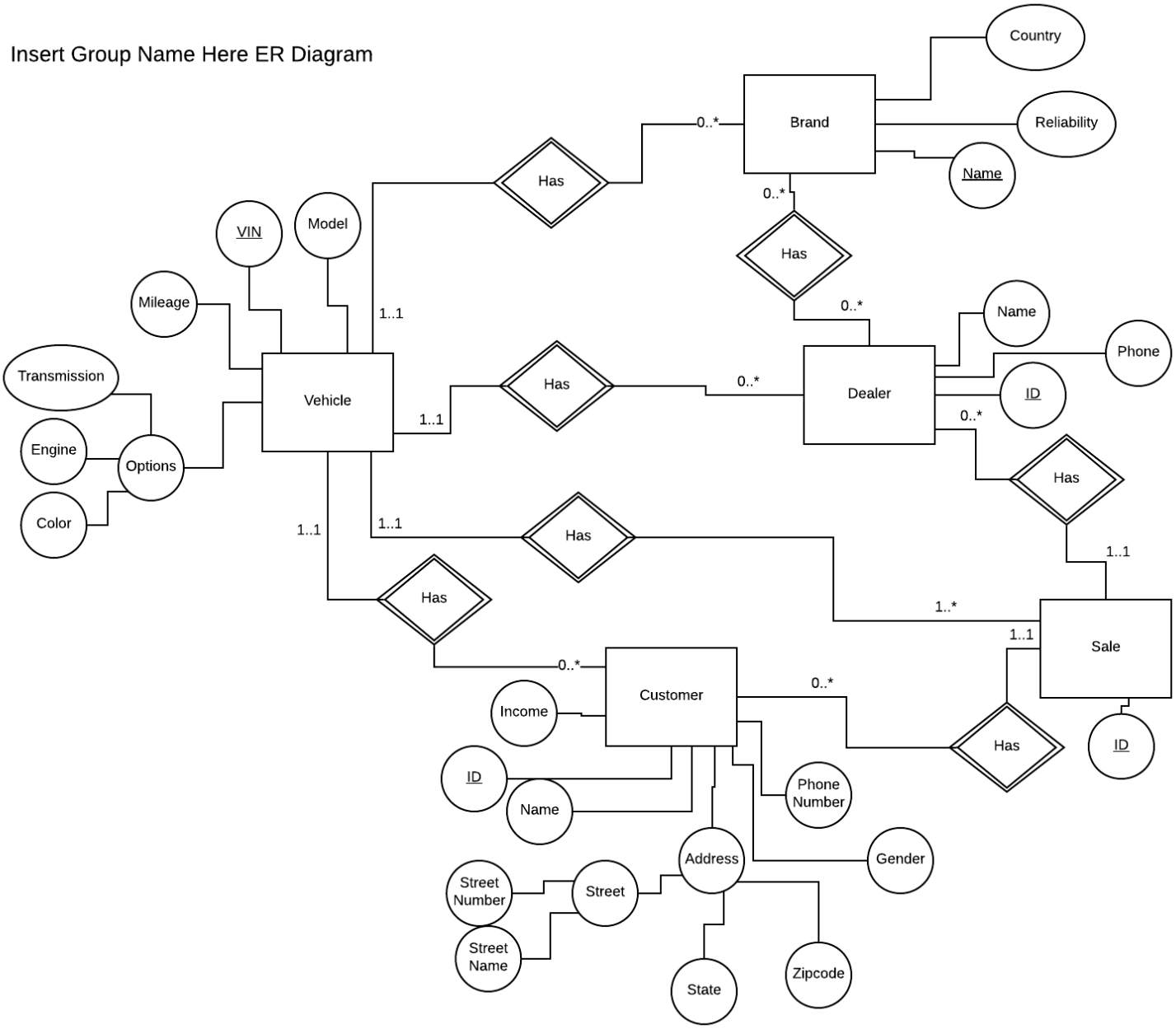
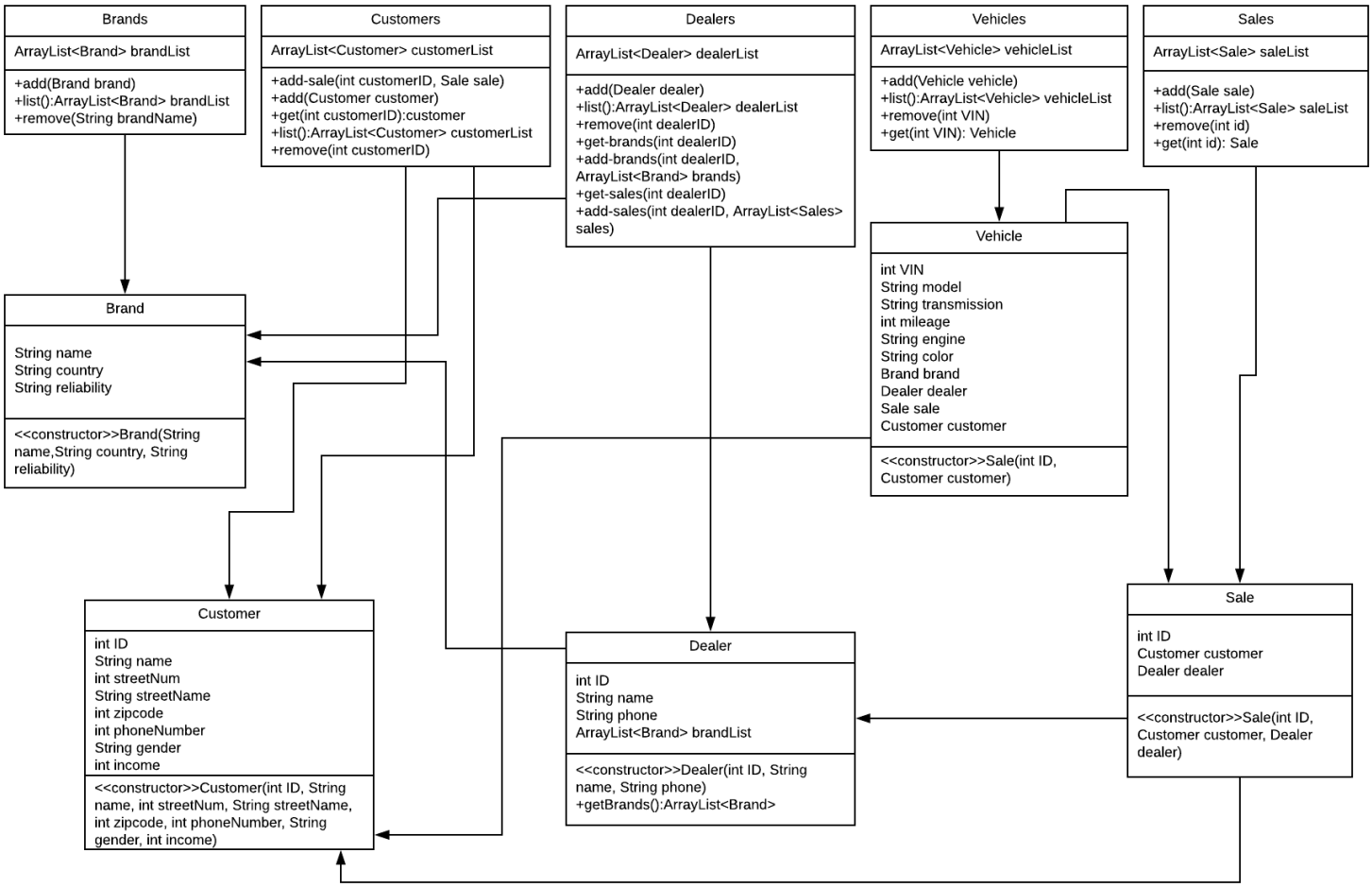


Figure 1: ER diagram

Insert Group Name Here Application UML Diagram



Application Software

The user interface will be a command line interface using a tree of commands that allow access to customers, dealers, vehicles, and information related to the relationships between them.

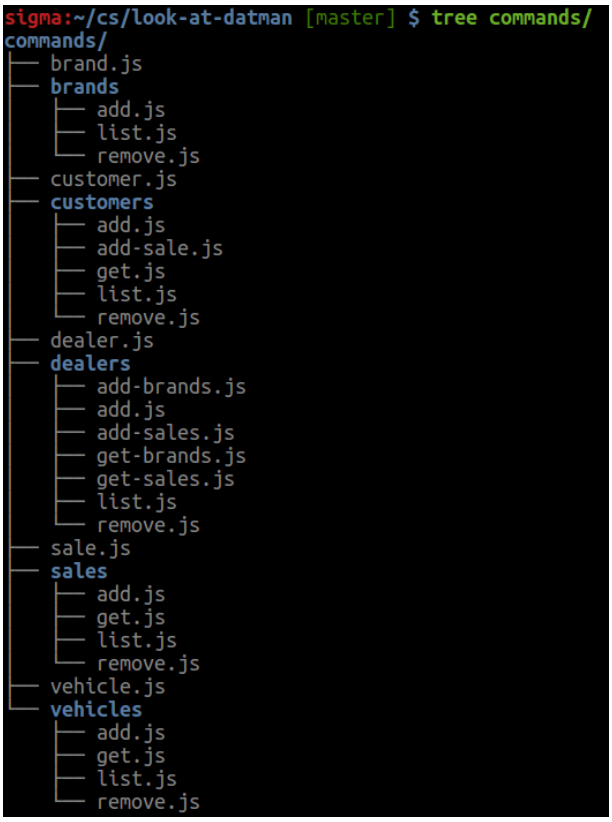


Figure 2: Minimal command tree of available actions

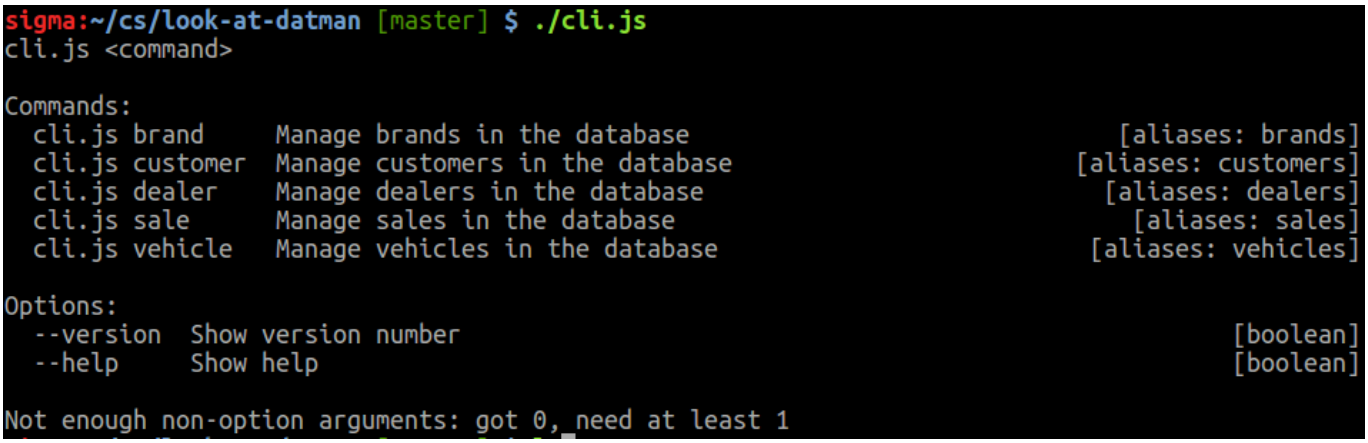


Figure 3: CLI help prompt

The CLI will internally handle formatting and assembling more complicated queries before executing it on the database. It will directly query the database with relevant queries to fetch data for display. The user-facing input side of the software (argument capturing, command line flags, command tree) is handled by the `yargs.js` package while the table display formatting is handled by the `cli-table3` package (both available on npmjs).

Use Cases

This application's intended users are data administrators and developers who will use this as an API. Database functionality will be abstracted away into our application so that users can provide simple queries and commands to perform common actions such as executing a sale, searching up dealers, or registering users. This application will likely be used as a backbone in external point-of-sale software or invoked directly from the command line by an administrator.

Sample SQL

Creating the tables:

```
create table if not exists customer (
  id int primary key,
  income int,
  name varchar(255),
  phone varchar(255),
  gender varchar(16),
  address_street varchar(255),
  address_state varchar(255),
  address_zipcode varchar(255)
);

create table if not exists dealer (
  id int primary key,
  name varchar(255),
  phone varchar(255)
);

create table if not exists brand (
  name varchar(255) primary key,
  country varchar(255),
  reliability varchar(255)
);

create table if not exists sale (
  id int primary key,
  customer int references customer(id),
  dealer int references dealer(id)
);

create table if not exists vehicle (
  vin int primary key,
  model varchar(255),
  transmission varchar(255),
  mileage int,
  engine varchar(255),
  color varchar(255),
  brand varchar(255) references brand(name),
  dealer int references dealer(id),
  sale int references sale(id),
  customer int references customer(id)
);

create table if not exists brand_dealer (
  brand varchar(255) references brand(name),
  dealer int references dealer(id)
);
```

Fetching all vehicles purchased by a customers in NY:

```
select model,brand from vehicle where customer in
  (select id from customer where address_state = "NY");
```

Set a brand's reliability:

```
update brand set reliability="poor" where name="Ford";
```

Add a new vehicle for a dealer:

```
insert into vehicle values(
  1243, "Accord", "Automatic", 10000, "V6", "Ugly", "Honda", 3, NULL, NULL)
```

User Interface

If a query is executed via the command line, the data returned will be nicely formatted into a terminal compatible table for display.

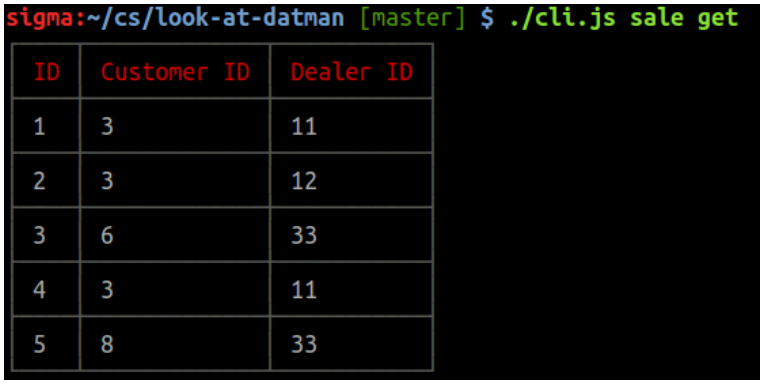


Figure 4: Formatted table in terminal

Configuration data and query parameters will be specified by command line arguments or flags for relevant commands.

Sample Data

Included in the project zip as CSVs.

Attributes Domain

Vehicle:

- Vin: A unique identification code consisting of a combination of 17 numbers and capital letters. The domain is any combination of numbers and letters totaling 17 characters. Is the primary key for Vehicles.
- Options: Consists of three other attributes: Engine, Color and Transmission. This consists of any three combinations of the previous attributes.
- Engine: Type of engine that the car has, this domain is any kind of engine that is used in cars i.e. (VEE, INLINE, STRAIGHT, etc).
- Color: This is the color of the car, as stated it is any word that accurately describes the color of the car (black, grey, blue, etc).
- Transmission: Type of transmission the car has, this domain is any type of transmission a car may have (Automatic, Manual, Dual-Clutch, etc).
- Model: The exact model of the car, can be any valid car model which is a collection of characters, ie (Camry, Highlander, Enclave, etc).
- Mileage: The mileage that the car has traveled, will be an integer that is usually 4 to 7 digits.

Dealer:

- ID: Unique identifier for the dealer will be a unique combination of letters and numbers. Is the primary key for Dealers.
- Name: Name of the dealer can be any combination of characters.

- Phone: Phone number for the dealer can be any valid phone number, meaning a three-digit area code and a seven-digit number.

Brand:

- Country: Country of origin for the brand, will be the name of any country which can be any combination of characters.
- Reliability: Adjectives describing the reliability of the brand, domain will be (Great, Fair, Poor).
- Name: The name of the brand itself. It can be any combination of letters and will be used as the primary key for Brand.

Sale:

- ID: Unique identifier for the sale will be a unique combination of letters and numbers and will be used as the primary key of Sales.

Customer:

- ID: Unique identifier for the customer, will be a unique combination of letters and numbers and is the primary key of Customer.
- Name: Will be the full name of the customer, which can be any combination of letters.
- Address: Address of the customer, will be any valid address string consisting of the street address, state and zip code.
- Street: The street address of the customer this includes the street name and the number.
  - Street Number: The street number will be the street number of the customer, this will be a combination of numbers.
  - Street Name: The name of the street the customer lives on, will be a combination of letters containing its suffix.
  - Zip code: Represents the zip code the customer lives in, will be six-digits long.
- Phone Number: This is the customers given phone number, domain will be any valid phone number, meaning that it's a 10 digit number consisting of the 3 digit area code and their number.
- Gender: Will be the gender of the customer, it will be a single letter representing the gender.
- Income: The customer's annual income, typically a number between 5-8 digits.

You can find all my notes at `http://omgimanerd.tech/notes`. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech