**Final Report**

**Intro to**

**Database Management Systems**

May 5th, 2019

**Team Members**

Brandon Volesky

Colby Chandler

Liam Floyd

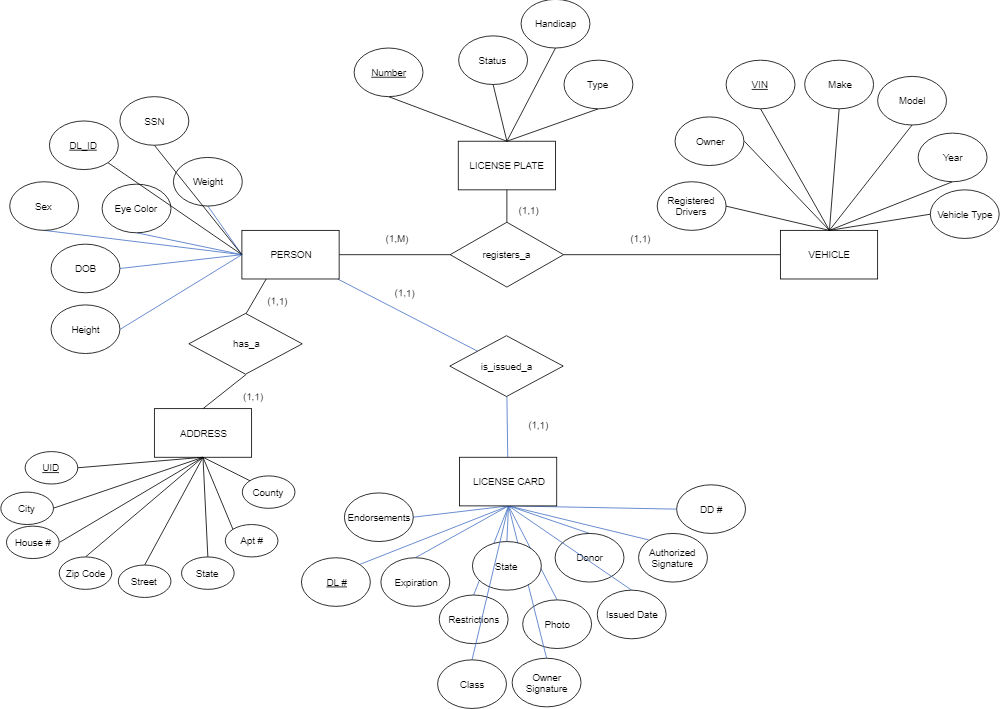
Elizabeth Koch

Mao Zheng

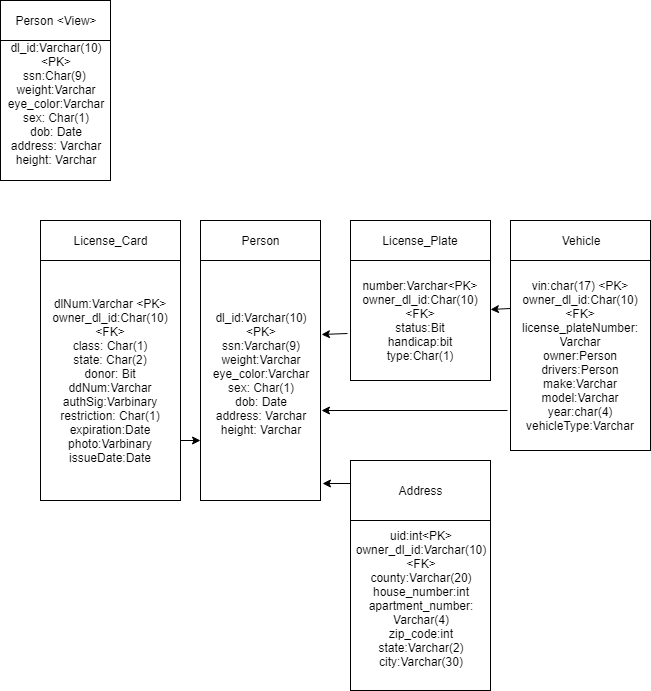
**Summary**

For our project, we decided to create a database for organizing queries that would be made by employees of the Department of Motor Vehicles (DMV). Typical use would include registering new cars, getting regular or custom license plates, and obtaining a driver’s license. Although we hope the database will primarily be used by the DMW, the database could also be used by law enforcement officers. It is a great resource for apprehending criminals, for example it can search for particular make/models with part of their license plate number. Alone, part of the number may not help because many cars have that pattern in their plate, but the join with make/model allows that to narrow it down a lot. That query could also return the owner’s name and address, so one query can return a lot of information.

**ER Diagram**



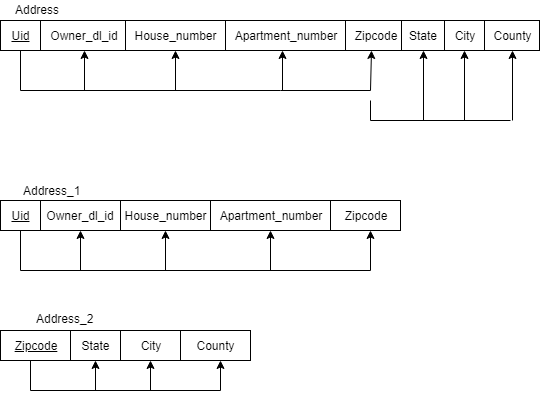
**Schema**



For the view, we figured the DMV employee would be using the owner’s driver’s license to access their account, so we set that as our view. This gives some security as the license has a photo so the employee can verify that they are accessing the correct account. The driver’s license ID can be used then to access everything else in the database. This way the owner SSN isn’t being used as a foreign key.

**Normalization**

The address was not in 3rd normal form because zipcode can uniquely identify the city, state and county. The normalization resulted in the following design

dnl

**DDL**

The SQL statements to create the tables are below

create table person (

dl\_id varchar(10) not null,

first varchar(15) not null,

last varchar(15) not null,

ssn varchar(9) not null,

sex char not null,

eye\_color varchar(3) null,

date\_of\_birth date not null,

height varchar(6) null,

weight varchar(3) null,

primary key(ssn)

);

create table license\_plate (

number varchar(6) not null,

type char(1) null,

handicap bit not null,

status bit not null,

owner\_dl\_id varchar(10) not null,

vehicle\_vin varchar(10) not null,

primary key(number)

);

create table license\_card (

endorsements varchar(20) null,

dl\_number varchar(10) not null,

expiration date not null,

restrictions char null,

state varchar(2) not null,

class char not null,

photo varbinary(50) not null,

owners\_signature varchar(6) not null,

donor bit null,

issued\_date date not null,

authorized\_signature varbinary(50) not null,

dd\_number int not null,

owner\_dl\_id varchar(10) not null,

primary key(dl\_number)

);

create table vehicle (

registered\_drivers varchar(6) not null,

owner\_dl\_id varchar(10) not null,

vin varchar(17) not null,

make varchar(20) not null,

model varchar(20) not null,

year varchar(4) not null,

vehicle\_type varchar(20) not null,

license\_plate varchar(6) not null,

primary key(vin)

);

create table address1 (

uid int not null,

owner\_dl\_id varchar(10) not null,

county varchar(20) not null,

house\_number int not null,

apartment\_number varchar(4) null,

zip\_code int not null,

primary key(uid)

);

Create table address2 (

Zip\_code int not null,

state varchar(2) not null,

street varchar(20) not null,

city varchar(30) not null,

primary key(zip\_code)

);

**Queries**

Here is an example of some queries that we came up with

/\*find all owners who have more than 1 car\*/

select last, count(vehicle.vin) as Num\_of\_cars

from person

INNER JOIN

vehicle

on person.dl\_id=vehicle.owner\_dl\_id

group by last

HAVING COUNT(vehicle.vin)>1;

/\*find everyone who's license starts with 1\*/

select first, last

from person

INNER JOIN

license\_plate

on person.dl\_id=license\_plate.owner\_dl\_id

where license\_plate.number LIKE '1%';

/\*find the city of drivers of an SUV with license starting 12\*/

select city

from address

INNER JOIN

person

on address.owner\_dl\_id=person.dl\_id

INNER JOIN

vehicle

on vehicle.owner\_dl\_id=person.dl\_id

INNER JOIN

license\_plate

on license\_plate.owner\_dl\_id=vehicle.owner\_dl\_id

where vehicle.vehicle\_type='SUV' and license\_plate.number LIKE '12%';

**Future Work**

One thing we didn’t get about to implementing was a view for registering a new driver who just turned 16 or one who has not previously obtained a license. Currently the view requires a driver’s license number to access that driver’s data. This new view would be used by the employees to register a new user after they pass the driving test, or for those who pass the paper exam to obtain a driver’s permit at 15. Driver’s permits are not addressed at all in this database, so that is another thing that could be added. Additionally, there could be some distinction between licenses for those over and under 21, for example below 21 are vertical and above 21 are horizontal to make it easier to distinguish between the two.

Another thing that could be implemented in the future is the possibility for someone to have multiple licenses, for example if someone needs a motorcycle and a commercial driver’s license. This would change the schema quite a bit as driver’s license number is used as the foreign key throughout the database.