CMPUT 291 Mini Project 1 Report

By Yi Ding, Boyuan Dong

The general overview of the system

The project demonstrating functionalities that result from combining SQL with Python and build a system that keeps the enterprise data in a database and to provide services to users.

- Running command: python3 Mini-Project1.py prj-test.db

User guide & software design

In this system, user can do:

Login with password and userid, we implement a login screen function and then let the user to choose an operation. All user inputs are case insensitive, except the password.

Login as registry agent:

We implement each operation as a function

- *Register a birth*: Providing the first name, the last name, the gender, the birth date, the birth place of the newborn, as well as the first and last names of the parents. If any of the parents is not in the database, user need to provide information about the parent including first name, last name, birth date, birth place, address and phone.
- *Register a marriage*: Providing the names of the partners. If any of the partners is not found in the database, user need to provide information about the partner including first name, last name, birth date, birth place, address and phone
- *Renew a vehicle registration:* providing an existing registration number and renew the registration.
- *Process a bill of sale*: Recording a bill of sale by providing the vin of a car, the name of the current owner, the name of the new owner, and a plate number for the new registration. If the name of the current owner (that is provided) does not match the name of the most recent owner of the car in the system, the transfer cannot be made.
- *Process a payment*: Recording a payment by entering a valid ticket number and an amount.
- *Get a driver abstract*: Entering a first name and a last name and get number of tickets, the number of demerit notices, the total number of demerit points received both within the past two years and within the lifetime of a driver. The user can see the tickets ordered from the latest to the oldest and detail information. If there are more than 5 tickets, at most 5 tickets will be shown at a time, and the user can select to see more.

Login as traffic officer:

We implement each operation as a function

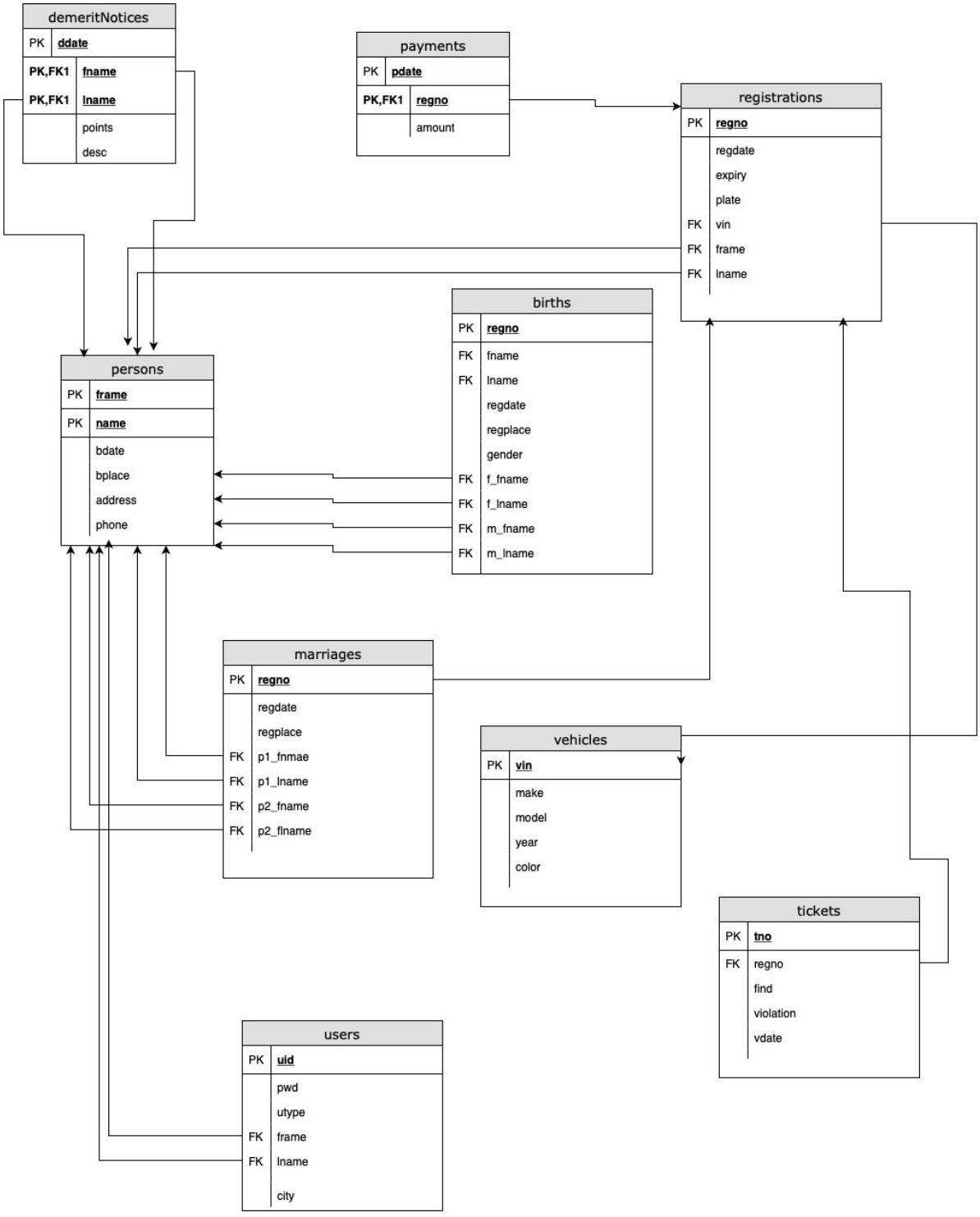
- *Issue a ticket*: Providing a registration number and see the person name that is listed in the registration and the make, model, year and color of the car registered. Then the user should be able to proceed and ticket the registration by providing a violation date, a violation text and a fine amount.
- Find a car owner: Providing one or more of make, model, year, color, and plate to see the car owner and detailed informations

More functions:

- Get person information: Allowing user to register person's information if someone is not in our database.
- Add one year: A function to modify the date in the database.

Detailed Description

- persons: Stores all exist persons information. We can search for exact person by its primary key: (fname, lname).
- births: Stores all newborn registrations, we can search for exact registration by its primary key: regno. The newborn baby and its parents are all stored in persons.
- marriages: Stores all marriages information, each partner has two registrations, one as partner1 the other one as partner2, we can search for one registration by its primary key: regno. Partner1 and partner2 are all stored in persons.
- vehicles: Stores all the cars' information.
- registrations: Stores all registrations with a vehicles. We can search for one registration by its primary key: regno. We can search for the vehicles' detailed information by searching the registration.vin as the vin in vehicles, the vin stored in vehicles as a primary key. Driver is stored in persons.
- tickets: Stores all tickets information. We can search for one ticket by its primary key: tno. We can check the detailed registration by using tickets.regno as regno in registrations. So that, we are able to know which vehicle and who get the ticket.
- demeritNotices: Stores all information demeritNotices. We can search for one demeritNotices by its primary key: (ddate,fname,lname). The driver who get the demeriNotice is in persons.
- payments: Stores the payments information of fine in tickets. We can search for one payment by its primary key: (tno, pdate). Notices that one fine can be paid multiple times, and the total amount of payments should less than or equal to the amount of fine of the ticket. The ticket can be found by using payments.tno as tno in tickets.
- users: Stores all users information who will use this system and operation. The users information are stored in persons. Different type (a or o) of users will have different permissions to manipulate the data. Only when both username and password are correct this user is allowed to login the system.


demeritNotices
PK ddate
PK,FK1 fname
PK,FK1 lname
points
desc
payments
PK pdate
PK,FK1 regno
amount
registrations
PK regno
regdate
expiry
plate
FK vin
FK frame
FK lname
births
PK regno
FK fname
FK lname
regdate
regplace
gender
FK f_fname
FK f_lname
FK m_fname
FK m_lname
persons
PK frame
PK name
bdate
bplace
address
phone
marriages
PK regno
regdate
regplace
FK p1_fnmae
FK p1_lname
FK p2_fname
FK p2_flname
vehicles
PK vin
make
model
year
color
tickets
PK tno
FK regno
find
violation
vdate
users
PK uid
pwd
utype
FK frame
FK lname
city

Testing Strategy

We test our program with a database which has different kinds of data, and we run our program with different scenarios of input: invalid character, empty, case-insensitive. Testing each function one by one.

New person: Can add a new person when its information not in the database in persons.

Date: Check the valid format of date are all the same as xxxx-xx-xx with a valid date.

Phone: Check the valid format of phone are all the same as xxx-xxx-xxxx with 12 characters.

Current Owner: Check the most recent current owner if exist in the registration with a valid name.

Payment amount: Check the total payment amount if less than or equal to the fine of the ticket.

Driver name: Check driver's name user entered exist in database.

Group Work Break-Down Strategy

We first build a menu of the total system and then try to realize the basic operations. Then optimize the methods and functions by adding more Error checkings and keeping tests with different cases.

Boyuan Dong [Spent 21 hours to work]: Develop the main structure of the project, program testing for each function implementation, database build, fix bugs.comment on how to evaluate query and test data.

Yi Ding [Spent 25 hours to work]: Develop each function of the project. Testing each functions and fix bugs.

We use google drive to keep the project on track.comment on how to evaluate query and test data.