QUERY TRAFFIC VIOLATION RECORD DATABASE SYSTEM

GROUP: GROUP 10

INSTRUCTOR: MR. TIAN TANG

SECTION: DBMS 1002

Group Members

徐铂凯 MICHAEL 1730014072

李欣岳 ERIKA 1730026048

林彦志 LARRY 1730026065

欣岳 李

Table of Contents

Project Description

Background Information

According to World Health Organization, the deaths related to the road traffic reach 1.35 million per year in the global range. And only in China, 250 thousand people die each year due to the traffic accident. The majority of these accidents are caused by the drivers' unsafe driving. Lots of traffic offenses like over speeding and running the red light occur every day, which heavily threaten both the pedestrians and other drivers. In addition, some uncivilized behaviors like throwing trash out the window happen from time to time. These kind of behaviors can also be potential threats for life safety. Many of the systems currently only focus on the offensive record such as over-speed, but for those uncivilized behaviors, these kind of systems seldom record. Therefore designing a database system to not only record the violation data but also the uncivilized behavior is urgent and necessary.

Purpose

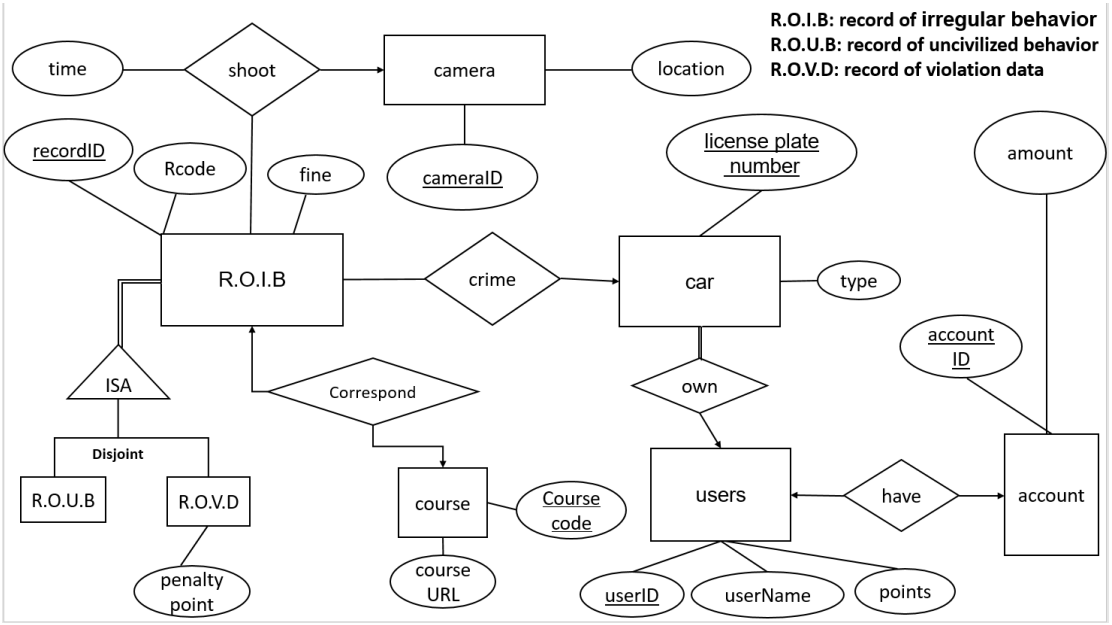
As we said above, the current database system only focuses on the violation data. We want to design a database system not only can show the violation data but also the uncivilized behavior as well. In addition, our database system can give the penalty as well as give the lessons and courses to maximum assure that the car owner will not have these behaviors again.

Functionalities

Our database system will achieve the following functions:

1. Can allow new users register the database by typing their name, identified number, license plate number and car type.
2. Can allow users search for whether they have irregular records or not by inputting users name and users ID
3. Can allow users delete the records by their-selves after learn the courses related to the irregular behaviors consciously.
4. Can recharge to their account any number of amounts.

ER Diagram



Explanations:

Entity Account has attribute accountID which represents the bank card ID of the users, and attribute amount representing the money that the user has in our database system

Entity Users has three attributes userID, userName and points which represents the points that a user's driving license has.

Entity Car has two attributes license plate number and type, which represents the car type like saloon car.

Entity Camera has attributes cameraID and location which represents the position that an irregular behavior is shot.

Entity R.O.U.B is the abbreviation for the Record of Uncivilized Behavior and Entity R.O.V.D is the abbreviation for the Record of Violation Data. These two entities are the subclass of R.O.I.B, which is the abbreviation for the Record of irregular behavior.

Entity Course has two attributes, the Course_code and the CourseURL, which gives the details and lessons about the irregular behavior.

There are 5 relationship sets in our model, the Have, the Own, the Crime, the correspond, the shoot which has an attribute called date to record when the irregular behavior is shot, and the IS-A relationship set.

Assumptions

To guarantee our database system function well, we must give some assumptions as follows:

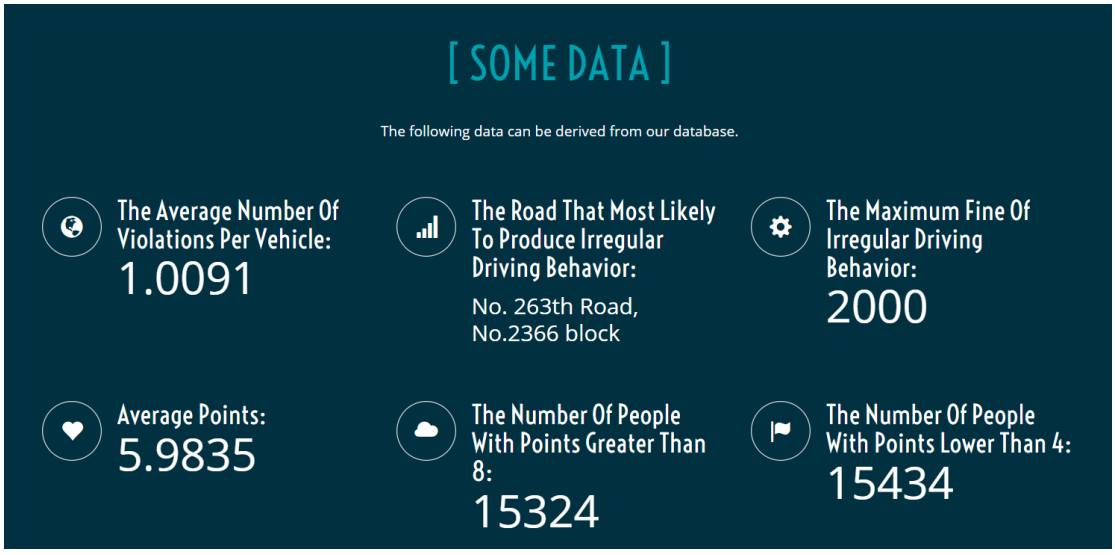
1. We can successfully connect to the database with all the citizen's information and have the right to access their information, including their ID number, name, car and their license plate number, etc. This should be supported by the police and any related departments.
2. We can successfully cooperate with the bank so that citizens can recharge their accounts in our system online. In addition, the account ID should be same as the bank card ID.
3. We can connect every camera on the street and cameras that we use to shoot citizen's irregular behavior are advanced enough that we can clearly identify what kinds of irregular behavior and citizens do. The camera should recognize different kinds of behaviors correctly and clearly. This should be supported by the technology departments.
4. We assume that we have get the support from the law or government policy that citizens who have irregular behavior have to take a course to increase their awareness.

Functions

After the camera takes an irregular behavior record, it automatically uploads the records to our database (Shoot, Record). Our web pages provide users with useful features such as registering accounts, checking for their own irregular behavior records, paying fines, deducting points and also recharge. Detailed introduction are below:

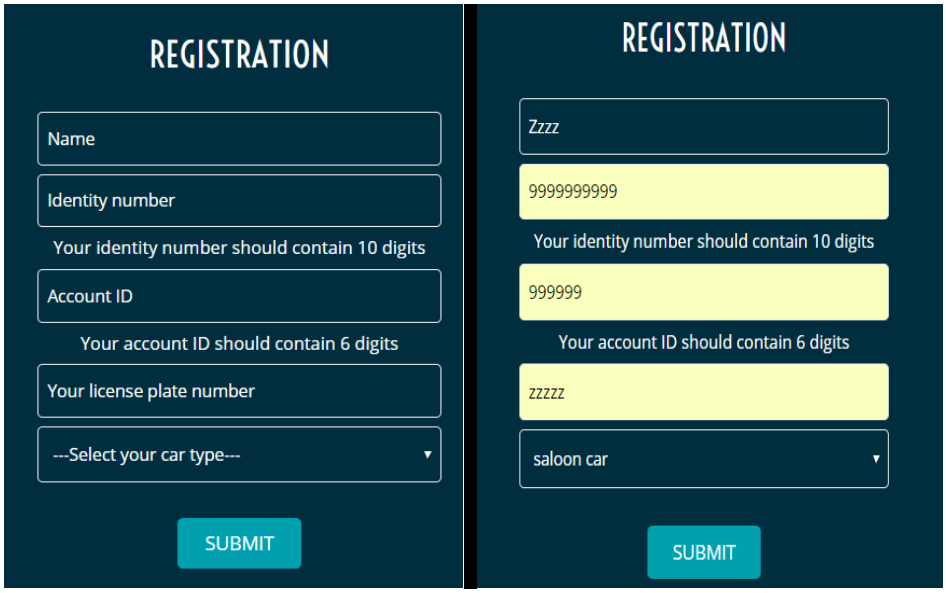
Homepage http://stuweb.uic.edu.hk/~m730026048/home.php

The homepage mainly includes our website name, promotional slogan, and some warning reminders and knowledgeable traffic safety promotion. The home page is also connected to several other feature pages in our website. What's more, on our homepage, we'll show some of the characteristics of the information in our database in real time, which are derived from our database. For example, the average number of irregular records per vehicle is 1.0091 for now. Most people have irregular behavior records is on the No263th Road, No.2366 block for now. This is our innovative features.



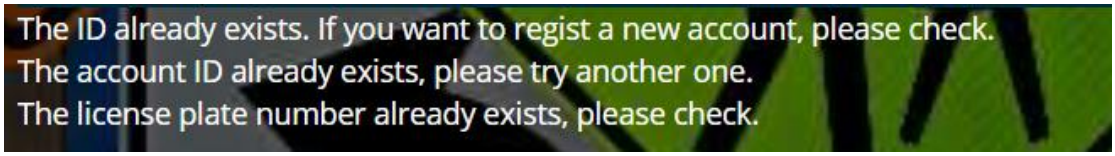
Registration http://stuweb.uic.edu.hk/~m730026065/registration.php

This page allows users to register for a new account. After entering your name, identity number, account ID you want create, license plate number, car type and clicking submit, it will automatically register a new account for you.



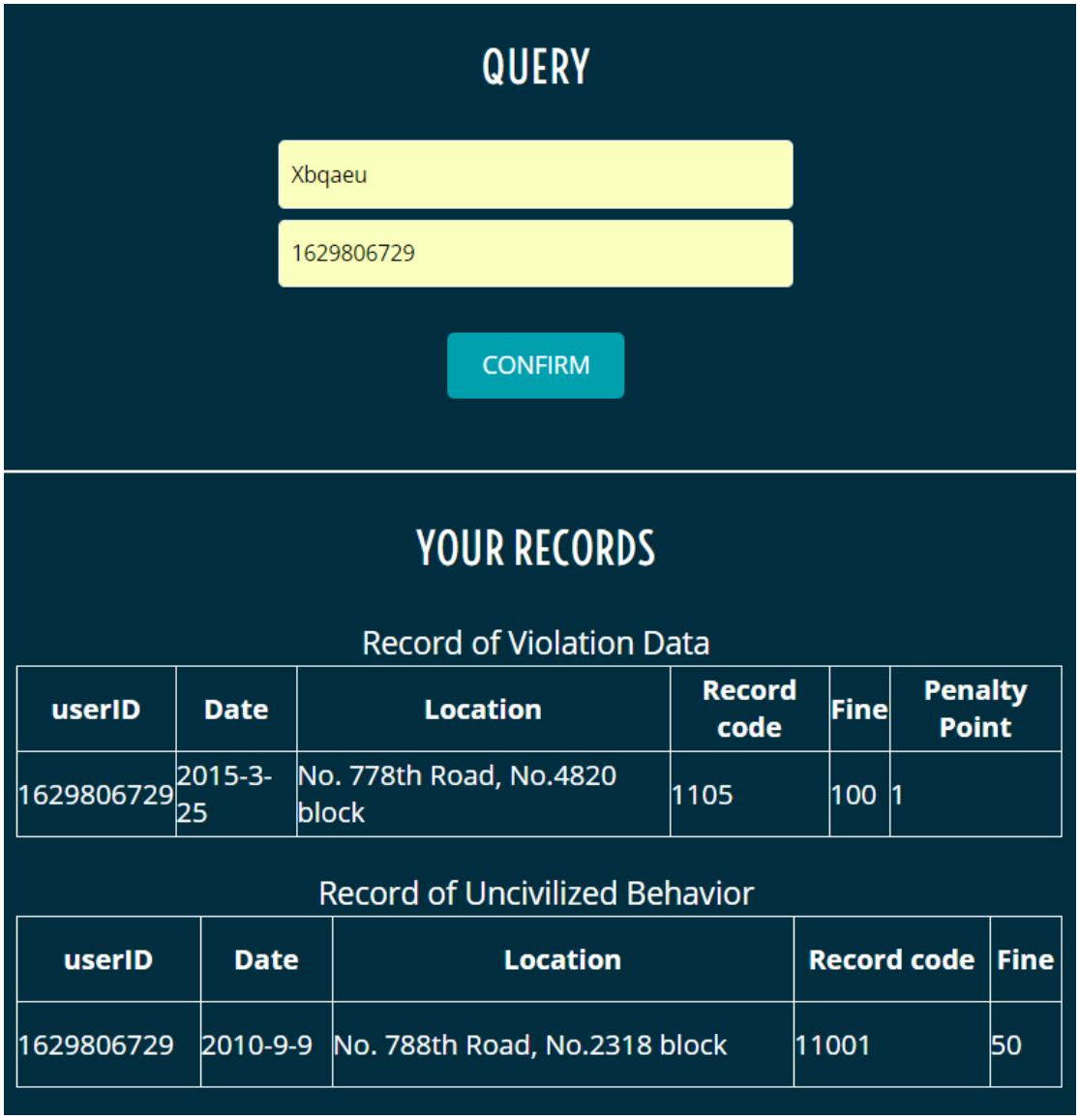
However, there exists some other cases:

1. Identity Format error: your identity number should contain 10 digits. Otherwise it will show “Identity number format error” and cannot register successfully.
2. Account ID format error: your account ID should contain 6 digits. Otherwise it will show “Account ID format error” and cannot register successfully.
3. We allow one person owns serval cars. So, when people want to the new car plate number to their account, they can still register on our webpage. After entering the name and ID number, new license plate number, it will show “the ID already exists.” and it can still add the new license plate number to their account if the license plate number doesn’t exist in our database.



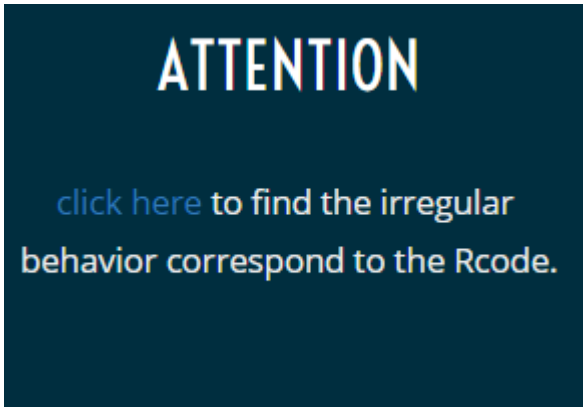
Query http://stuweb.uic.edu.hk/~m730026065/query.php

1. This webpage can display a lot of details about your irregular behavior records. After entering your name and ID number, it will show all the irregular behavior records you have generate (include all your car). The display information contains a lot of details, such as the date, the location you generate the record, the punishment for this record (fine and penalty points).



Also, the webpage will display the Record code, which you can refer to what kind of irregular behavior you have made.

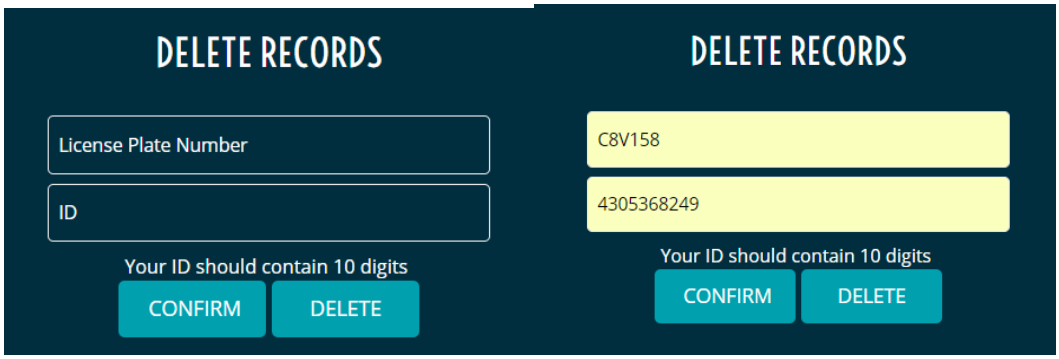
http://stuweb.uic.edu.hk/~m730026048/co.html

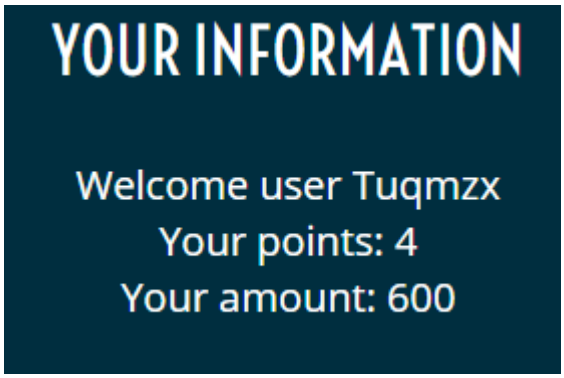


Manage http://stuweb.uic.edu.hk/~m730026065/manage.php

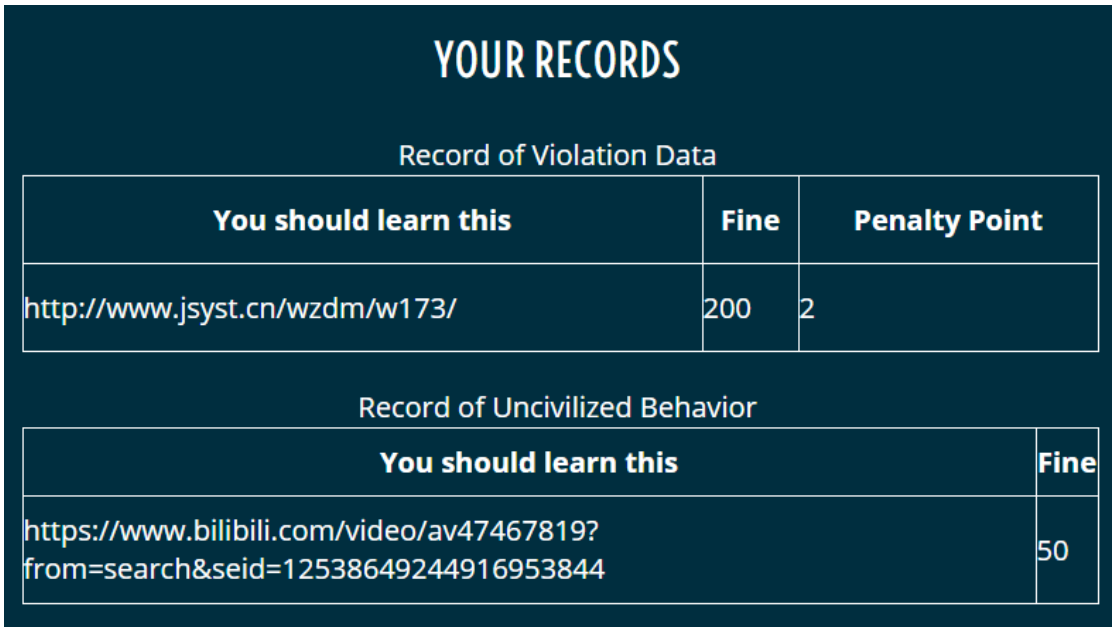
This page has very powerful functions. It has two main functions, querying and deleting records. Although in real life, people can't delete their own records, this needs to be done by the police, but people can still reduce their points and pay fines. We are here only to improve people's awareness of traffic safety, not to make mandatory learning requirements. We only need them to consciously complete the relevant courses.

Query: This shows the information is less than the information displayed on the Query webpage. After entering your license plate number and ID number, click confirm, it will display your personal information, such as the name, your remaining points, and the balance in your account.

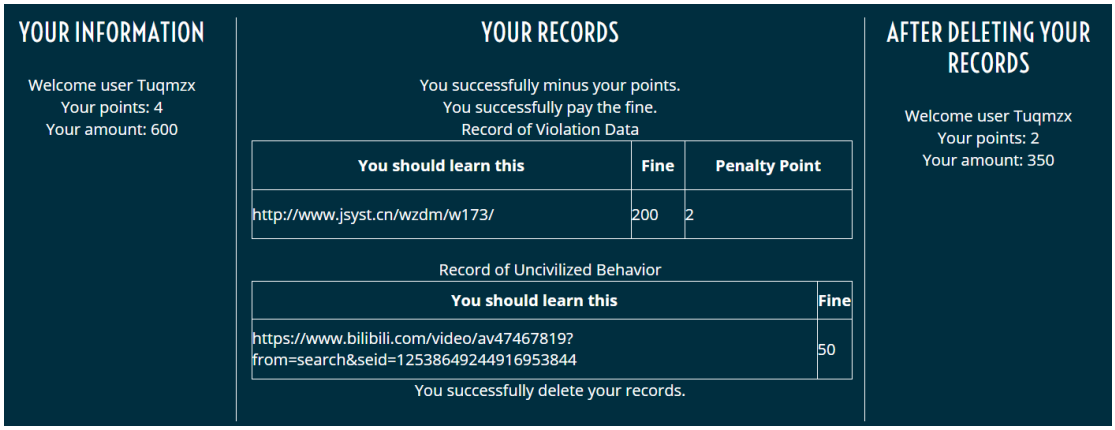




At the same time, the web page will also display your irregular behavior records (Record of violation data and Record of uncivilized behavior) and the corresponding punishment. Both records require a relevant course (show course URL) and a fine. But the violation behavior needs to deduce your points since it is more serious.

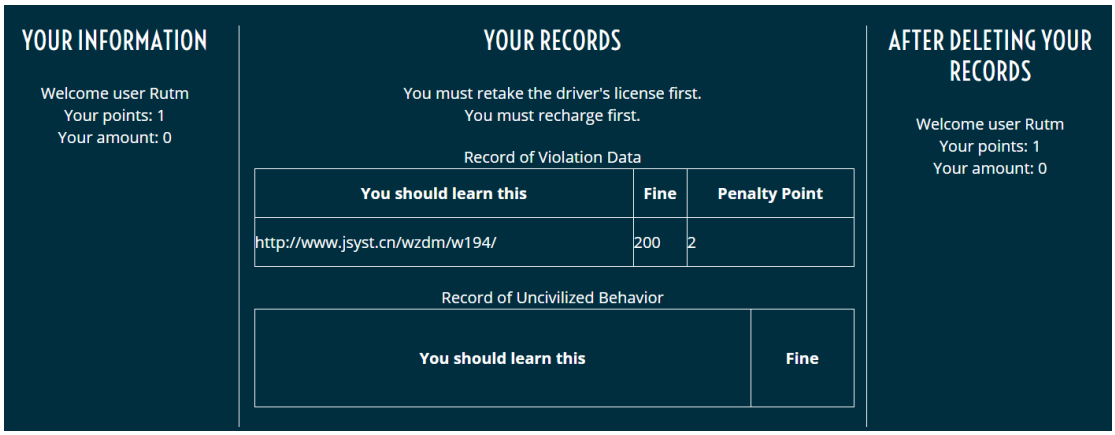


Delete: After entering your license plate number and ID number, click delete, it will display your personal information, such as the name, your remaining points, the balance in your account on the left-hand side before deleting your record. On the right-hand side, it will show the information after deleting the records. At the middle, it will still show the records that you delete. However, they are deleted already in the database. Once it is successfully done, it will show the notices: You successfully minus your points/ pay the fine/ delete your records. However, there exists some other cases:



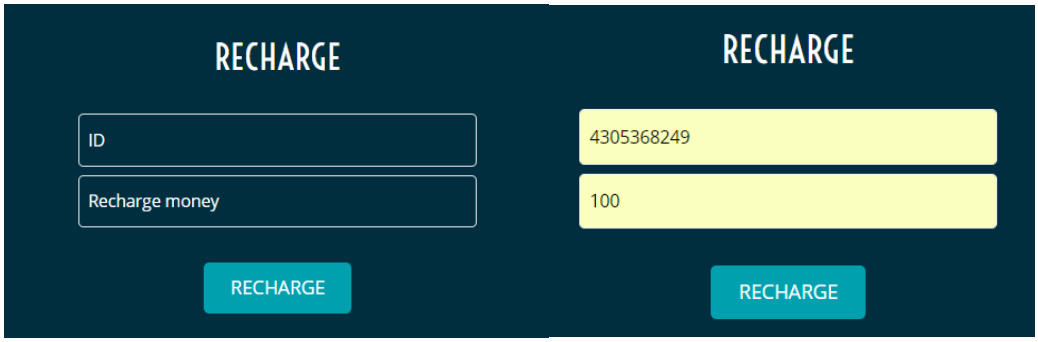
1. Not enough points: it will show: You must retake your license first.
2. Not enough amount: it will show: You must recharge first.

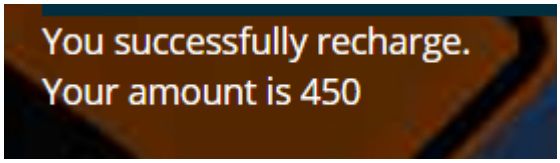
In both cases above, the record will not be deleted and the balance and points will remain unchanged.



Recharge http://stuweb.uic.edu.hk/~m730026065/recharge.php

This page is for you to recharge your amount when the balance is insufficient. After entering your ID number, recharge money number, click recharge, it will display the notice: You successfully recharge and the current amount of your account.





Other user information are provided below. Have a try if you want.

userID	LicensePlateNumber	userName	points	amount
3968360773	C0A119	Mbp	1	1400
4165978697	C0A125	Yiqxzhs	12	300
4050992649	C0A128	Seqh	1	800
4297287197	C0A139	Cjhu	5	100
4398990975	C0A147	Dufbzt	9	1000
2588209755	C0A148	Dety	3	100
1841916362	C0A158	Cvi	8	0
1433708010	C0A164	Muavt	4	900
3070648447	C0A165	Xay	0	200
1651532896	C0A175	Ejzmoq	5	1000
3589968910	C0A178	Wfm	3	1900
3608702521	C0A181	Rfxjtz	12	300
2190836694	C0A183	Oupc	3	1200
2731319990	C0A192	Cyql	2	1600
2732148920	C0A197	Ohlidi	8	300
2833231324	C0A201	Jzr	7	1000
3293766567	C0A204	Nrdy	4	1100
2610157135	C0A206	Nwjiyi	8	200
2698463703	C0A207	Sox	4	1300
3429529370	C0A208	Ekkf	11	300
2114339126	C0A209	Csii	11	1500
3362102780	C0A212	Ywswtnz	11	1500
4054465786	C0A219	Ywrbn	3	600
3128841223	C0A220	Yrp	3	100
3328346918	C0A225	Hhwz	6	1200

Functional dependency

We check the distribution of our database and we want to reduce the redundancy. The table ROIB (rid, Rcode, fine) have two functional dependency:

F= { recordID → Rcode, fine Rcode →fine}

This is not in 3rd normal form and will cause some redundancy, so we do the decomposition. The following are the process:

R2 = record = {recordID, Rcode }

R1 = ROIB = { Rcode, fine}

And ROIB have two subclass ROUB, ROVD and they completely divided the super class into two disjoint class, so we don't need to establish the table of ROIB and eventually our database has eight entities and five relationship sets, which are ROUB, ROVD, camera, record, car, user, account, course, correspond, shoot, crime, own and have. The relation schemas are as follow:

Record {recordID, Rcode }

ROUB (Rcode, fine)

ROVD (Rcode, fine, penalty_point)

Camera (cameraID, location)

Car (LicensePlateNumber, Type)

User (userID, userName, points)

Account (accountID, amount)

Course (Course_code, CourseURL)

Correspond (Rname, Course_code)

Shoot (recordID, cameraID, date)

Crime (recordID, license plate number)

Own (license plate number, type, userID)

Have (userID, accountID)

The functional dependencies are:

Record:

recordID → RcodeROUB

ROUB:

Rcode → fine

ROVD:

Rcode→ fine, penalty_point

Camera:

cameraID→ location

Car:

LicensePlateNumber→ Type

User:

userID→ userName, points

Account:

accountID→ amount

Course:

Course_code→ CourseURL

Correspond:

Rcode→ Course_code

Shoot:

recordID→cameraID, date

Crime:

recordID→license plate number

Own:

license plate number→type, userID

Have:

userID→accountID

The SQL Codes and Explanation

First we create our table in our database and following are the SQL code:

```
Create Account(
accountId int primary key,
amount int
)
```

- Create the table Account and accountID is the key.

```
Create Camera(
CameraID int primary key,
location tinytext
)
```

- Create the table Camera and CameraID is the key.

```
Create Car(
LicensePlateNumber varchar(20) primary key,
Type varchar(20)
)
```

- Create the table Car and LicensePlateNumber is the key.

```
Create Correspond(
Rcode int primary key,
Course_code varchar(10)
)
```

- Create the table Correspond and Rcode is the key.

```
Create Course(
Course_code varchar(20) primary key,
CourseURL text
)
```

- Create the table Course and Course_code is the key.

```
Create Crime(
```

recordID varchar(10) primary key,

LicensePlateNumber varchar(20)

)

- Create the table Crime and recordID is the key.

Create Have(

userID varchar(20) primary key,

accountID int

)

- Create the table Have and userID is the key.

Create Own(

LicensePlateNumber varchar(20) primary key,

Type varchar(20),

userID varchar(20)

)

- Create the table Own and LicensePlateNumber is the key.

Create Record(

recordID varchar(10) primary key,

Rcode int

)

- Create the table Record and recordID is the key.

Create ROUB(

Rcode int primary key,

fine int

)

- Create the table ROUB and Rcode is the key.

Create ROVD(

Rcode int primary key,

penaltyPoint int,

fine int

)

- Create the table ROVD and Rcode is the key.

  Create Users(

  userID varchar(20) primary key,

  username varchar(10),

  points int

)

- Create the table Users and userID is the key.

  CREATE TABLE Shoot(

  recordID VARCHAR(10) PRIMARY KEY,

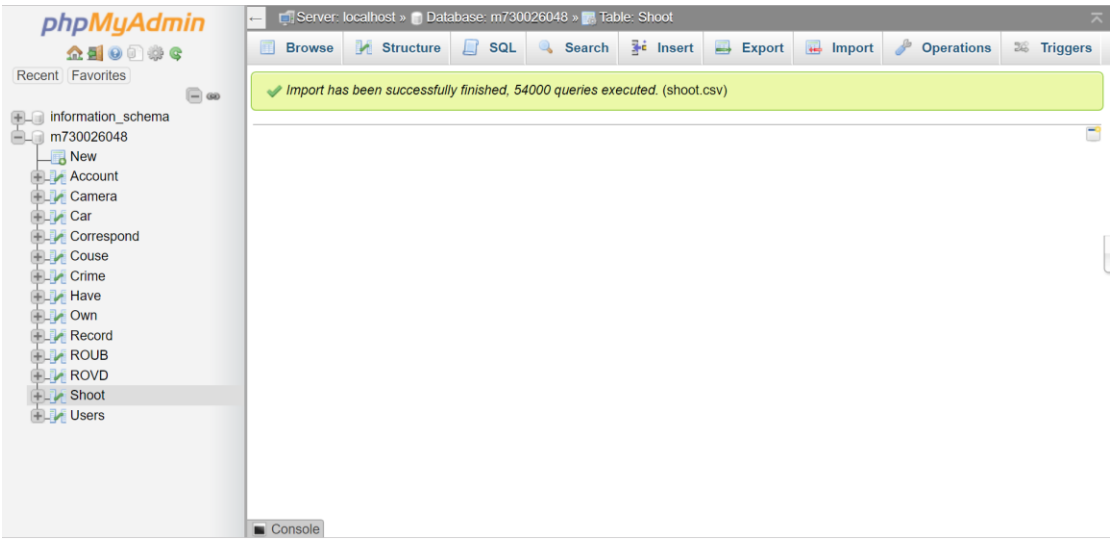
  cameraID INT,

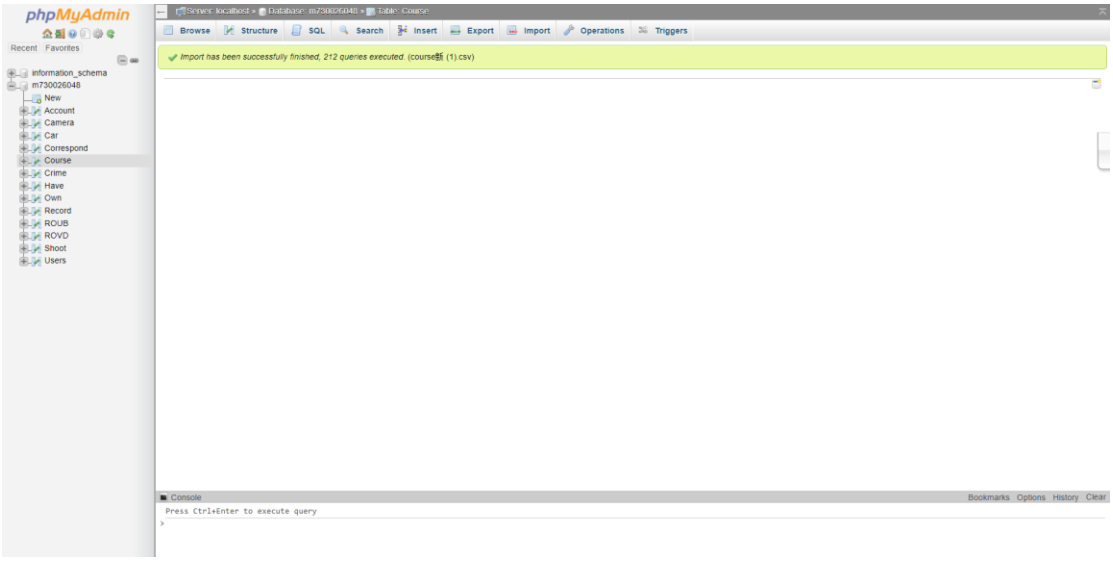
  date TEXT

)

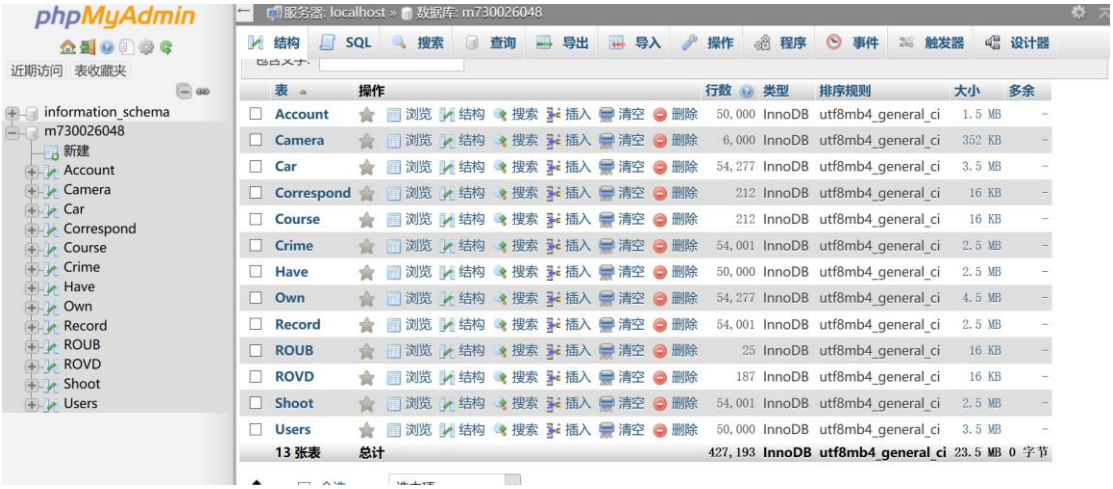
Create the table Shoot and recordID is the key.

And we import the data into the database and following are some result:





And then we establish our database:



- To check if the user we want to insert already there.

```
select *
from Users
where userID = '$userID'
```

- To check if the account we want to insert already there.

```
select *
from Account
where accountID = '$accountID'
```

- To check if the car we want to insert already there.

```
select *
from Car
```

where LicensePlateNumber = '$plateNum

- To do the registration, we want to insert a user into our table, and we read the value from the php and for each user, we assume the default points is 12.

```
INSERT INTO Users (userID,userName,points)
VALUES ('$userID','$nameValue',12);
```

- To do the registration, we want to insert a car into our table, and we read the value from the php

```
INSERT INTO Car (LicensePlateNumber,Type)
VALUES ('$plateNum','$carType');
```

- To do the registration, we want to insert a relation into our table, and we read the value from the php

```
INSERT INTO Own (LicensePlateNumber,type,userID)
VALUES ('$plateNum','$carType','$userID');
```

- To do the registration, we want to insert a relation into our table, and we read the value from the php

```
INSERT INTO Account (accountID,amount)
VALUES ('$accountID',0);
```

- To do the registration, we want to insert a relation into our table, and we read the value from the php

```
INSERT INTO Have (accountID,userID)
VALUES ('$accountID','$userID');
```

- We want to query the result of the user's ROVD and we use this SQL to find the record of this user.

```
SELECT userID, date, location, Record.Rcode, ROVD.fine,
ROVD.penaltyPoint
FROM Users natural join Own
    natural join Crime
    natural join Shoot
```

```
natural join Camera
natural join Record
natural join ROVD
WHERE userID = '$userID' and userName = '$nameValue'
```

- We want to query the result of the user's ROUB and we use this SQL to find the record of this user.

```
SELECT userID, date, location, Record.Rcode, ROUB.fine
FROM Users natural join Own
    natural join Crime
    natural join Shoot
    natural join Camera
    natural join Record
    natural join ROUB
where userID = '$userID' and userName = '$nameValue'
```

- If the user wants to recharge money into their account, we can use this SQL to update the amount.

```
UPDATE Account, Have
SET Account.amount=Account.amount+'$rechargeValue'
WHERE Account.accountID=Have.accountID AND
userID='$userID'
```

- We want to find the record under the license plate number we read in php

```
SELECT userID, userName, amount, points
FROM Users natural join Have natural join Account natural join
Own
WHERE userID = '$userID' and LicensePlateNumber = '$Platenum'
```

- Find the total points the user should decrease

```
SELECT sum(penaltyPoint)
FROM Crime NATURAL JOIN Record NATURAL JOIN ROVD
WHERE LicensePlateNumber = '$Platenum'
```

- Find the points the user has

```
SELECT points
FROM Users
WHERE userID='$userID'
```

- Find the total fine the user should pay due to ROVD

```
SELECT sum(fine)
FROM Crime NATURAL JOIN Record NATURAL JOIN ROVD
WHERE LicensePlateNumber = '$Platenum'
```

- Find the total fine the user should pay due to ROUB

```
SELECT sum(fine)
FROM Crime NATURAL JOIN Record NATURAL JOIN ROUB
WHERE LicensePlateNumber = '$Platenum'
```

- Find the amount of money the user has

```
SELECT amount
FROM Have NATURAL JOIN Account
WHERE userID='$userID'
```

- Delete the points on user

```
UPDATE Users, Own
SET Users.points = Users.points - '$value4'
WHERE Users.userID=Own.userID AND Users.userID='$userID'
```

- Delete the amount on user

```
UPDATE Account, Have
SET Account.amount='$balance'
WHERE Account.accountID=Have.accountID AND
userID='$userID'
```

- We want to delate the record under the license plate number we read in php.

```
DELETE FROM Crime
where LicensePlateNumber = '$Platenum'
```

- Find the average number of violations per vehicle

```
SELECT AVG(number)
FROM (SELECT COUNT(recordID) AS number
FROM Crime
GROUP BY LicensePlateNumber) AS N
```

- Find the road that most likely to produce irregular driving behavior

```
SELECT location
FROM Camera
WHERE Camera.CameraID IN(
 (SELECT CameraID
FROM Shoot
GROUP BY(CameraID)
 HAVING COUNT(recordID)=(
 SELECT MAX(number)
 FROM (
SELECT CameraID,count(recordID) AS number
    FROM Shoot
    GROUP BY(CameraID))AS N1)))
```

- Find the maximum fine of irregular driving behavior

```
SELECT MAX(fine)
FROM ROVD
```

- Find average points of the users

```
SELECT AVG(points)
FROM Users
```

- Find the number of people with points greater than 8

```
SELECT COUNT(userID)
FROM Users
WHERE points>8
```

- Find the number of people with points lower than 4

```
SELECT COUNT(userID)
```

FROM Users

WHERE points<4

Difficulties and Solutions

Difficulties:

1. Come out with an idea (project topic) that is innovative
2. What features do our web pages have to implement?
3. PHP and html are hard since we are not familiar with them.
4. How to convert user-entered content on the webpage to a part of the SQL statement
5. SQL code, especially Update

Solutions:

1. Brain storm—based on the life experience and what people really need in the real life
2. After drawing our ER model, we design the basic framework of our website (how many webpages we need, what function provided, how to connect different webpages, etc.)
3. Learning from online resources. Baidu, Google, etc. Ask the classmates and CST students, thank them
4. Same as point 3
5. Learning from the PPT provided on this course. Also learning from the Internet.

Improvement

Our webpage has basically implemented most of the functions of our project and also applied most of the SQL statements. On our homepage, to achieve the function to display the general information about the irregular behavior records, we apply *groupby, aggregation function.* To achieve the function to register a new account for the users, we apply *insert function.* To display the users' information on our webpage, we apply *select function.* And also we also provide a lot of

friendly tips to our users in the process of using our website, such as *successfully registrar, format error*, etc.

However, there still has a lot of improvements. The situation in real life is more complex than we image. It still need to test more. And also we should generate a back-end management system that provides the police with system maintenance and record changes (we assume that the camera automatically uploads the irregular behavior records to our database after they shot).