**数据库原理综合实验报告**

**题　　目:**  停车收费管理系统

**班 级：** 软工202

**组员：**

**姓名：** 彭长辅 **学号：** 2000770043

**姓名：** 曾祥龙 **学号：** 2000770037

**院　　系：** 计算机科学与技术学院

**专业年级：**  大二

目录

[一、 数据库设计 4](#_Toc106133821)

[1. 需求分析和安全性方面描述： 4](#_Toc106133822)

[1) 需求分析： 4](#_Toc106133823)

[2) 安全性和完整性方面： 4](#_Toc106133824)

[2. 概念结构设计： 5](#_Toc106133825)

[1) 各类实体 5](#_Toc106133826)

[2) 转换为关系模式： 7](#_Toc106133827)

[3. 逻辑结构设计 7](#_Toc106133828)

[1) 模式设计 7](#_Toc106133829)

[2) 建表以及插入的数据 8](#_Toc106133830)

[3) 子模式设计 11](#_Toc106133831)

[4) 索引设计 13](#_Toc106133832)

[4. 数据库的安全性和完整性 13](#_Toc106133833)

[1) 定义触发器 13](#_Toc106133834)

[2) 用户定义 13](#_Toc106133835)

[3) 权限定义 14](#_Toc106133836)

[5. 案例 14](#_Toc106133837)

[触发器测试 14](#_Toc106133838)

[二、 服务端设计 15](#_Toc106133839)

[1. 简介 15](#_Toc106133840)

[2. 项目结构解释 15](#_Toc106133841)

[3. 开发思路以及部分代码 16](#_Toc106133842)

[三、 客户端设计 18](#_Toc106133843)

[1. 简介 18](#_Toc106133844)

[2. 项目结构解释 18](#_Toc106133845)

停车收费管理系统设计

**简介**：本系统针对社区内车主车辆，以及车辆停车，费用缴纳等需求提供后台管理服务，文档中具体说明数据库设计的思路，以及前后端的技术路线及展示。

项目已托管于Github，地址为

[JinlinXingjian/DataBaseHomeWork: 数据库作业仓库 (github.com)](https://github.com/JinlinXingjian/DataBaseHomeWork)

# 数据库设计

## 需求分析和安全性方面描述：

### 需求分析：

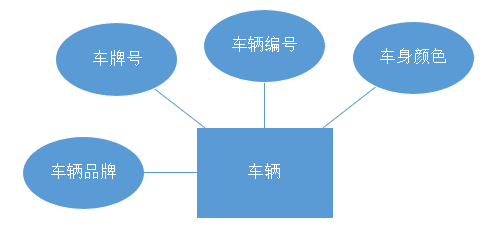
* 记录车主信息
* 记录车辆信息
* 记录车位信息
* 记录车辆出入日志
* 计算需要缴纳的费用

### 安全性和完整性方面：

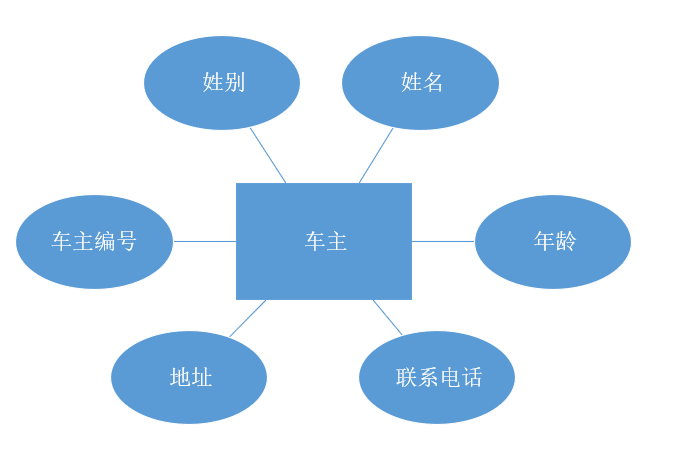
* 用户：车主，管理员
* 权限：车主能够注册，登录系统，可以查看自己拥有的车辆，管理员能够对车主，车位，车辆等信息进行增删改查
* 设计视图：根据车主性别，是否缴费等信息形成视图
* 设计触发器：根据不同车位的收费标准以及停车时长在插入日记记录表时计算出停车费用

## 概念结构设计：

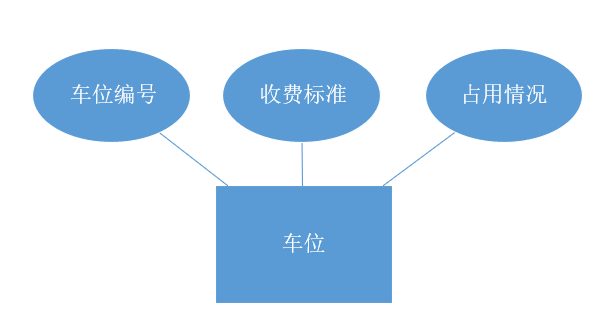
### 各类实体



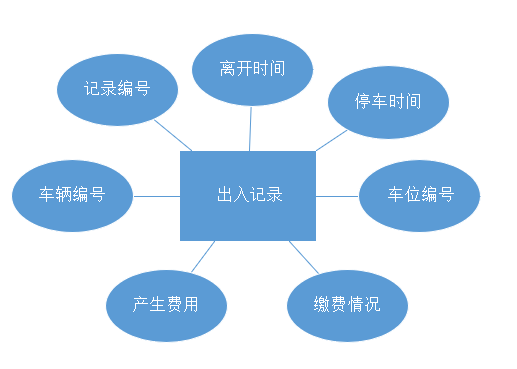
（图一） 车辆实体



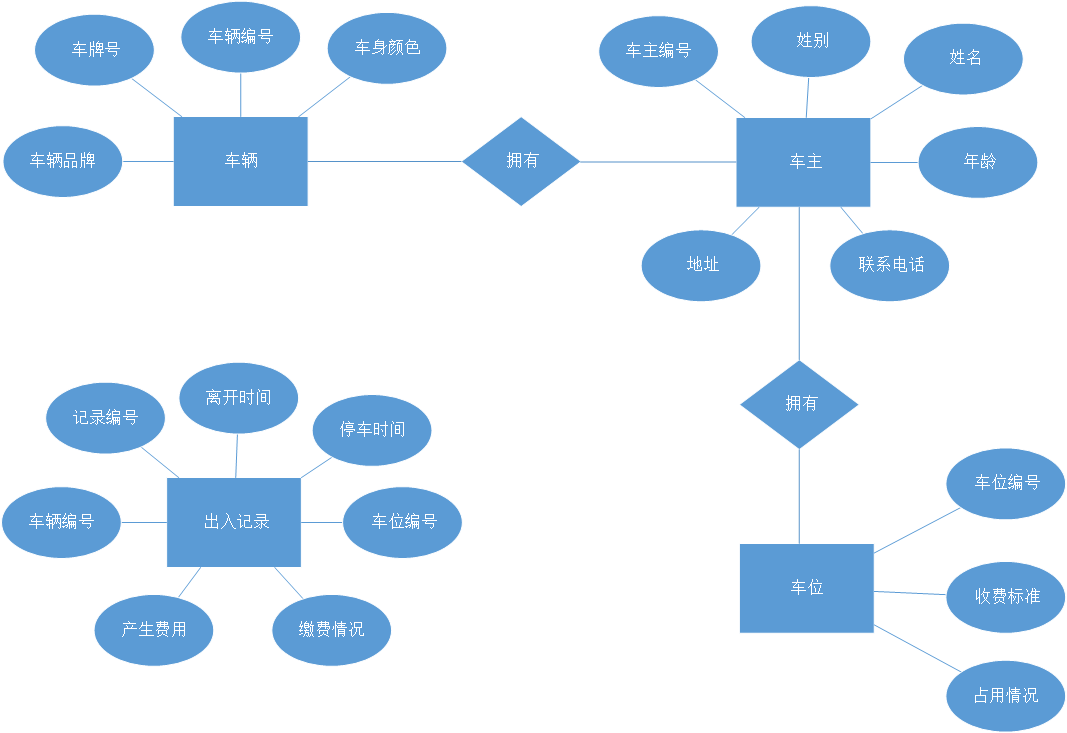
（图二）车主实体



（图三）车位实体



（图四）出入记录实体



（图五） 实体联系图

### 转换为关系模式：

车辆（车辆编号，车牌号，车辆品牌，车身颜色，车主编号）

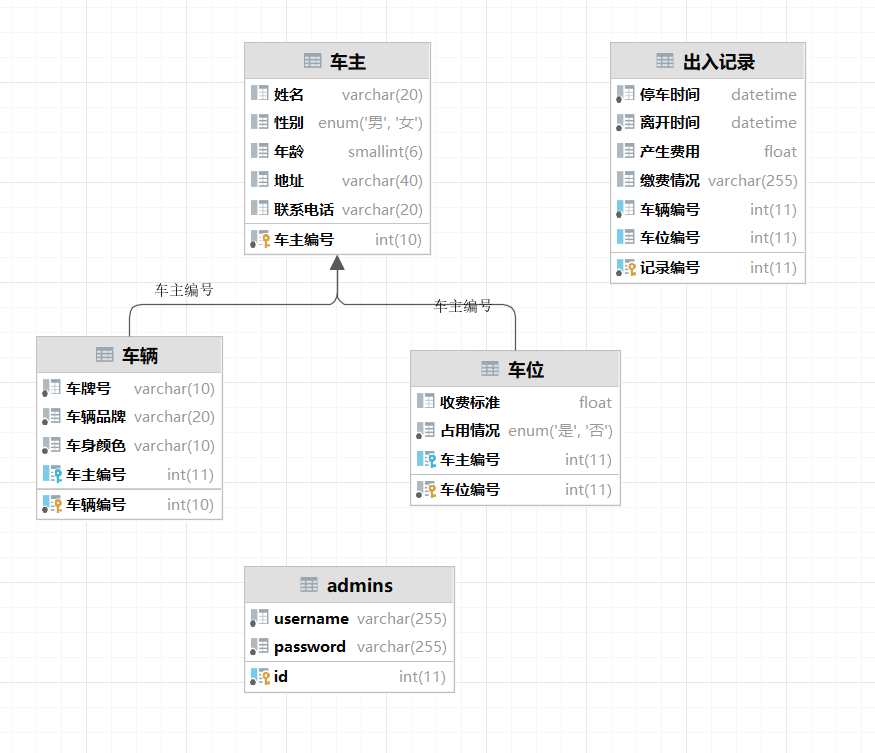
车主（车主编号，姓名，性别，年龄，地址，联系电话）

车位（车位编号，收费标准，占用情况，车主编号）

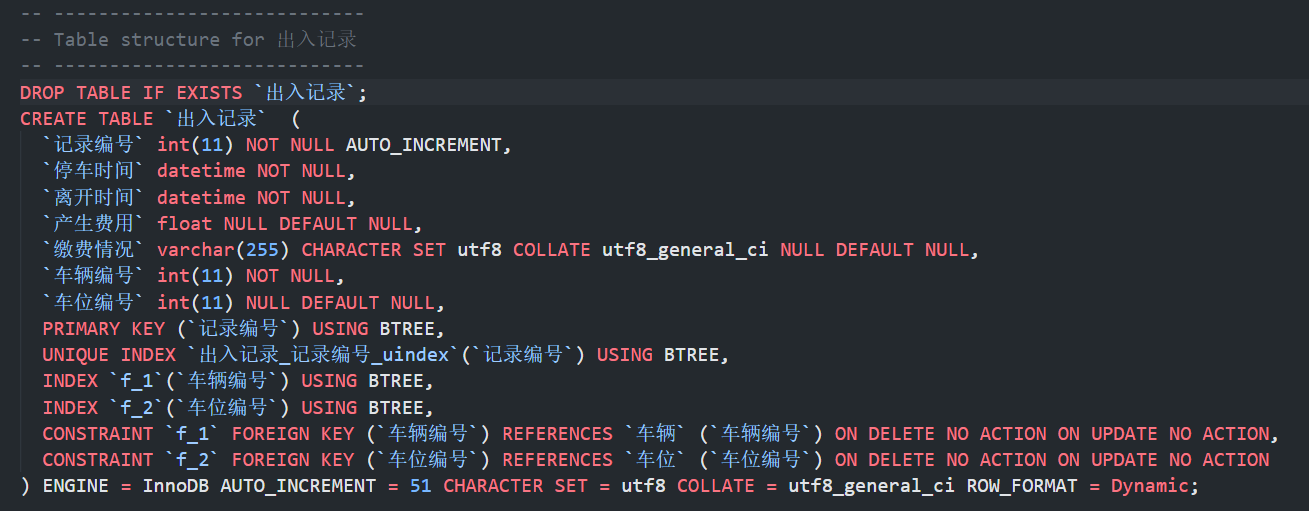
出入记录（记录编号，停车时间，离开时间，产生费用，缴费情况，车辆编号，车位编号）

## 逻辑结构设计

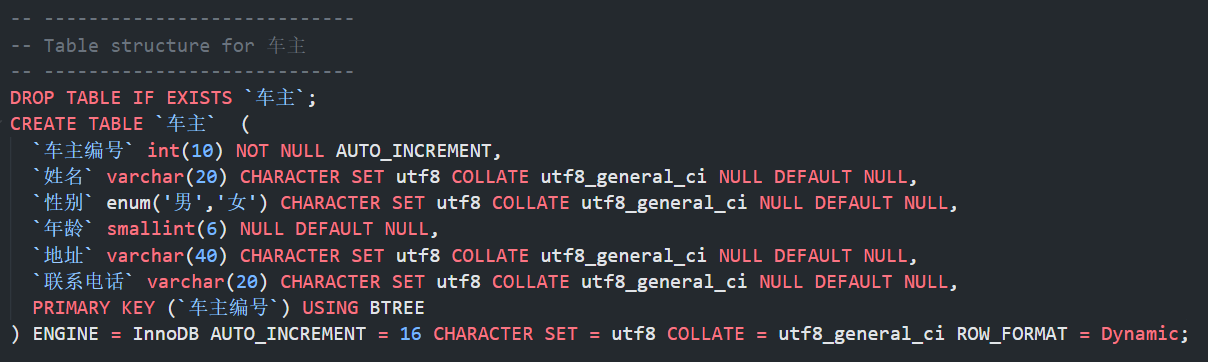
### 模式设计



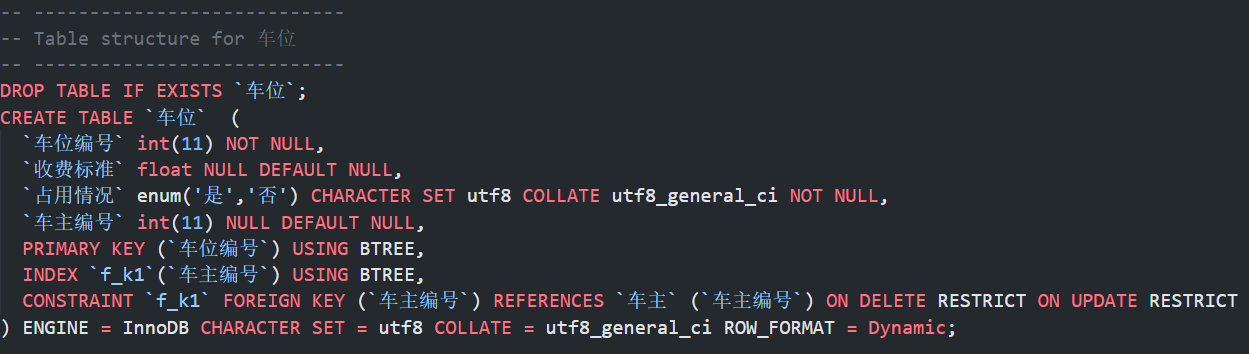
### 建表以及插入的数据



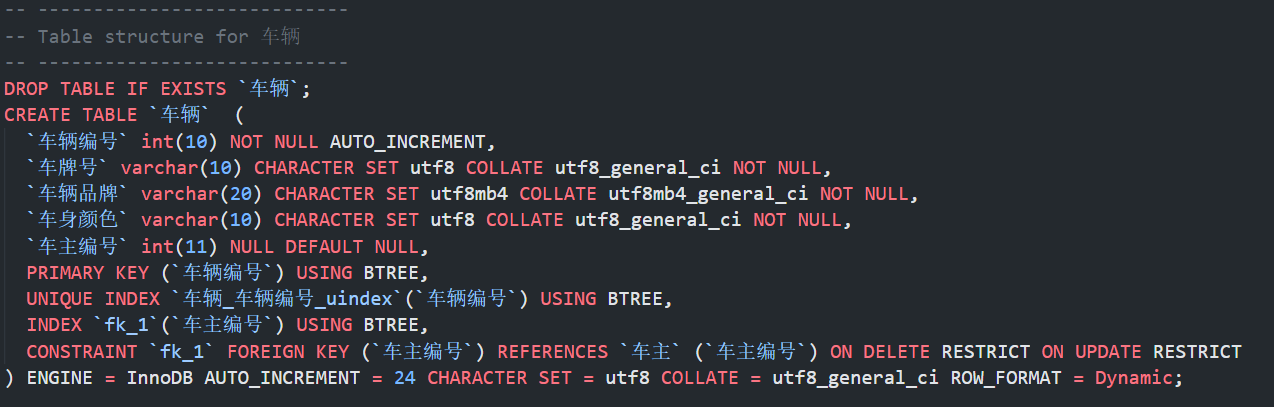
（1）创建出入记录表



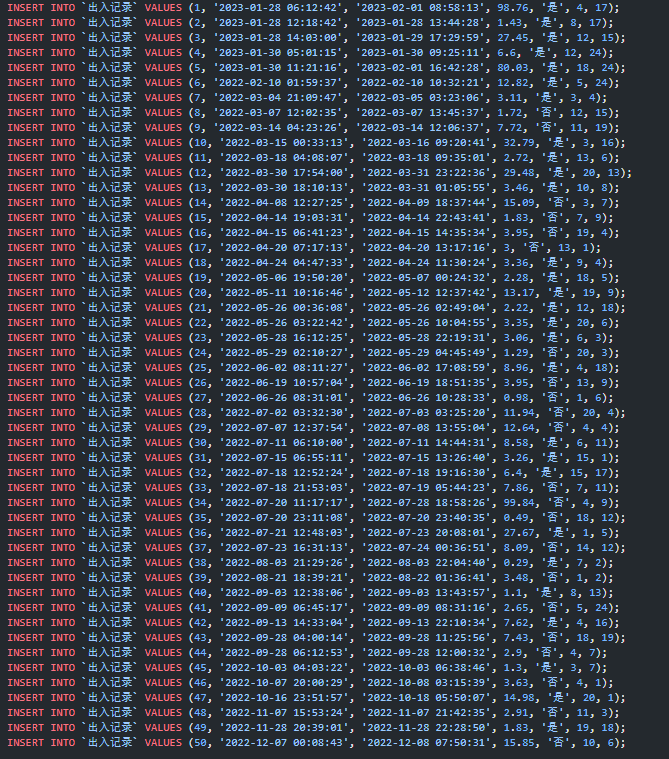
（2）创建车主表



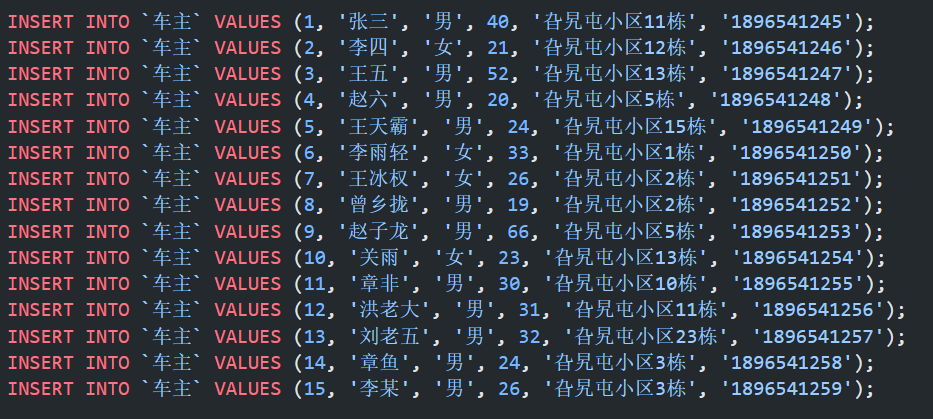
（3）创建车位表



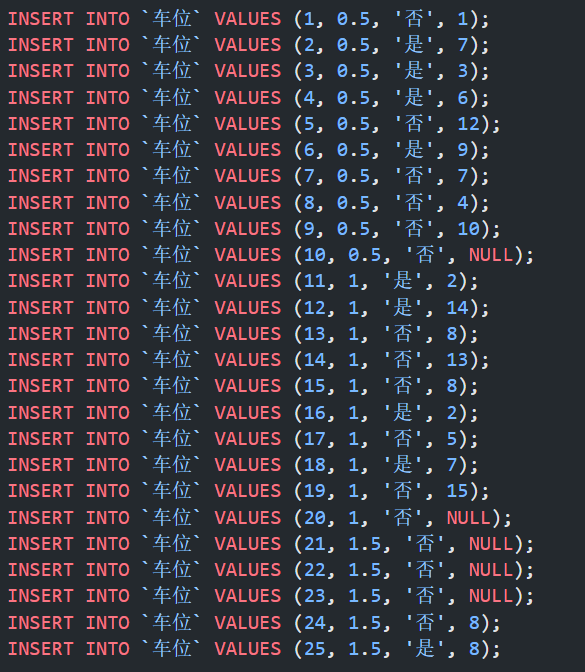
（4）创建车辆表



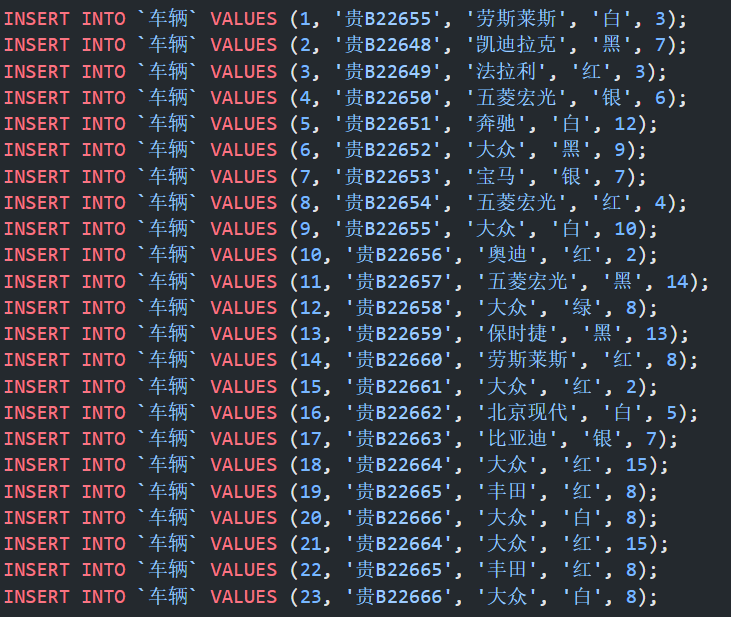
（5）向出入记录中插入数据



（6）向车主表中插入数据



（7）向车位表中插入数据



（8）向车辆表中插入数据

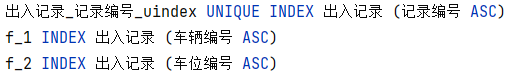
### 子模式设计





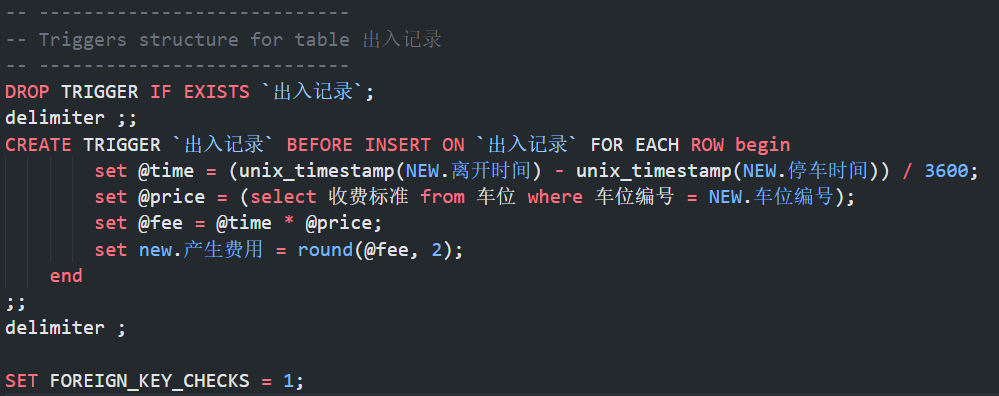


### 索引设计

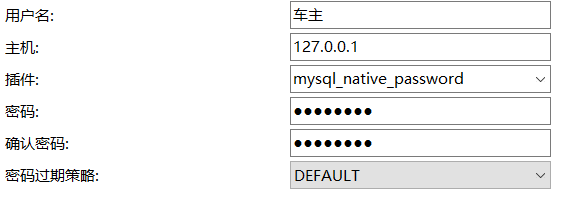


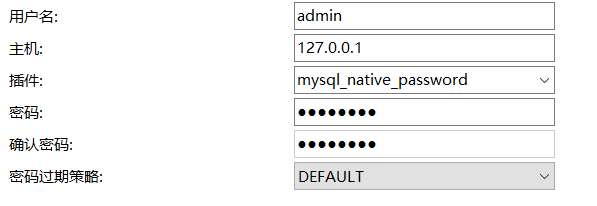
## 数据库的安全性和完整性

### 定义触发器

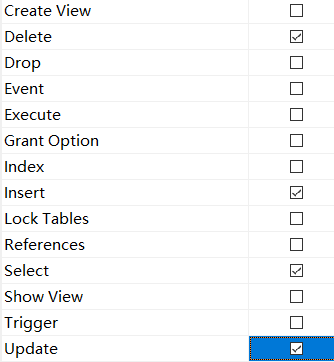
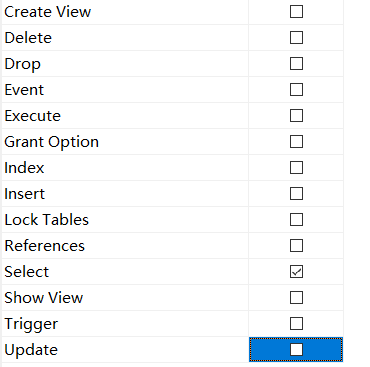


### 用户定义





### 权限定义

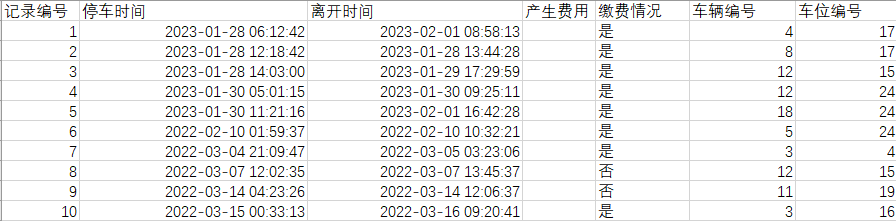


车主权限 管理员权限

## 案例

### 触发器测试

插入数据前：



插入数据后：（自动根据车位收费标准计算停车费用）



# 服务端设计

## 简介

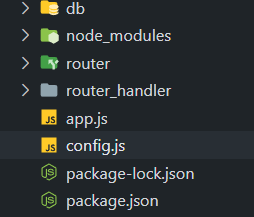
服务端使用node.js开发，开发语言为JavaScript，开发框架为[express.js](https://www.expressjs.com.cn/)。

主要实现功能有：

* 针对管理员进行的登录注册
* 对管理员密码使用bcrypt算法进行加密
* 采用jwt验证机制进行登录（即生成有时效的token，访问者需提供token才能进行访问）
* 编写对车位，车辆，车主三个表的增，删，查，改接口，以及对出入记录的查询接口

## 项目结构解释

项目结构图如下

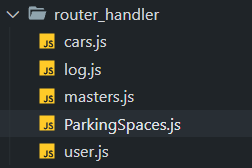
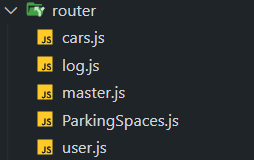


其中

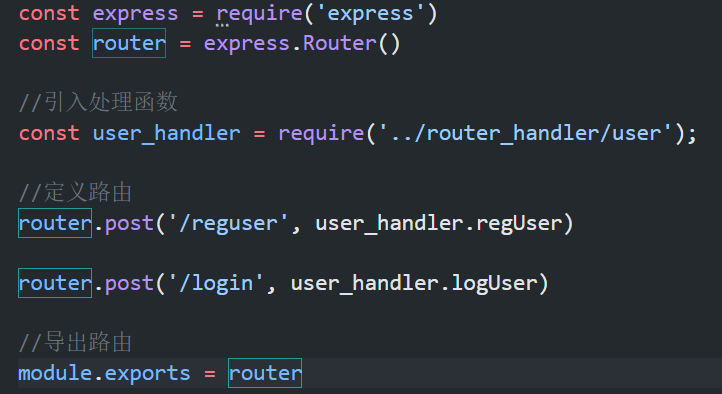
* db是配置数据库的程序，规定了使用哪个数据库，哪个用户等信息，并导出数据库模块
* node\_modules是项目使用到的第三方依赖
* router中规定了用户访问的路由，以及绑定了router\_handler中的对应处理函数，如规定一个post请求的前缀以及处理函数
* router\_handler中定义了具体的处理函数
* app.js 是服务器启动的入口，其中定义了服务器实例，引入了定义的路由，定义了全局错误处理的中间件
* config.js中定义了一些全局配置，如加密算法中的密钥字符串以及其他参数
* package-lock.json中定义了第三方依赖使用的依赖以及版本号，确保了npm install 后的一致性
* package.json 记录了服务器使用的开发依赖和上线依赖，入口文件，以及发布的版本号和包名等信息

## 开发思路以及部分代码

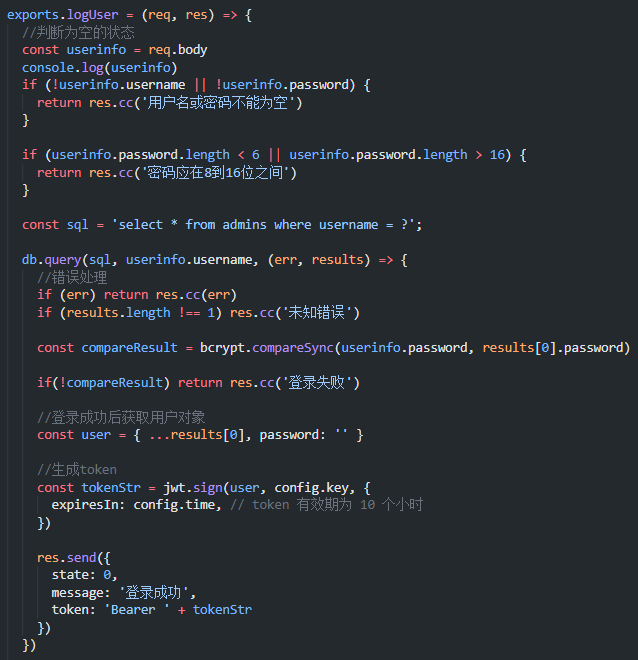
服务端开发时采用模块化的开发思路，即数据库中的每一个表都在服务端代码中体现成一个具体的路由模块，如下（router以及router\_handler）



每一个路由模块只处理相关的请求，例如关于user的请求



以及router\_handler中的对应user模块



使用模块化的开发思想，让服务器后期维护十分容易

除此之外，服务器中还使用了部分中间件（数据到达路由回调函数之前的处理函数，可实现类似于请求拦截器的功能）如



# 客户端设计

## 简介

## 项目结构解释