



成绩

中国农业大学

课程论文

(2022-2023 学年春季学期)

论文题目: 数据库综合实验设计大纲——车辆出租管理系统

课程名称: 数据库原理与实践实验

任课教师: 王庆、陈雷

班 级: 计算机 211、计算机 211

学 号: 2021307150609、2021321010112

姓 名: 王伟、姚攀峰

课程论文格式要求（封皮的背面）：

1. 课程论文采用统一封面，以左侧为准装订成册。

2. 课程论文一律使用标准 A4 复印纸打印或使用标准 A4 复印纸手写稿形式上交。

3. 论文打印的格式要求：

论文标题（使用隶书二号加黑；一级标题、二级标题、三级标题分别使用宋体三号、四号及小四号并加加黑）

摘要、关键字（需使用宋体小四号）

正文（使用宋体小四号，行距 20 磅）

参考文献（使用宋体五号）

数据库综合实验设计大纲—车辆出租管理系统

一、选题

简要描述题目的要求，应用的场景（200-300 字）

车辆出租管理系统：

车辆出租管理系统帮助各地门店管理员进行租车管理，便捷、高效为会员提供租车服务。为车辆管理员提供便捷操作，包括车辆管理、门店管理、门店车辆信息查询和订单查询等方面，采用 C#交互，界面干净整洁，便于车辆管理员管理系统信息；同时，会员可以不经过车辆管理员一体化进行查车、预约、提车、续约、还车等操作，省略了与管理沟通的中间步骤，直接与车辆出租系统直接交互，不仅节省了人力，而且快速实现租车服务，满足个人出行需要。

二、需求分析

2.1 数据库系统的业务描述（包括用户、主要功能、系统边界或限制）（500-1000 字）

车辆出租管理系统中的数据库系统主要承担着数据存储和管理的任务，是整个系统的核心。数据库系统的业务描述包括以下内容：

用户：车辆出租管理系统中的数据库系统主要面向管理员和用户，管理员可以通过数据库系统实现车辆信息、门店信息和客户信息的管理和订单查询等操作，客户可以通过数据库系统查车、预约、提车、续约、还车、订单查询等操作。

主要功能：数据库系统的主要功能包括车辆信息管理、门店信息管理、会员信息管理、订单管理、费用计算、出租/还车管理等。

1. 车辆信息管理：包括车辆的基本信息、车辆分类、车辆编号、车辆牌照、车辆出租价格等。
2. 门店信息管理：包括门店的基本信息、门店名称、门店编号等。
3. 会员信息管理：包括客户的基本信息、客户的账号和密码等。
4. 订单管理：包括会员编号、车辆编号、租车日期、还车日期、租车门店、还车门店、订单金额等。
4. 费用计算：包括定金缴纳、车辆出租时间和车辆出租价格以及是否逾期自动计算出租费用和应收金额。
5. 出租/还车管理：包括车辆的出租和归还，记录车辆出租和归还的时间、地点和是否逾期等。

系统边界或限制：数据库系统的边界主要是与其他模块的数据交互。例如：

- 1、会员或管理员登陆时，本系统防止了 SQL 注入，加强了数据库的安全性。
- 2、当会员未预约车时，无法进行取车和还车。
- 3、当会员已取车，不能再次预约车辆。

- 4、车辆不在门店时，无法修改或删除车辆。
 - 5、门店内有即将需要还车的门店或门店还有车的门店，无法删除门店。
 - 6、出租管理和还车管理模块需要与数据库系统交互以更新车辆状态和订单信息。
- 数据库系统的限制主要是数据安全性和数据一致性，本系统确保了数据的完整性、准确性和保密性。此外，本系统还考虑了系统的性能和稳定性，确保系统能够快速响应和稳定运行。

2.2 主要业务逻辑（或业务规则）（500 字左右）

会员：

会员先注册账户，然后登录账户，依次进行预约车辆、到店取车、续约车辆与归还车辆操作，不可以未预约车辆进行到店取车或续约车辆与归还车辆，也不可提取车辆之后，再次预约车辆，一个用户一次只能组一次车，违反此规定时均会有相应提示。

会员每次预约都是成功的，相当于为了线下提车更加方便（提前缴纳押金等操作可提前进行），会员在预约理想车辆时，预约的是该理想车型，并缴纳押金，直到取车时，提走相应车型的某辆车，如果改车没有，需要选择其他车型的车辆提车。

在车辆还未到截至日期时，可以进行续租操作，延迟归还时间，如果没有续租，那么结账时会把超出部分多支付款 50%。还车时，确认相应信息之后，缴纳费用即可完成还车操作，完成整个租车过程。会员也可以进行历史订单查询的功能，查询过往订单。

管理员：

管理员主要负责车辆信息、门店信息的管理和门店的车辆信息、订单查询。

车辆管理：分为车辆的增删改查，增车辆时，如果所增车辆为新车型，那么需要设置新的车型价格，并且置该车型的所有数量为 1，否则，无需设置新车价格，自动对该车型的所有数量+1；删除或修改车辆信息时，如果该车辆不在门店则无法删除或修改，提示报错信息；查询车辆时，可以通过下拉框和模糊查询进行多条件查询，便捷了管理员操作。

门店信息管理：同样分为门店的增删改查，增加、查询和修改没有额外限制，但删除门店时，需要当该门店无车需要还且该门店内没有车时，才可以删除门店。

2.3 数据字典（字数不限）

1. 数据项

数据项名：	uid	carid	storeid	adminid
数据项含义：	唯一标识会员身份的号码，包含会员的注册年月信息。	唯一标识车辆的号码	唯一标识门店的号码	唯一标识管理员的号码
别名：	会员编号	车辆编号	门店编号，提车 门店标号，还车 门店编号	管理员编号

定义:	会员编号 =U202001001 (文本型)	车辆编号 =C00003 (文本型)	门店编号 =S00003 (文本型)	管理员编号 =admin
位置:	t_user, t_user_order, t_car_rent	t_car_pla, t_car_rent, t_car_info, t_car_res, t_user_order	t_car_pla, t_car_rent, t_car_res, t_store_info, t_user_order	t_admin

2. 数据结构

会员:

数据结构名称:	res	rent	delay	return
含义说明:	车辆预约	提车	续约	还车
数据组成:	会员编号+车型+预计提车日期+预计还车日期+预计提车门店+预计还车门店+押金是否缴纳	会员编号+车辆编号+提车日期+提车门店+归还门店	预计归还日期+系统时间	车辆编号+提车日期+还车日期+提车门店+还车门店+订单金额

管理员:

数据结构名称:	车辆管理	门店管理
含义说明:	进行车辆信息的增删改查	进行门店信息的增删改查
数据组成:	车辆编号+车型信息+车牌号	门店编号+门店名称+门店地址+是否存在

3. 数据流

会员:

数据流名称:	车辆预约	到店提车	续约	还车
说明:	会员直接与系统进行预约操作	会员直接与系统进行提车操作	会员直接与系统进行续约操作	会员直接与系统进行还车操作
数据流来源:	会员	会员	会员	会员
数据流去向:	车辆出租管理系统	车辆出租管理系统	车辆出租管理系统	车辆出租管理系统
平均流量:	100 次/天	100 次/天	80 次/天	100 次/天
高峰流量:	300 次/天	300 次/天	200 次/天	300 次/天
数据组成:	会员编号+车型+	会员编号+车辆	预计归还日期+	车辆编号+提车

	预计提车日期+ 预计还车日期+ 预计提车门店+ 预计还车门店+ 押金是否缴纳	编号+提车日期+ 提车门店+归还 门店	系统时间	日期+还车日期+ 提车门店+还车 门店+订单金额
--	--	---------------------------	------	--------------------------------

管理员：

数据流名称：	车辆管理	门店管理
说明：	进行车辆信息的增删改查	进行门店信息的增删改查
数据流来源：	管理员	管理员
数据流去向：	车辆出租管理系统	车辆出租管理系统
平均流量：	5 次/天	1 次/天
高峰流量：	10 次/天	3 次/天
数据组成：	车辆编号+车型信息+车牌号	门店编号+门店名称+门店地址 +是否存在

4. 数据存储

数据存储名：	管理员信息表	车辆信息表	车辆所在门店表	车辆租车表
说明：	存放管理员的账户和密码	存放每辆车的具 体信息	存放每个门店的 车辆所有情况	提车之后产生的 租车表
编号：	t_admin	t_car_info	t_car_pla	t_car_rent
数据流来源：	车辆出租管理系统	管理员	管理员	会员
数据流去向：	车辆出租管理系统	车辆出租管理系统	车辆出租管理系统	车辆出租管理系统
数据组成：	管理员编号+管 理员密码	车辆编号+车辆 型号+牌照	车辆编号+门店 编号+现有数量	会员编号+车辆 编号+提车日期+ 提车门店+还车 门店
数据量：	基本为某一定 值，10	500 次/天	500 次/天	200 次/天
存取频率：	几乎为 0	10 次/天	100 次/天	100 次/天

数据存储名：	车辆预约表	门店信息表	车型信息表	会员信息表	账单表
说明：	存放会员预 约车辆的具	存放门店的具 体信息	存放不同车 型的信	存放会员的 具体信息	存放账单信息

	体信息				
编号:	t_car_res	t_store_info	t_type_info	t_user	t_user_order
数据流来源:	会员	管理员	管理员	会员	会员
数据流去向:	车辆出租管理系统	车辆出租管理系统	车辆出租管理系统	车辆出租管理系统	车辆出租管理系统
数据组成:	会员编号+车型+预计提车日期+预计还车日期+预计提车门店+预计还车门店+是否缴纳押金	门店编号+门店名称+门店地址+是否存在	车型+每日租车单价+现有数量	会员编号+会员姓名+注册日期+密码	会员编号+车辆编号+提车日期+还车日期+提车门店+还车门店+账单金额
数据量:	100 次/天	50 条	200 条	200 条	500 次/天
存取频率:	100 次/天	几乎为 0	几乎为 0	5 次/天	100 次/天

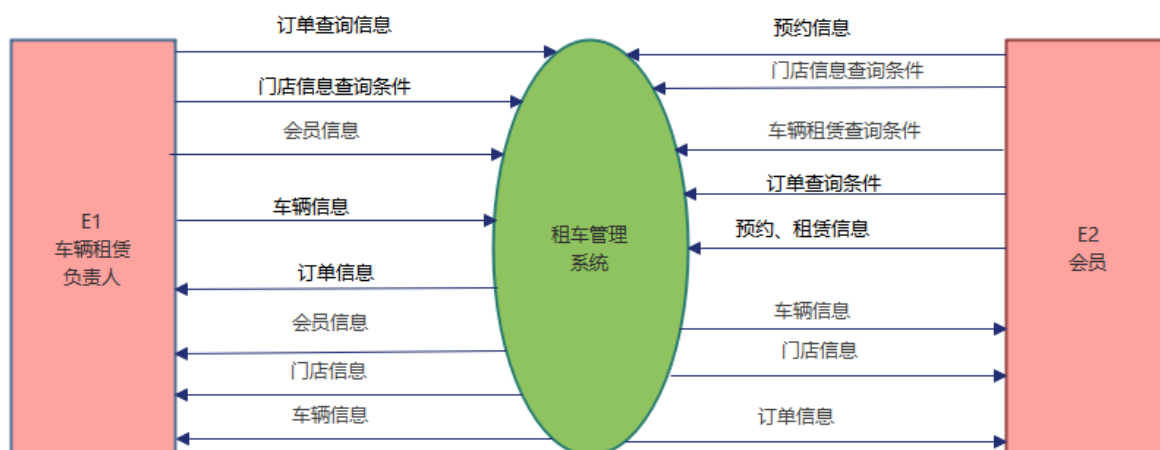
5. 处理过程

处理过程:	预约车辆	到店取车	续约	还车
说明:	找到会员心仪的车辆并预约	到店取车	续约	还车
输入:	会员的查询条件（车辆型号、取车门店、还车门店、提车日期、还车日期）	点击所预约车型	最终归还日期	账单金额
输出:	满足条件的车辆并预约	输出所需车型	将归还日期改写	生成账单信息
处理:	如果存在满足条件的车辆，那么显示，否则需要重新选择其他车辆。选择成功后，缴纳押金并预约，存入预约表。	如果存在预约时的车型，那么顺利提车，否则，需要选择其他车型提车。	如果未逾期，则顺利续约，否则，无法续约，还车时超出部分多付款 50%。	顺利缴纳账单金额之后，完成本次租车过程。

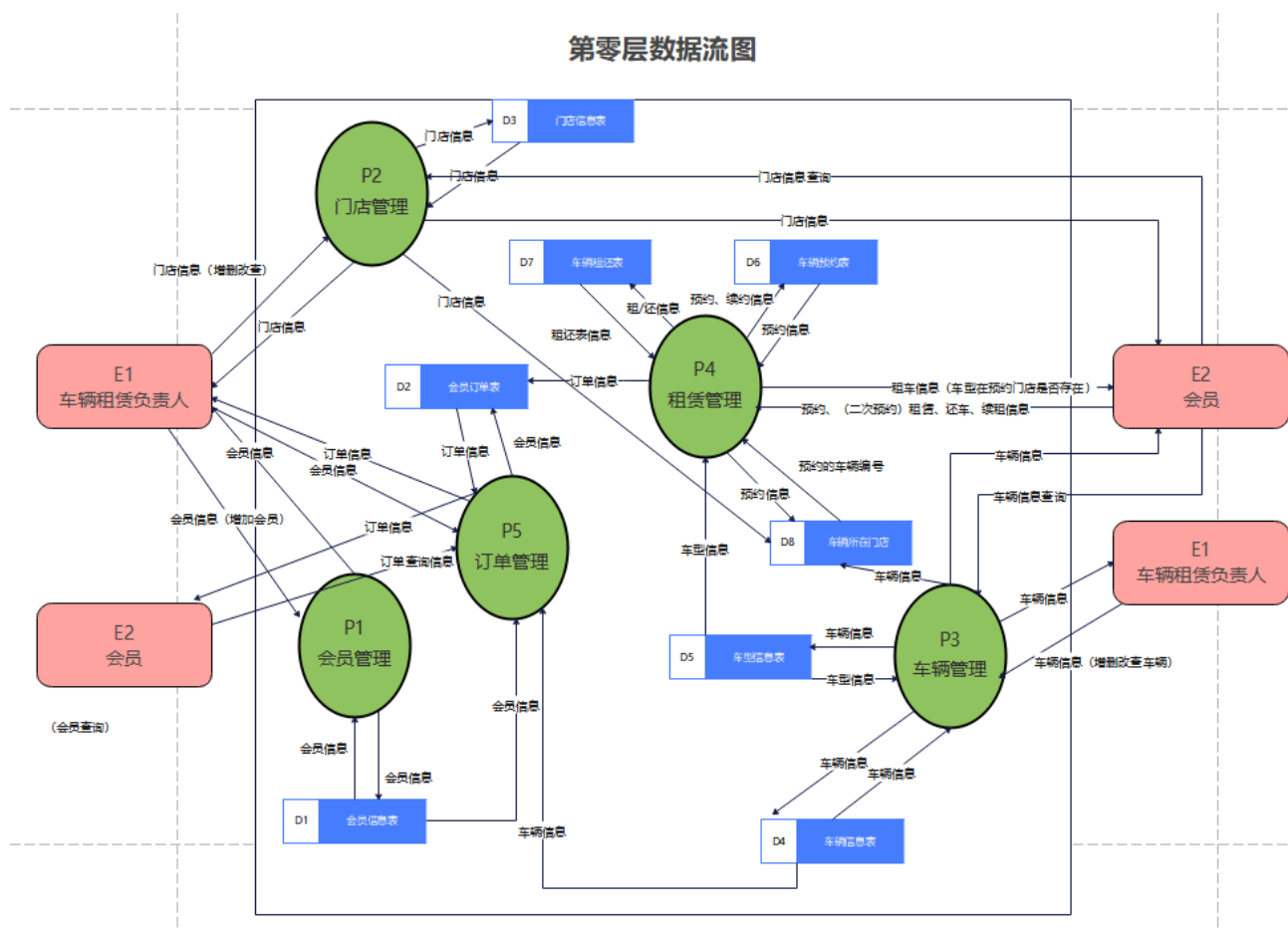
2.4 数据流图（字数不限）

按照应用情况，分析数据流情况，画出顶层、0 层、1 层等层的数据流图，并做必要描述。

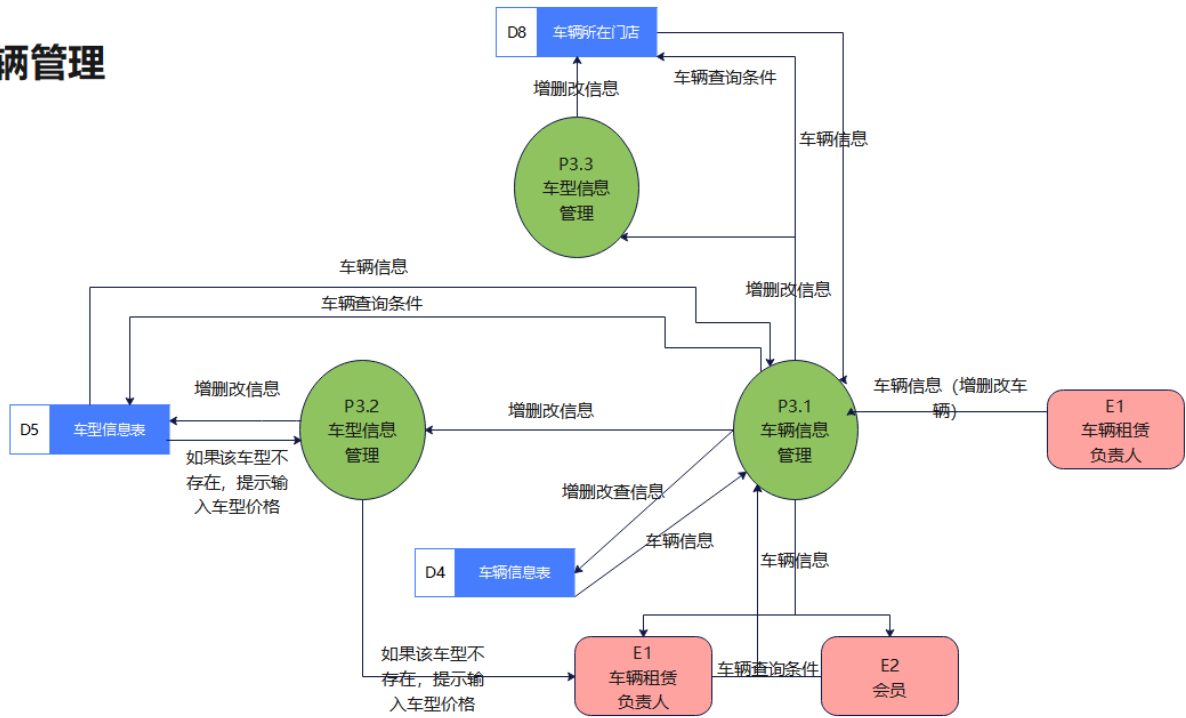
顶层数据流图



第零层数据流图

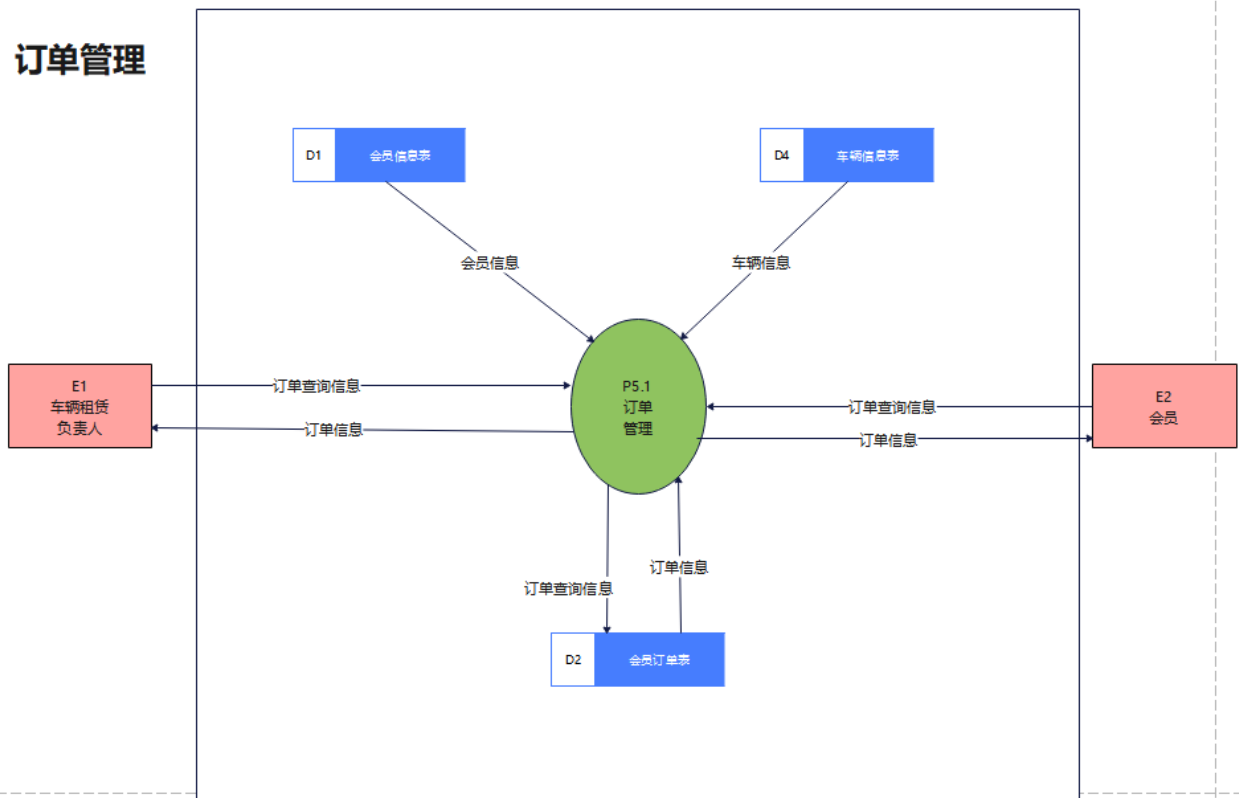


车辆管理

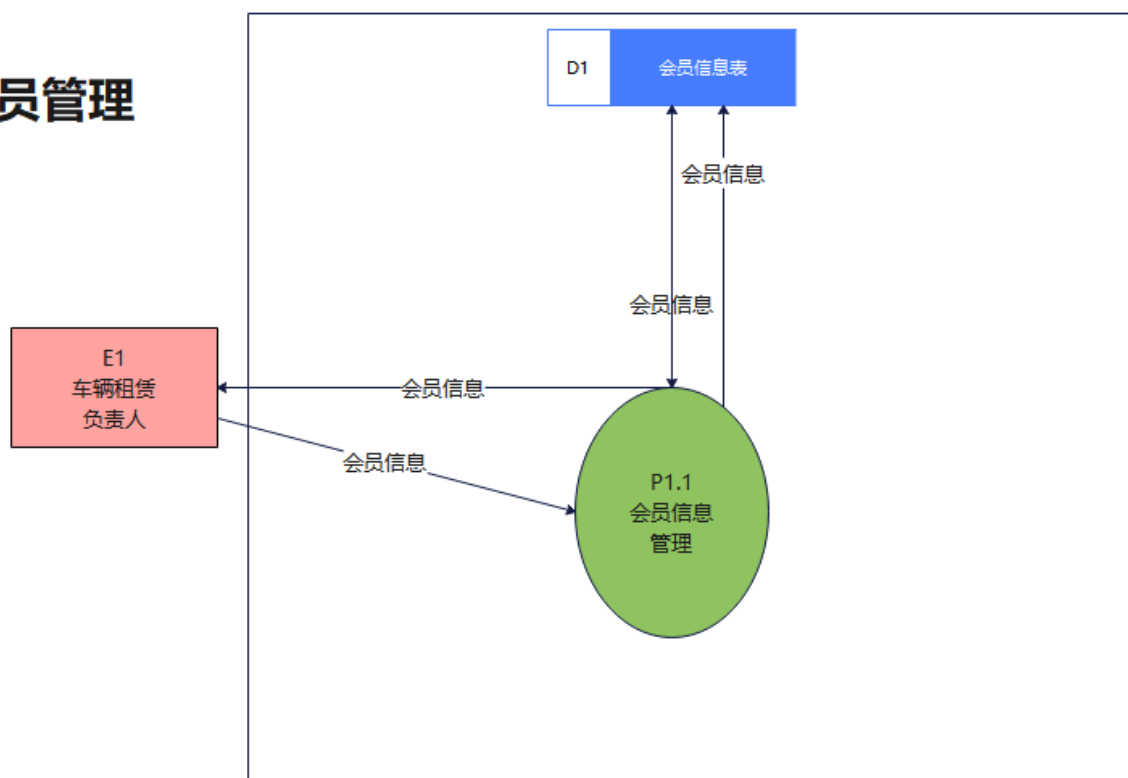


如果添加车型为新的车型，那么需要设置该车型的价格，反之，则不需要设置新车型价格。

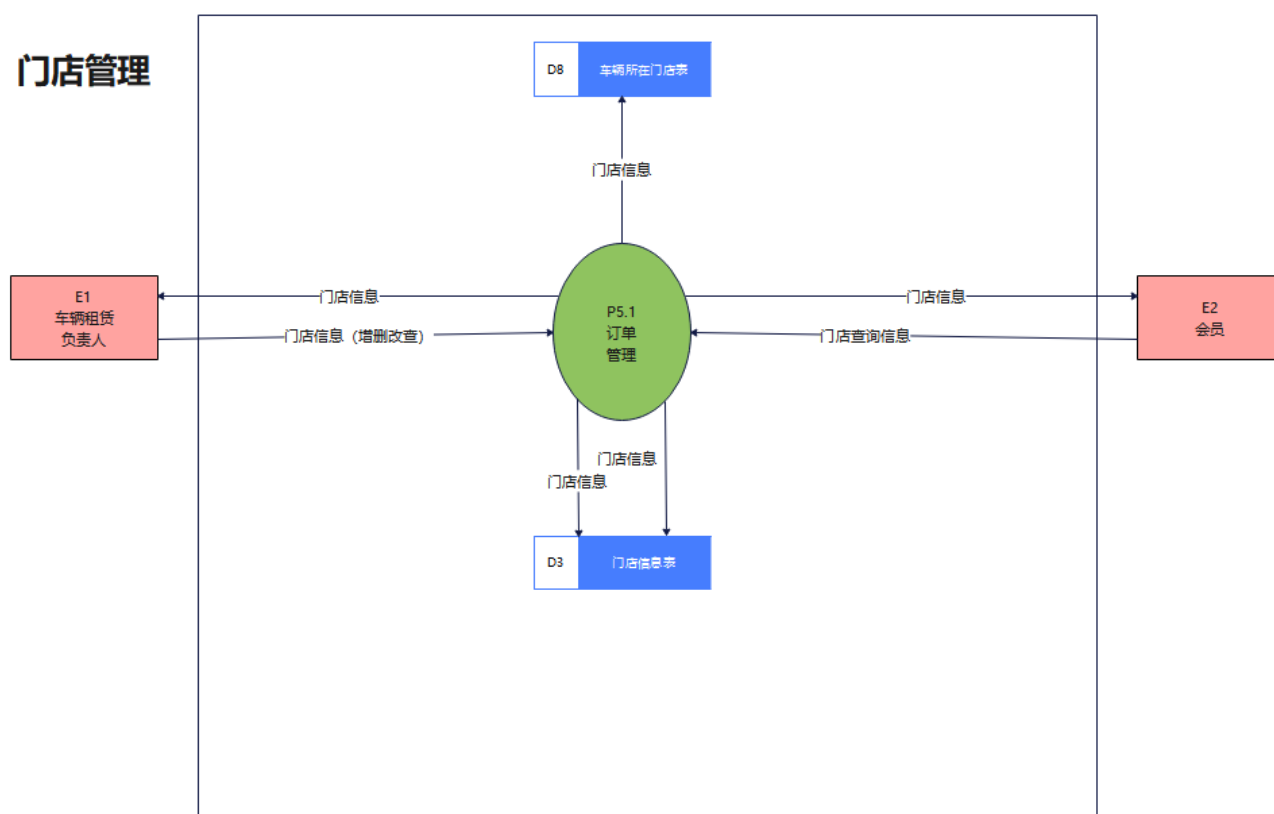
订单管理

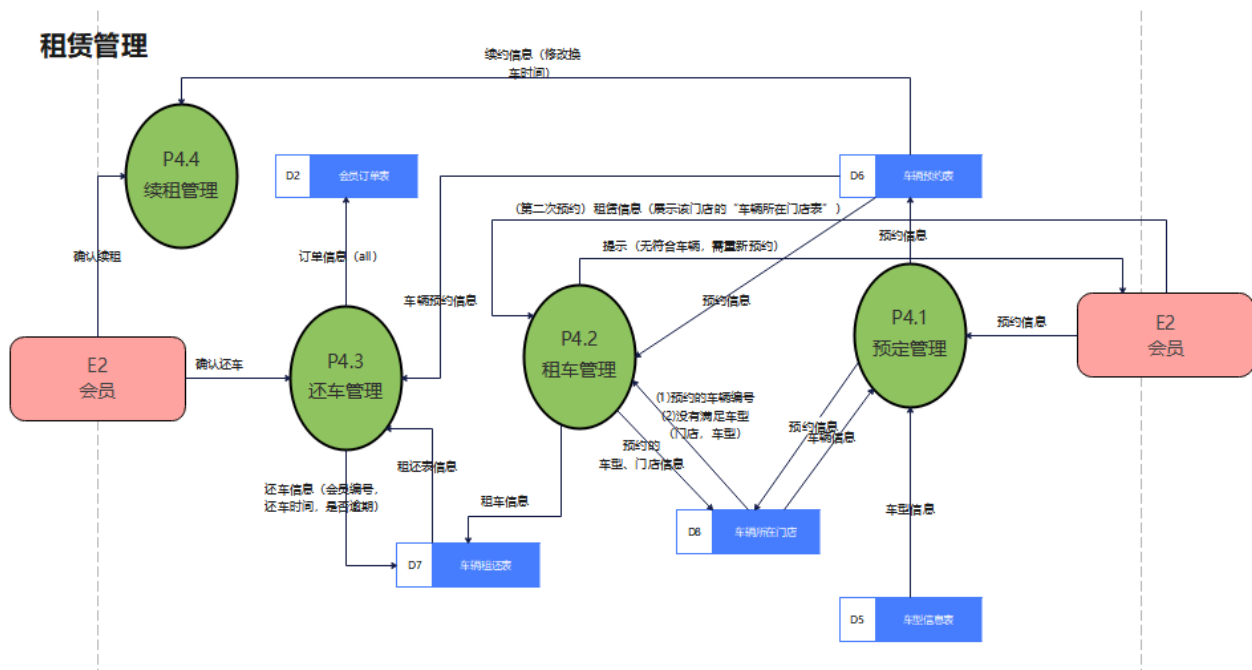


会员管理



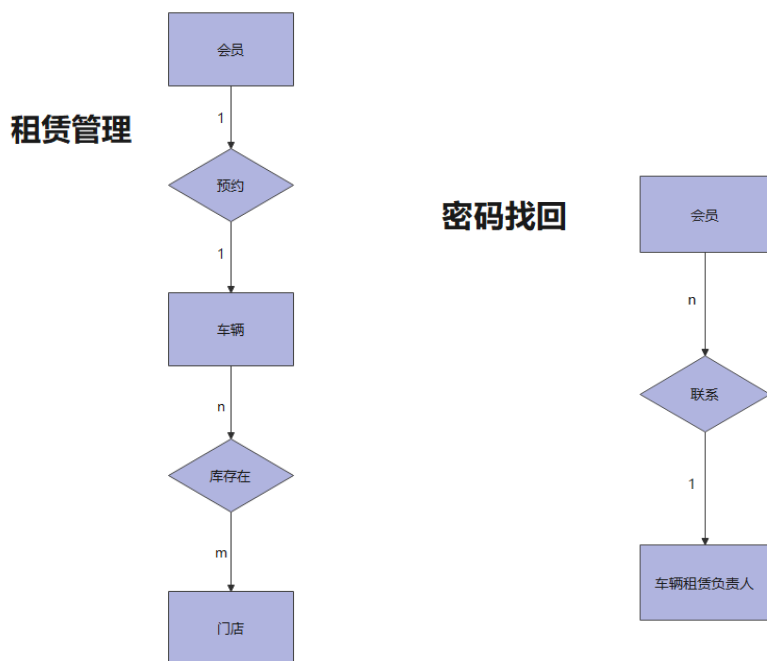
门店管理





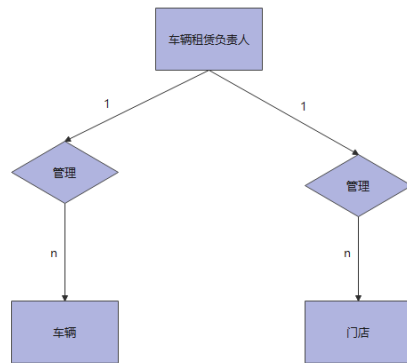
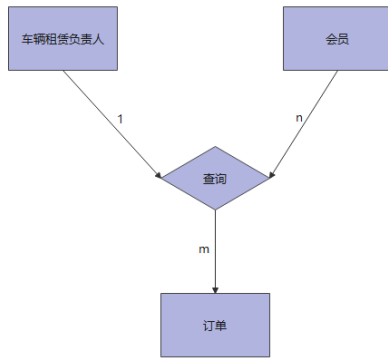
三、概念设计（字数不限）

3.1 ER 图（画局部 E-R 图，再合并画整体 E-R 图）

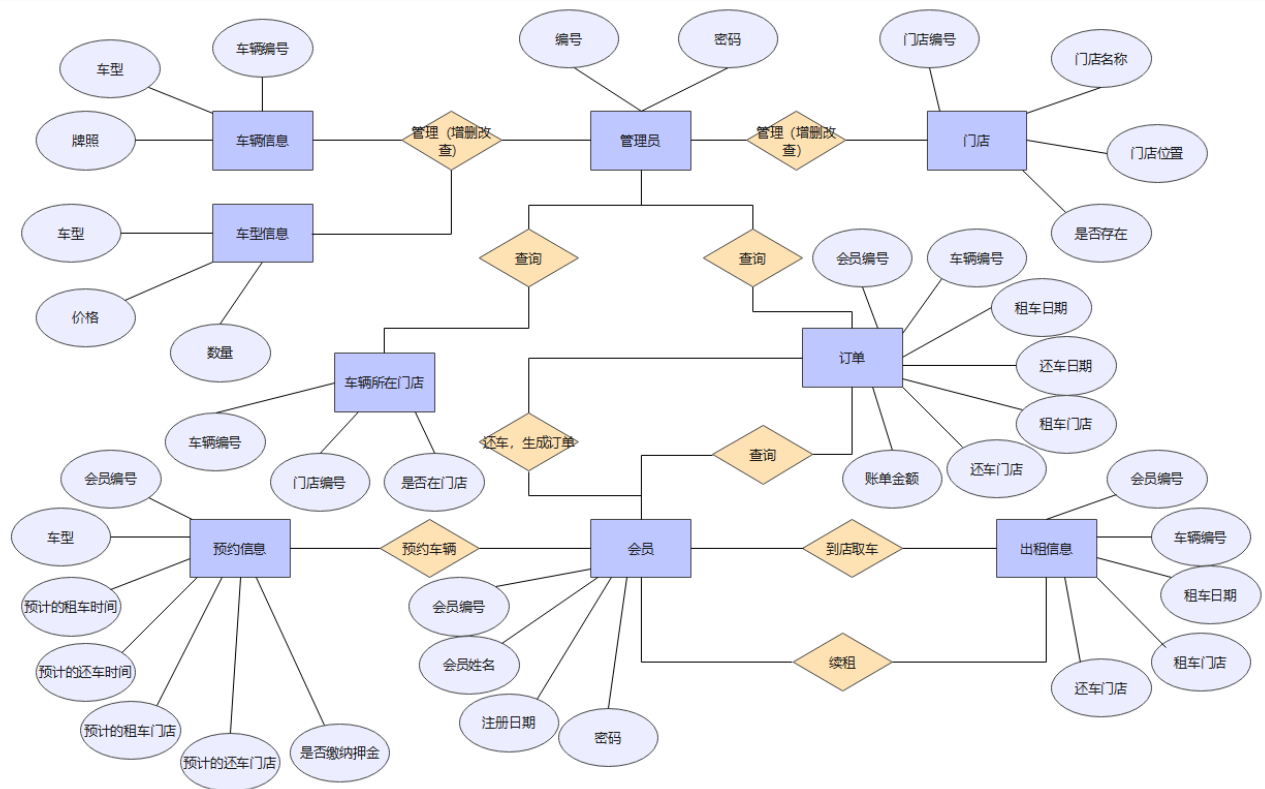


车辆和门店管理

订单查询



总的 ER 图



四、逻辑结构设计（字数不限）

4.1 数据库关系模式：

① 列出数据库中所有的关系模式，每个关系模式要求标出主码以及函数依赖关系，关系模式之间的主码外码关系。

数据库中关系模式如下：

a) 会员信息 t_user

(会员编号 uid 会员名字 uname 入会时间 udate 登陆密码 upsw)

主码为会员编号 uid

函数依赖关系: uid→uname, uid→update, uid→upsw

b) 车辆信息 t_car_info

(辆编号 carid 车型名称 cartype 车辆牌照 carlic)

主码为 carid

函数依赖关系: carid→cartype, carid→carlic

c) 车型信息 t_type_info

(车型名称 cartype 车型价格 rentprice 车型数量 type_num)

主码为 cartype

函数依赖关系: cartype→rentprice, cartype→type_num

d) 门店信息 t_store_info

(门店编号 storeid 门店名称 storename 门店地点 storepla

If_exists 判断当前门店是否存在)

主码为 storeid

函数依赖关系: storeid→storename, storeid→storepla, storeid→if_exists

e) 管理员登录表 t_admin

(管理员 id adminid 管理员密码 adminpsw)

主码为 id

函数依赖关系为: adminid→adminpsw

f) 会员订单: t_user_order

(会员编号 uid 车辆编号 carid 租车时间 rentdate 还车时间 returndate 租车门店 rentstore 还车门店 returnstore 支付金额 payment)

函数依赖关系为: (uid, carid, rentdate)→returndate, (uid, carid, rentdate)→rentstore,

(uid, carid, rentdate)→returnstore, (uid, carid, rentdate)→payment,

(uid, carid, returndate)→rentdate, (uid, carid, returndate)→rentstore,

(uid, carid, returndate)→returnstore, (uid, carid, returndate)→payment,

会员编号为外码参照 t_user 表的 uid

g) 车辆预约: t_car_res

(会员编号 uid 车型名称 cartype 租车时间 rentdate 还车时间 returndate 租车门店 rentstore 还车门店 returnstore 缴纳押金 paydepo)

函数依赖关系为: (uid, cartype, rentdate)→returndate,

(uid, cartype, rentdate)→rentstore, (uid, cartype, rentdate)→returnstore,

(uid, cartype, rentdate)→paydepo, (uid, cartype, returndate)→rentdate,

(uid, cartype, returndate)→rentstore, (uid, cartype, returndate)→returnstore,

(uid, cartype, returndate)→paydepo,

会员编号为外码参照 t_user 表的 uid

车辆型号为外码参照 t_type_info 表的 cartype

h) 车辆租还表: t_car_rent

(会员编号 uid 车辆编号 carid 租车时间 rentdate 租车门店 rentstore
还车门店 returnstore)

函数依赖关系为: (uid, carid, rentdate) → rentstore, (uid, carid, rentdate) → returnstore

会员编号为外码参照 t_user 表的 uid

i) 车辆所在门店: t_car_pla

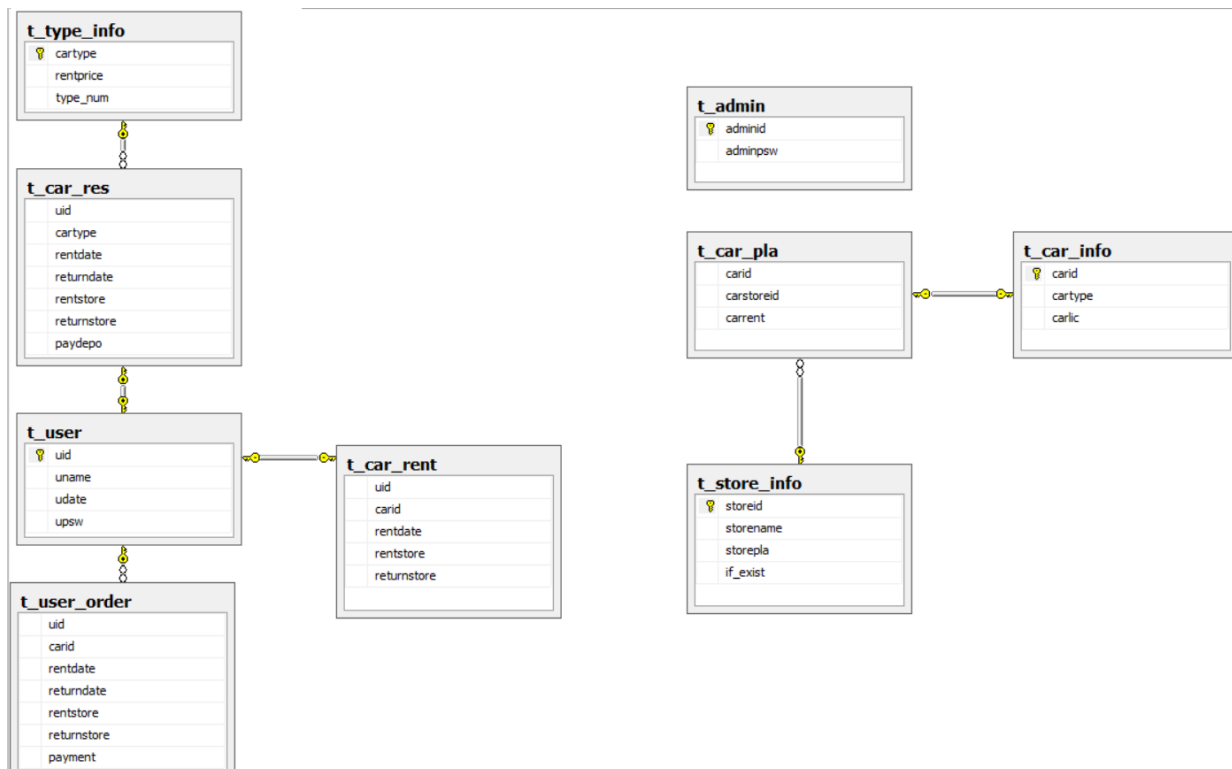
(车辆编号 carid 车辆所在门店编号 carstoreid 车辆是否出借 (BOOL) carrent)

函数依赖关系为: carid → carstoreid, carid → carrent, (carid, carstoreid) → carrent

车辆编号为外码参照表 t_car_info 表的 carid

车辆所在门店为外码参照表 t_store_info 表的 storeid

数据库整体关系如下:



② 每个关系模式满足的范式

a) 会员信息 t_user

所有决定因素都包含码, 属于 BCNF

b) 车辆信息 t_car_info

所有决定因素都包含码, 属于 BCNF

c) 车型信息 t_type_info

所有决定因素都包含码, 属于 BCNF

d) 门店信息 t_store_info

所有决定因素都包含码，属于 BCNF

e) 管理员登录表 t_admin

所有决定因素都包含码，属于 BCNF

f) 会员订单： t_user_order

没有非主属性对码的传递依赖，属于 3NF

g) 车辆预约： t_car_res

没有非主属性对码的传递依赖，属于 3NF

h) 车辆租还表： t_car_rent

没有非主属性对码的传递依赖，属于 3NF

i) 车辆所在门店： t_car_pla

没有非主属性对码的传递依赖，属于 3NF

③ 说明此时的每个关系基表，与数据流图中的数据存储的对应关系。

对应关系如下：

关系基表	数据流图中的数据存储
用户信息表	D1 会员信息表
车辆信息表	D4 车辆信息表
车型信息表	D5 车型信息表
门店信息表	D3 门店信息表
管理员登录表	-----
会员订单	D2 会员订单表
车辆预约表	D6 车辆预约表
车辆租还表	D7 车辆租还表
车辆所在门店表	D8 车辆所在门店表

④ 根据系统需求分析，给出主要的查询操作以及更新、插入、删除操作

系统界面部分：

主要是用户登录以及管理员登录进行的查询

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
select * from t_user where uid='U202001001';
select * from t_admin where adminid='admin';
```

根据 ID 进行用户信息表或者管理员信息表的查询。

注册界面设部分：

主要是添加用户信息到用户信息表的插入操作

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
insert into t_user(uid,uname,update,upsw) values('{label4.Text}','{textBox1.Text}','{DateTime.Now}','{textBox2.Text}');
```

管理员界面部分：

a) 订单查询部分：

涉及用户的订单查询

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
select uor.uid,t_user.uname,uor.rentdate,uor.returndate,st1.storename,st2.storename,t_car_info.cartype,uor.payment
from t_user_order uor,t_user,t_store_info st1,t_store_info st2,t_car_info
where t_user.uid=uor.uid and st1.storeid=uor.rentstore and st2.storeid=uor.returnstore and t_car_info.carid=uor.carid;
```

b) 门店车辆信息管理部分：

涉及门店车辆信息的查询

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
select t_car_pla.carstoreid,t_store_info.storename,t_store_info.storepla,t_car_pla.carid,t_car_info.cartype,
t_car_info.carlic,t_car_pla.carrent
from t_car_pla,t_store_info,t_car_info
where t_car_pla.carid=t_car_info.carid and t_car_pla.carstoreid=t_store_info.storeid;
```

c) 车辆管理部分：

车辆信息查询

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
select * from t_car_info
```

i. 车辆添加部分：

首先进行车辆信息表的添加

然后进行车型信息表相关车型信息加 1

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
insert into t_car_info values('{label5.Text}','{textBox2.Text}','{textBox3.Text}');
update t_type_info set type_num=type_num+1 where t_type_info.cartype='{textBox2.Text}';
```

ii. 车辆修改部分：

车辆牌照的修改

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
update t_car_info set carlic='{textBox3.Text}' where carid='{ID}'
```

iii. 车辆删除部分

在车辆信息表中删除当前车辆

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
delete from t_car_info where carid='{id}';
```

将车型信息表中的相关车型数量减 1

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
update t_type_info set type_num=type_num-1 from t_car_info,t_type_info
where type_num>0 and t_car_info.carid='{id}' and t_car_info.cartype=t_type_info.cartype;
```

d) 门店管理部分：

门店相关信息查询方面

```
SQLQuery3.sql - (I...\QUQ02\86155 (52))*
use car_rent;
select *from t_store_info where if_exist=1;
```

i. 门店添加部分:

门店的添加, 添加后的门店一定存在所以当前门店的 if_exist 一定是 1

```
SQLQuery3.sql - (I...\QUQ02\86155 (52))*
use car_rent;
insert into t_store_info values('{label4.Text}','{textBox2.Text}','{textBox3.Text}','1')
```

ii. 门店修改部分:

门店名称的修改

```
SQLQuery3.sql - (I...\QUQ02\86155 (52))*
use car_rent;
update t_store_info set storename='{textBox2.Text}',storepla='{textBox3.Text}' where storeid='{ID}';|
```

iii. 门店的删除

```
SQLQuery3.sql - (I...\QUQ02\86155 (52))*
use car_rent;
update t_store_info set if_exist = 0 where storeid = '{id}';
```

设置门店 if_exist 属性为 0

用户界面部分:

a) 订单查询部分:

用户订单查询 (只能查询自己的, 所以相较于上边的管理员订单查询多了用户 ID 的条件)

```
SQLQuery3.sql - (I...\QUQ02\86155 (52))*
use car_rent;
select uor.uid,t_user.uname,uor.rentdate,uor.returndate,st1.storename,st2.storename,t_car_info.carttype,uor.payment
from t_user_order uor,t_user,t_store_info st1,t_store_info st2,t_car_info
where t_user.uid=uor.uid and st1.storeid=uor.rentstore and st2.storeid=uor.returnstore and t_car_info.carid=uor.carid
and t_user.uid='{Data.UID}';
```

b) 到店取车部分:

预约信息的查询

```
SQLQuery3.sql - (I...\QUQ02\86155 (52))*
use car_rent;
select cartype,rentdate,returndate,rentstore,returnstore,paydepo from t_car_res where uid = '{ID}';|
```

租车信息的添加

```
SQLQuery3.sql - (I...\QUQ02\86155 (52))*
use car_rent;
insert into t_car_rent(uid,carid,rentdate,rentstore,returnstore) values('{ID}','{selectedcarid}','{DateTime.Now}','{rent_s}','{return_s}');
```

c) 车辆归还与续租部分:

首先对租车情况进行查询

```
SQLQuery3.sql - (I...\QUQ02\86155 (52))*
use car_rent;
select rent.carid,car.cartype,rent.rentdate,res.returndate,s1.storepla,s2.storepla,uer.uname
from t_car_res res,t_car_rent rent,t_store_info s1,t_store_info s2,t_car_info car, t_user uer
where rent.uid = '{ID}' and rent.uid = res.uid and car.carid = rent.carid and s1.storeid = rent.rentstore and s2.storeid = res.returnstore
and rent.carid = car.carid and uer.uid = rent.uid;
```

续约存储过程的执行

```
SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
exec xu_yue @id = '{ID}',@t_date = 'return_date';
```

续约存储过程包含一个更新操作

```
GO
CREATE proc xu_yue
    @id VARCHAR(45),
    @t_date DATE
as
BEGIN
UPDATE t_car_res
SET t_car_res.returndate = @t_date
WHERE uid = @id;
END
GO
```

还车存储过程的执行:

```
建表(无多余列).sql - (...UQ02\86155 (53)) SQLQuery3.sql - (I...QUQ02\86155 (52))*
use car_rent;
exec return_car @id = '{ID}',@r_date = 'return_date',@payment = payment,@over = 1;
```

还车存储过程

包含 4 个查询操作以及一个更新操作

```
GO
CREATE proc return_car
    @id VARCHAR(45),
    @r_date DATE,
    @payment FLOAT,
    @over TINYINT
as
BEGIN
/*删除预约表数据*/
DELETE FROM t_car_res WHERE uid = @id;
/*添加用户订单数据*/
DECLARE @car_id VARCHAR(45);
DECLARE @rent_d DATE;
DECLARE @rent_s VARCHAR(45);
DECLARE @return_s VARCHAR(45);
select @car_id = carid from t_car_rent where uid = @id;
select @rent_d = rentdate from t_car_rent where uid = @id;
select @rent_s = rentstore from t_car_rent where uid = @id;
select @return_s = returnstore from t_car_rent where uid = @id;
insert into t_user_order values(@id,@car_id,@rent_d,@r_date,@rent_s,@return_s,@payment);
DELETE FROM t_car_rent WHERE uid = @id;
END
GO
```

d) 车辆预约部分:

查询当前没有出租的车辆

```
建表(无多余列).sql - (...UQ02\86155 (53)) SQLQuery3.sql - (...QUQ02\86155 (52))*
use car_rent;
select car.carid,car.cartype,car.carlic,store.storepla from t_car_info car,t_car_pla pla,t_store_info store
where car.carid = pla.carid and store.storeid = pla.carstoreid and pla.carrent = 0
```

调用预约的存储过程

```
建表(无多余列).sql - (...UQ02\86155 (53)) SQLQuery3.sql - (...QUQ02\86155 (52))*
use car_rent;
exec res @id = 'ID',@type = 'type',@rent_d = 'rent_date',@return_d = 'return_date',@rent_s = 'Dkentstore',@return_s = 'returnstore';
```

预约存储过程包含

两个查询操作以及一个添加操作

```
GO
CREATE proc res
@id VARCHAR(45),
@type VARCHAR(45),
@rent_d DATE,
@return_d DATE,
@rent_s VARCHAR(100),
@return_s VARCHAR(100)
as
BEGIN
DECLARE @rent_id VARCHAR(45);
DECLARE @return_id VARCHAR(45);
select @rent_id = storeid FROM t_store_info where storepla = @rent_s;
select @return_id = storeid FROM t_store_info where storepla = @return_s;
insert into t_car_res values(@id,@type,@rent_d,@return_d,@rent_id,@return_id,1);
END
GO
```

4.2 应用系统结构设计（系统模块构成；每个模块的输入和输出 IPO 表以及功能说明）



功能说明：dataGridView 显示所有车辆的情况，管理员点击 button 进行车辆选择、操作选择

然后进行车辆操作，最后将操作显示在 dataGridView 中同时也存储到数据库中。



功能说明：dataGridView 显示所有门店的情况（这里的门店不显示不存在的门店（通过 if_exist 属性可以完成）），管理员点击 button 进行门店选择、操作选择然后进行门店操作，最后将操作显示在 dataGridView 中同时也存储到数据库中。



功能说明：dataGridView 显示所有订单的情况，同时管理员可以进行门店、车型以及日期条件的查询来找到符合自己要求的订单，并且支持筛选条件的清除。



功能说明：dataGridView 显示门店的车辆情况，管理员可以根据自身的查询条件选择要查询的门店名称或者门店地址完成信息的筛选，筛选后的结果同样显示在 dataGridView 表中。



功能说明：dataGridView 显示当前用户（通过用户 ID 完成查询）所有订单的情况，同时用户可以进行门店、车型以及日期条件的查询来找到符合自己要求的订单，并且支持筛选条件的清除。



功能说明：dataGridView 显示当前可以进行预约的车型（即当前没有出租的车型），用户根据自身情况可以进行车型的选择、取车门店的选择，选择完成之后进行还车门店以及日期的选择，最后完成取车，并将结果存储到数据库。



功能说明：根据用户的预约信息，dataGridView 显示当前门店符合用户要求的车型，然后用户进行相关车型的选择，完成取车操作，将结果存储到数据库。



功能说明：根据用户的租车信息，label 显示当前用户的租车情况，用户进行还车/续租操作的选择，当进行还车操作时，用户缴纳相应费用后完成数据存储。当进行续租操作时，用户根据需求选择续租时间，然后进行数据更改。

五、物理结构设计（字数不限）

5.1 查询事件和更新事件的频度分析（模拟一下真实情况）

① 首先进行日志收集

```
SQLQuery1.sql - (L...QUQ02\86155 (54))
SELECT TOP 1000
    ST.text AS '执行的SQL语句',
    QS.execution_count AS '执行次数',
    QS.total_elapsed_time AS '耗时',
    QS.total_logical_reads AS '逻辑读取次数',
    QS.total_logical_writes AS '逻辑写入次数',
    QS.total_physical_reads AS '物理读取次数',
    QS.creation_time AS '执行时间',
    QS.*
FROM sys.dm_exec_query_stats QS
CROSS APPLY
    sys.dm_exec_sql_text(QS.sql_handle) ST
WHERE QS.creation_time > '2020-08-18'
ORDER BY
    QS.total_elapsed_time DESC;
```

在 SQL SERVER 数据库管理界面中运行如下语言，查看最近执行的数据库操作，可以明显的看出数据库执行的操作以查询操作为主。

部分操作查询结果如下：

执行的SQL语句	执行次数	耗时	逻辑读取次数	逻辑写入次数	物理读取次数	执行时间	sql_handle	statement_start_offset	statement_end_offset
31 select c.name,c.description from sys.schemas c where c.id...	9	303	36	0	0	2023-06-22 15:41:11.920	0x02000000f6e8b33509ca53048c6daeef741c78262f507cef	0	-1
32 use [master] SELECT (select default_schema_name from sys...	1	302	4	0	1	2023-06-22 15:44:38.080	0x02000000edf807326c0f49d418db45efcdafad35d19307	26	-1
33 SELECT tr.name AS [Name], tr.object_id AS [ID], CASE WH...	1	294	5	0	2	2023-06-22 15:44:38.177	0x0200000005232560869aefcccd57fcd071405ca0a8930ff47	0	-1
34 SELECT s.name AS [Name], 'Server([Name])' + quotename(...	1	211	62	0	1	2023-06-22 15:40:55.863	0x0200000006a9d93118a240847621c7c7e06198389c6487f02	0	-1
35 SELECT s.name AS [Name] FROM sys.schemas AS s ORDE...	1	165	2	0	1	2023-06-22 15:44:38.060	0x0200000011434331ce84fd36963c6409070ff64088133e5a	0	-1
36 SELECT s.name AS [Name] FROM sys.schemas AS s ORDE...	1	144	2	0	1	2023-06-22 15:44:38.103	0x0200000011434331ce84fd36963c6409070ff64088133e5a	0	-1
37 use [msdb] SELECT (select default_schema_name from sys...	1	144	4	0	1	2023-06-22 15:44:38.100	0x02000000063ea3f06d84283bd31a2e4a43272299ff48649d1	22	-1
38 SELECT s.name AS [Name] FROM sys.schemas AS s ORDE...	1	137	2	0	1	2023-06-22 15:44:38.093	0x0200000011434331ce84fd36963c6409070ff64088133e5a	0	-1
39 SELECT s.name AS [Name] FROM sys.schemas AS s ORDE...	1	130	2	0	1	2023-06-22 15:44:38.130	0x0200000011434331ce84fd36963c6409070ff64088133e5a	0	-1
40 create table #tmp_uniquetagsycpackages (name synam...	1	122	0	0	0	2023-06-22 15:42:27.363	0x02000000089c25f190e3080e27e1c5ef0903a1eabcc30eef	168	386
41 SELECT s.name AS [Name] FROM sys.schemas AS s ORDE...	1	121	2	0	1	2023-06-22 15:44:38.120	0x0200000011434331ce84fd36963c6409070ff64088133e5a	0	-1
42 SELECT SCHEMA_NAME(tbl.schema_id) AS [Schema], tbl.n...	1	98	17	0	0	2023-06-22 15:44:38.210	0x020000000c150e3254f4832d858a2f203700fc3873a57fdoc	0	-1
43 declare @pub_name sysname select @pub_name = null ...	1	80	0	0	0	2023-06-22 15:41:08.330	0x020000000ec49f3689d4c29e68e98a58c9f8f39523898754	2318	3878
44 use [model] SELECT (select default_schema_name from sys...	1	78	4	0	0	2023-06-22 15:44:38.087	0x02000000020828e0747248883410dfeec9d1202937d8676d2	24	-1
45 (@_msparam_0 nvarchar(4000),@_msparam_1 nvarchar(400...	2	74	4	0	0	2023-06-22 15:38:55.553	0x020000000d5be1837aa4aff08e2d04f9c4e93df8ab984e1b	286	762
46 DECLARE @ts_policy_automation_enabled bit S...	1	68	3	0	0	2023-06-22 15:41:05.667	0x020000000f8b89d3a9b11d97156e278ff64f35f4e39f3c67	600	-1
47 SELECT db.name AS [Name], db.database_id AS [ID], CAS...	1	65	20	0	0	2023-06-22 15:44:38.137	0x020000000804ad300471b828770a49c18e082a5edbe936656	0	-1
48 SELECT tr.name AS [Name], tr.object_id AS [ID], CASE WH...	1	54	3	0	0	2023-06-22 15:44:38.160	0x02000000f73c93383ec45c9f8c6e9e5299c3b35c8b4bd492	0	-1

② 模拟真实情况的查询以及更新事件分析

针对我们设计的车辆出租管理系统，模拟用户使用系统的整个周期，考虑正常情况（用户可以正常选取自己的预约车型），用户进行车辆续租与不进行车辆续租的比例为 1：3，进行查询事件以及更新事件的频度统计。查询事件以及更新事件的频度计算公式如下：

$$\begin{cases} t_{query} = \frac{(t_{query1} + t_{query2} * 3)}{4} \\ t_{update} = \frac{(t_{update1} + t_{update2} * 3)}{4} \end{cases}$$

首先在进入相关用户操作前，会有操作条件的判断，进入预约操作界面之前要查询用户是否已经预约车辆，还车阶段要查询是否取车，取车阶段查询是否有预约以及是否已经取车，即进行 4 次查询。

- 预约车辆：**首先 DataGridView 通过没有多余条件的查询查询出各个门店所拥有的未出租车辆，然后根据用户使用下拉框的选择（可以对车辆型号以及取车门店进行选择）查询当前门店所拥有的并未出租的车辆信息，系统会根据下拉框选项对左边的 DataGridView 所显示的数据进行实时更新，同时用户也可直接点击 DataGridView 中的信息完成车龄型号以及取车门店的选择，在车辆符号正常租车公司数量的条件下，用户为了操作的方便以及快捷，一般需要进行两次下拉框的点击，即进行两次查询，当一切选择就绪，用户可以进行车辆的预约，车辆预约之后会将相应的信息填写进车辆预约表 t_car_res。

即本次涉及 3 个查询时间的执行以及 1 个插入事件的执行。

- 到店取车：**取车阶段，首先进行用户预约情况的查询及显示，即查询车辆类型、租车日期、预计还车日期、租车门店、还车门店以及已经缴纳的费用，涉及一次通过表的连接进行的查询（查询条件是用户的 ID），选择车辆后将信息插入车辆租赁表 t_car_rent。

基本词操作涉及 1 次查询事件以及一次插入事件。

- 车辆续租：**续租阶段，首先进行用户租车情况的查询及显示，涉及一次通过表的连接进行的查询（查询条件是用户的 ID），当进行续约时，用户选择续约的日期，然后对车辆租赁表 t_car_rent 进行更新，涉及一次更新操作。

- d) 车辆归还：与续租操作类似，首先进行一次租车情况查询，然后通过存储过程进行车辆的归还，存储过程中涉及 4 次查询操作以及 1 次插入操作，一次删除操作。
- e) 订单查询：使用下拉框对用户需要查询的订单进行查询，用户可以直接按照需求对条件进行选择然后点击“查询”按钮进行查询一般需要一次查询。

根据上述查询、更新事件的频度公式以及对系统的业务逻辑中数据库操作的分析，可得：

$$\begin{cases} t_{query} = \frac{(10 + 9 * 3)}{4} = 9.25 \\ t_{update} = \frac{(5 + 4 * 3)}{4} = 4.25 \end{cases}$$

即用户在整个预约车辆、取车、归还车辆的过程中涉及的查询操作大约有 9.25 次，更新操作（包含插入操作以及删除操作）大约有 4.25 次

根据每天的最高流量（300 次每天）和平均流量（100 次每天）的流量预计，每天平均查询次数为 925 次，最高查询次数为 2775 次，平均更新次数为 425 次，最高更新次数为 1275 次。

5.2 表索引的建立（根据频度和性能需求，设计各个表的索引）

根据上述操作数量的统计，对车辆预约表 t_car_res、车辆租赁表 t_car_rent 的 uid 属性创建唯一非聚簇索引

```
-- 创建唯一非聚集索引
create unique nonclustered          --表示创建唯一非聚集索引
index index_car_res                 --索引名称
on t_car_res(uid)                  --数据表名称（建立索引的列名）
with
(
    ignore_dup_key=on,              --表示向唯一索引插入重复值会忽略重复值
    statistics_norecompute=off      --表示启用统计信息自动更新功能
)

-- 创建唯一非聚集索引
create unique nonclustered          --表示创建唯一非聚集索引
index index_car_rent               --索引名称
on t_car_rent(uid)                 --数据表名称（建立索引的列名）
with
(
    ignore_dup_key=on,              --表示向唯一索引插入重复值会忽略重复值
    statistics_norecompute=off      --表示启用统计信息自动更新功能
)
```

对车辆所在门店表 t_car_pla 的 carid 字段创建唯一非聚簇索引，carstroeid 属性创建普通索引。

```

-- 创建唯一非聚集索引
create unique nonclustered          --表示创建唯一非聚集索引
index index_pla_carid              --索引名称
on t_car_pla(carid)                --数据表名称（建立索引的列名）
with
(
    ignore_dup_key=on,              --表示向唯一索引插入重复值会忽略重复值
    statistics_norecompute=off      --表示启用统计信息自动更新功能
)

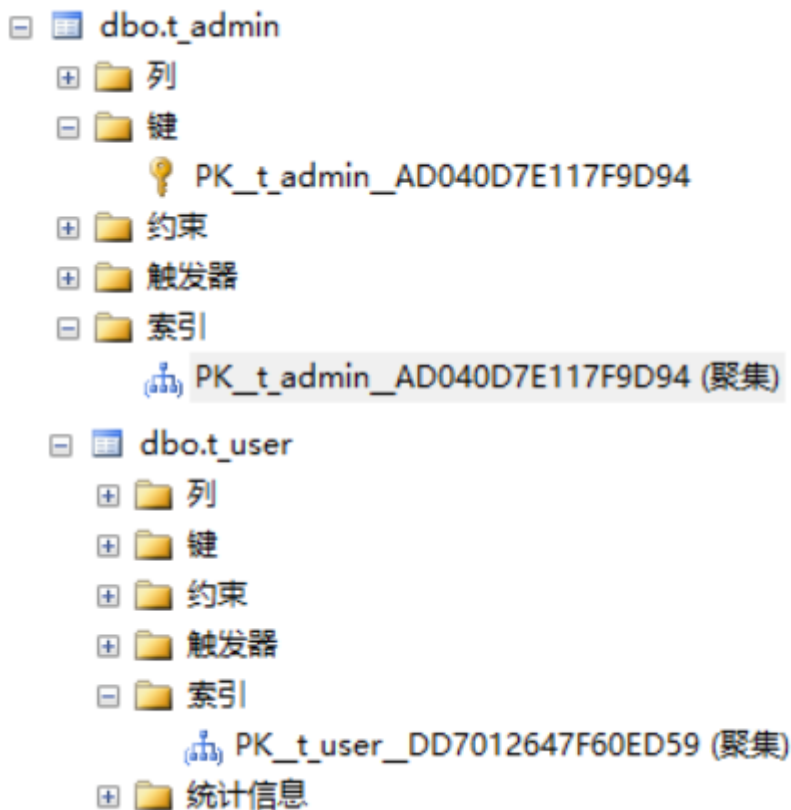
-- 创建索引
CREATE INDEX Index_pla_store
ON t_car_pla (carstoreid)

```

用于加快查询操作的速度。

5.3 聚簇（如果有需要的话则描述，否则可省略）

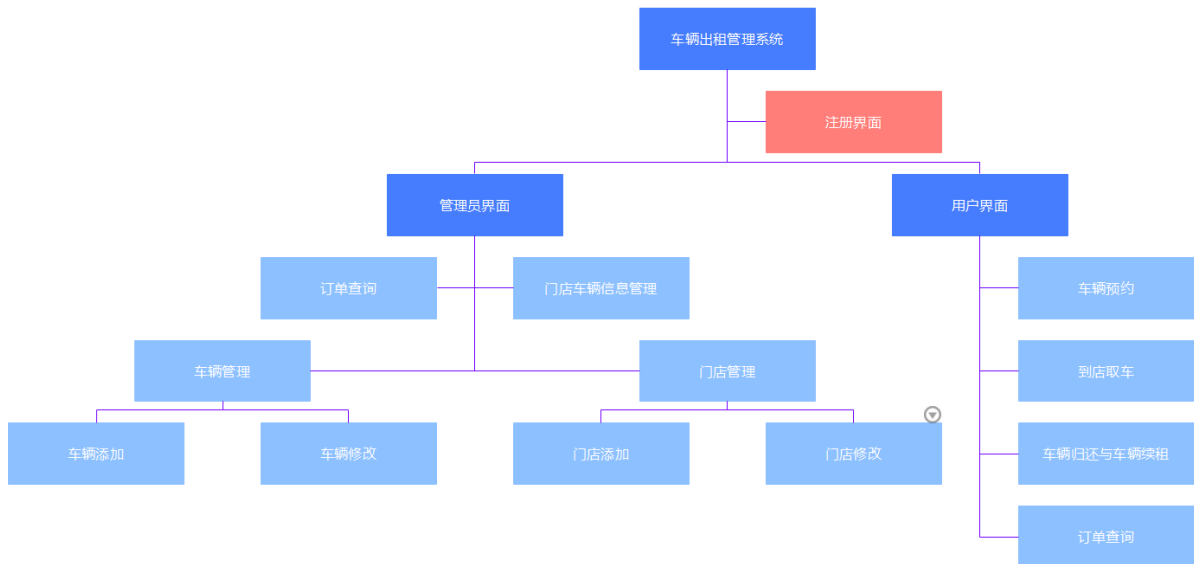
SQL Server 对于已经创建的包含主键的表创建了聚簇索引，下面是对部分表聚簇索引的显示：



六、界面设计（字数不限）

不要截图，设计阶段是对界面的组成和布局的设计。例如一共有几张页面（页面之间的层级关系，可以用树型图表示），每个页面的布局（可以手画或电脑画）

页面的层次关系如下：



下面分别对各个界面进行描述：

系统界面设计：

界面包括用户登录、管理员登录以及会员注册、密码找回四部分。用户和管理员通过输入账号和密码分别进入用户界面和管理员界面。

① 注册界面设计：

用户通过系统中的注册按钮进入注册界面，系统编号以及用户编号为系统生成，用户只需输入用户命以及密码即可完成会员注册。

register

用户编号: label4

当前时间为: label6

输入您的用户名:

输入密码:

确认密码:

注册 返回

② 管理员界面设计:



车辆管理员界面，设计菜单界面 menuStrip 实现管理员对功能（功能包括：车辆管理、门店管理、门店的车辆信息管理以及订单查询）的选择。

a) 订单查询界面设计:

点击订单查询，进入下面界面，可以通过输入会员 ID，会员姓名（支持模糊查询），选择租车日期、还车日期（通过 monthCalendar 进行选择），下拉框选择借车门店、还车门店和车辆类型完成订单查询。同时支持点击清空查询条件。

会员编号	会员姓名	借车日期	还车日期	借车门店	还车门店	车辆类型	账单金额
------	------	------	------	------	------	------	------

会员编号:

会员姓名:

借车日期: ▼

还车日期: ▼

借车门店: ▼

还车门店: ▼

车辆类型: ▼

b) 门店车辆信息管理设计:

门店编号	门店名称	门店地址	车辆编号	车辆型号	车牌号	车辆是否租赁
------	------	------	------	------	-----	--------

门店编号:

门店名称: ▼

门店地址: ▼

包括 dataGridView 显示查询所得数据, 以及下拉框显示门店名称以及门店地址, 同时支持输入门店编号进行车辆信息查询, 支持清空查询条件。

c) 车辆管理界面设计:

车辆管理界面，包含左侧的 dataGridView 显示车辆信息（包括车辆编号、车辆型号、车牌号），右侧按钮进行操作选择，以及支持车辆号、下拉框显示车型，车牌号对相关车辆进行查询，同时可以进行相关查询条件的清除。当点击添加车辆或者修改车辆时，会出现下面界面。

i. 车辆添加界面设计：

添加车辆自动生成车辆的编号，然后输入车辆的型号以及车辆的车牌号，当不存在当前车型时，需要进行车型价格的设定。

ii. 车辆修改界面设计：

修改车辆

车辆编号: label4

车辆型号: label5

车牌号:

修改车辆 返回

在车辆信息管理界面选中相关车辆，然后进入当前界面自动显示车辆的编号以及型号，管理人员可以进行车辆车牌号的修改。

d) 门店管理界面设计：

门店管理界面，包含左侧的 dataGridView 显示门店信息（包括门店编号，门店名称以及门店地址），右侧按钮进行操作选择，以及支持门店号、门店地址、门店名称对相关门店进行查询，同时可以进行相关查询条件的清除。当点击添加门店或者修改门店时，会出现下面界面。

门店管理页面

门店编号	门店名称	门店地址
------	------	------

当前选中的门店是:
null

添加门店 修改门店

删除门店 刷新

条件查询:

门店号:

门店名称:

门店地址:

查询 清空筛选条件

返回

i. 门店添加界面设计：

添加门店自动生成门店的编号，然后输入门店的名称以及门店的地址，点击按钮添加门店完成门店的添加。



门店编号: label4

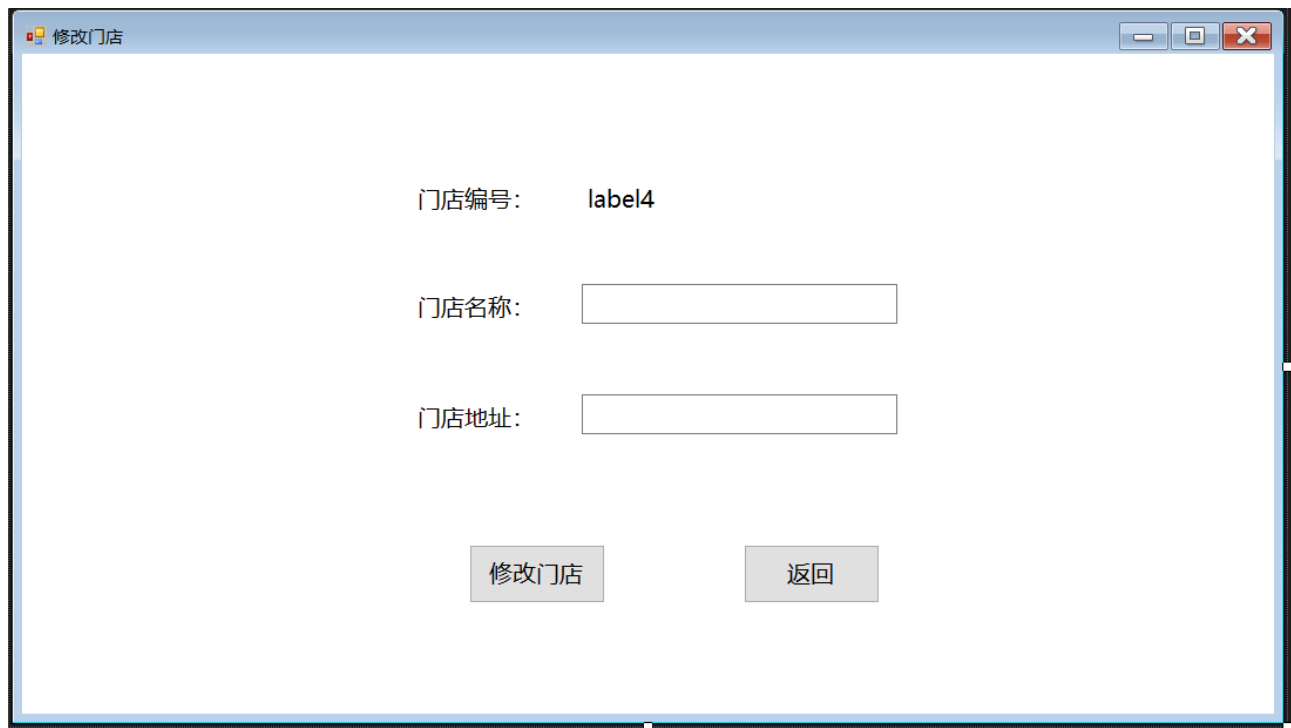
门店名称:

门店地址:

添加门店 返回

ii. 门店修改界面设计:

在门店信息管理界面选中相关门店，然后进入当前界面自动显示门店的编号，管理员可以进行门店名称以及门店地址的修改。



门店编号: label4

门店名称:

门店地址:

修改门店 返回

③ 用户界面设计:

车辆管理员界面，设计菜单界面 menuStrip 实现管理员对功能（功能包括：预约车辆、到店取车、归还车辆与车辆续租以及订单查询）的选择。



a) 订单查询界面设计:

点击订单查询，进入下面界面，系统根据直接显示用户 ID 以及用户姓名，可以通过选择租车日期、还车日期（通过 monthCalendar 进行选择），下拉框选择借车门店、还车门店和车辆类型完成订单查询。同时支持点击清空查询条件。

b) 到店取车界面设计:

包含查询按钮 label1 显示预约信息，dataGridView 显示车辆信息（供用户挑选车辆），当选中相关车型时，可以点击提车完成取车操作。

到店取车

您的预约信息：

2023年 6月22日

车辆类型

租车日期

还车日期

租车门店

还车门店

已缴纳押金/元

请挑选您的车辆

注意请点击车辆

查询预约信息

车辆编号	车辆型号	车牌号

提车

c) 车辆归还与续租界面设计：

归还车辆与续租车辆

会员编号

会员姓名

点击查询您的订单

查询

租借车型

车辆牌号

续约

还车

提车日期

还车日期

您想续租到：

2023年 6月22日

确认已缴费

提车门店

还车门店

信息确认无误 还车

信息确认无误，续租

点击查询显示当前用户的用车情况，点击续约按钮可以进行续约日期的选择，点击还车按钮可以进行还车操作。

d) 车辆预约界面设计：

左边为 dataGridview 显示当前没有出租的车辆信息，包含车辆编号、车辆型号、车牌号以及提车门店（地址），同时用户可以根据需求进行车辆型号，取车门店的选择，dataGridview 的内容会根据用户选择实时更新，在用户车辆选择完成之后，可以进行还车门店、借车日期、还车日期的选择，选择完毕用户进行押金缴纳，最后预约（信息确认无误，预约按钮）完成车辆预约操作。

七、功能实现（字数不限）

按照业务流程，走一遍，每一个步骤放一个截图。例如从登录（注册）-预约-挂号-就诊-填写病例并开药-缴费-取药……，每一个过程的执行过程截图。

会员需要先注册，会自动生成自增编号，该编号包含注册日期的年月，注册成功之后，返回登录界面，选择用户，输入相应的账户和密码登录，

Login

欢迎登录租车平台

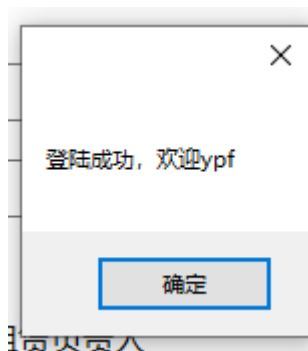
账户:

密码:

若您没有注册会员, 请先注册:

☒ 用户 ☐ 车辆租赁负责人

若您已注册会员, 忘记密码:



如果忘记密码, 需要联系系统管理员, 找回您的密码



用户可以预约车辆、到店取车、归还车辆和续租车辆、订单查询,

预约车辆

车辆编号	车辆型号	车牌号	提车地址
C00002	特斯拉MODEL3	京A88888	北京市海淀区学院...
C00004	比亚迪汉	浙AD4567	浙江省杭州市上城...
C00005	长安	京B67777	浙江省萧山区机场...
C00006	特斯拉MODELX	沪C90YU1	北京市海淀区学院...
C00009	比亚迪宋	豫C89IQS	北京市海淀区学院...
C00010	比亚迪宋	京D987BC	浙江省萧山区机场...
C00011	比亚迪汉	京E569BQ	浙江省杭州市上城...
C00012	保时捷911	京D0U189	上海市机场路号
C00013	仰望U8	苏G76532	上海市开封路号
C00015	大众	浙F89365	北京市朝阳区百望...
C00016	长安	苏B67YTR	北京市海淀区学院...
C00017	特斯拉MODELX	沪C90U18	江苏省南京市大学...
C00020	比亚迪唐	苏C89EFS	浙江省杭州市上城...
C00021	比亚迪宋	浙R76GBC	江苏省苏州市机场...
C00022	比亚迪唐	京T529BQ	北京市朝阳区百望...
C00023	比亚迪汉	豫E569BQ	广东省广州市学院...
C00024	保时捷911	沪D0U189	江苏省南京市大学...
C00026	比亚迪唐	京H896AY	北京市海淀区学院...
C00027	大众	浙FM93L5	广东省广州市学院...
C00028	长安	苏B6AE9H	北京市海淀区学院...
C00031	大众	粤ZVI49Q	上海市开封路号
C00032	长安	苏C89E2S	江苏省苏州市机场...

车辆型号:

取车门店:

预计归还门店:

提车日期:

预计归还日期:

温馨提示:
若您无法在预定时间
归还车辆,请及时续租,
否则会有相应费用扣除

清空筛选

缴纳押金

押金已缴纳

信息确认无误, 预约

通过下拉框的多条件查询和左侧窗体可以快速选择心仪车辆、缴纳押金并预约，进入到店取车界面，

到店取车

您的预约信息:

2023年 6月22日

车辆类型

租车日期

还车日期

租车门店

还车门店

已缴纳押金/元

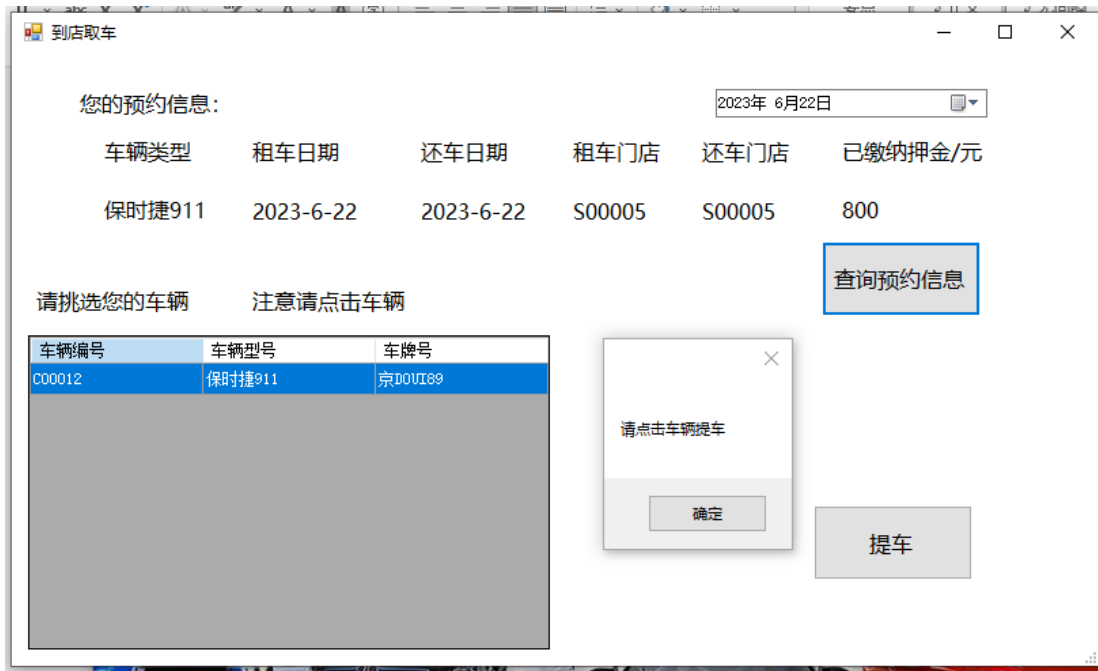
请挑选您的车辆

注意请点击车辆

查询预约信息

车辆编号	车辆型号	车牌号
------	------	-----

提车



进入归还车辆与续租车辆界面，
出租的车辆存在逾期和未逾期两种情况，

归还车辆与续租车辆

会员编号

U202101006

会员姓名

bv

租借车型

比亚迪宋

车辆牌号

C00007

提车日期

2023-5-30

还车日期

2023-6-10

提车门店

北京市朝阳区西大望路号

还车门店

浙江省萧山区机场路号

点击查看您的订单

查询

您已经逾期，请尽快还车，同时系统禁止您进行续租

确定

逾期则超出部分需要多支付 50%的金额，
未逾期即缴纳费用之后还车，

归还车辆与续租车辆

会员编号

U202101006

会员姓名

bv

租借车型

比亚迪宋

车辆牌号

C00007

提车日期

2023-5-30

还车日期

2023-6-10

提车门店

北京市朝阳区西大望路号

还车门店

浙江省萧山区机场路号

点击查看您的订单

查询

还车

推荐使用微信支付

微信支付二维码

请扫码支付

确认已缴费

进入查询订单界面，

订单查询

会员编号	会员姓名	借车日期	还车日期	借车门店	还车门店	车辆类型	账单金额
U202201005	as	2023/1/5	2023/1/9	完美租	神州租车	特斯拉MODEL Y	4000

会员编号: U202201005

会员姓名: as

借车日期:

还车日期:

借车门店:

还车门店:

车辆类型:

查询

清空筛选条件

返回

会员查询订单只能查询到自己曾经的订单记录，若过多时，可以通过右侧多条件筛选来查询，会员的业务功能部分结束。

下面是管理员的业务功能部分：

Login

欢迎登录租车平台

账户: admin

密码: ***

☐ 用户

☒ 车辆租赁负责人

登录

取消

若您没有注册会员，请先注册:

注册

若您已注册会员，忘记密码:

找回密码

管理员可以进行车辆管理、门店管理、门店的车辆信息和订单查询四个功能，



进入车辆管理界面，



可以通过下拉框和模糊查询快捷地进行多条件查询，并通过鼠标直接点击左侧记录进行车辆的删改，
删除车辆，



该车在门店可以删除，



该车不在门店，无法删除
修改车辆牌号，



修改车辆

车辆编号: C00016

车辆型号: 长安

车牌号: 苏B67YTR

修改车辆 返回

不在门店的车辆同删除，无法修改，只能修改在门店的车辆，
添加车辆时，分为两种，该车型为是否已有该车型，



添加车辆

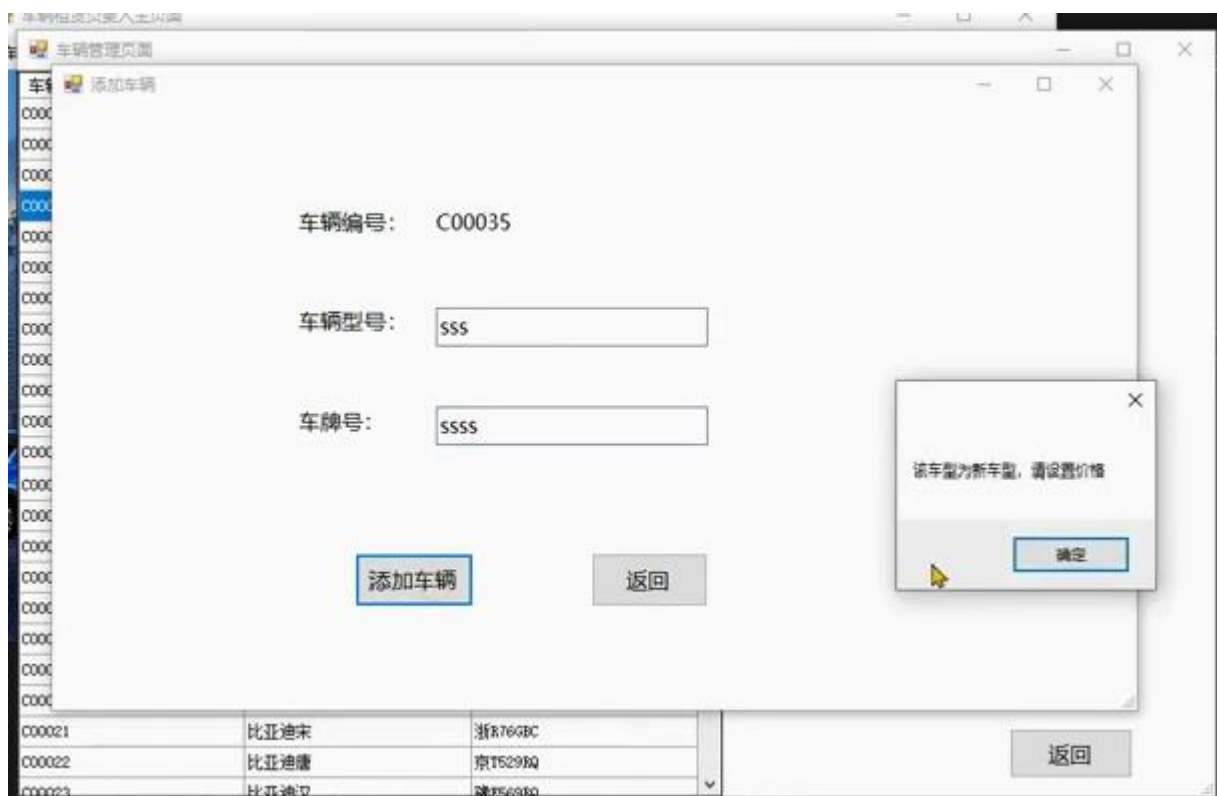
车辆编号: C00033

车辆型号: 比亚迪宋

车牌号: 浙A123456

添加车辆 返回

那么添加成功，同时对该车型的所有数量进行+1 操作，
如果该型号没有，那么需要设置新车型的车辆每日租价，



设置完成后，置该车型的所有数量为 1

下面进入门店管理，

门店管理页面

门店编号	门店名称	门店地址
S00001	天下租车	北京市海淀区学院路号
S00003	完美租	浙江省杭州市上城区金福桥路号
S00005	租车到家	上海市机场路号
S00006	租车一号	江苏省南京市大学路号
S00007	车辆世家	广东省广州市学院路号
S00008	租车壹号	江苏省苏州市机场路号
S00010	车行世界	上海市开封路号

当前选中的门店是：
S00007车辆世家

添加门店

修改门店

删除门店

刷新

条件查询：

门店号：

门店名称：

门店地址：

查询

清空筛选条件

返回

同车辆管理，此处不多赘述，
添加，

添加门店

门店编号：S00011

门店名称：

门店地址：

添加门店

返回

修改，

修改门店

门店编号：S00007

门店名称：

车辆世家

门店地址：

广东省广州市学院路号

修改门店

返回

删除，

门店管理页面

门店编号	门店名称	门店地址
S00001	天下租车	北京市海淀区学院路号
S00003	完美租	浙江省杭州市上城区金福桥路号
S00005	租车到家	上海市机场路号
S00006	租车一号	江苏省南京市大学路号
S00007	车辆世家	广东省广州市学院路号
S00008	租车壹号	江苏省苏州市机场路号
S00010	车行世界	上海市开封路号

当前选中的门店是：
S00008租车壹号

添加门店

修改门店

删除门店

刷新

条件查询：

×

该门店有车需要还，无法删除

确定

查询

清空筛选条件

返回

下面进入门店的车辆信息查询，

门店的车辆信息

门店编号	门店名称	门店地址	车辆编号	车辆型号	车牌号	车辆是否租赁
S00001	天下租车	北京市海淀区...	C00002	特斯拉MODEL3	京A88888	0
S00002	神州租车	北京市朝阳区...	C00003	特斯拉MODEL Y	沪AABC99	1
S00003	完美租	浙江省杭州市...	C00004	比亚迪汉	浙AD4567	0
S00004	租车世界	浙江省萧山区...	C00005	长安	京B67Y7Q	0
S00001	天下租车	北京市海淀区...	C00006	特斯拉MODEL Y	沪C90YU1	0
S00002	神州租车	北京市朝阳区...	C00007	比亚迪宋	浙B12EW6	1
S00004	租车世界	浙江省萧山区...	C00008	特斯拉MODEL3	豫ZUI49Q	1
S00001	天下租车	北京市海淀区...	C00009	比亚迪宋	豫C89IQS	0
S00003	完美租	浙江省杭州市...	C00011	比亚迪汉	京E569EQ	0
S00005	租车到家	上海市机场路号	C00012	保时捷911	京D0UI89	0
S00010	车行世界	上海市开封路号	C00013	仰望U8	苏G76532	0
S00008	租车壹号	江苏省苏州市...	C00014	特斯拉MODEL X	沪H996AY	1
S00009	车行天下	北京市朝阳区...	C00015	大众	浙F89365	0
S00001	天下租车	北京市海淀区...	C00016	长安	苏B67Y7R	0
S00006	租车一号	江苏省南京市...	C00017	特斯拉MODEL X	沪C90VU8	0
S00004	租车世界	浙江省萧山区...	C00018	保时捷	浙B12NW6	1
S00002	神州租车	北京市朝阳区...	C00019	大众	苏ZUI49Q	1
S00003	完美租	浙江省杭州市...	C00020	比亚迪唐	苏C89EFS	0
S00008	租车壹号	江苏省苏州市...	C00021	比亚迪宋	浙R76GBC	0
S00009	车行天下	北京市朝阳区...	C00022	比亚迪唐	京T529EQ	0

门店编号:

门店名称:

门店地址:

刷新

查询

返回

这对于管理员了解不同门店的车辆所有情况具有重要意义，
下面进入订单查询，

订单查询

会员编号	会员姓名	借车日期	还车日期	借车门店	还车门店	车辆类型	账单金额
U202001001	ypf	2020/12/5	2020/12/6	天下租车	完美租	比亚迪汉	1000
U202201005	as	2023/1/5	2023/1/9	完美租	神州租车	特斯拉MODEL Y	4000
U202101002	wsa	2020/12/6	2020/12/6	车行世界	租车到家	保时捷911	1000
U202309010	gf	2022/12/13	2022/12/15	车行天下	神州租车	大众	900
U202305008	lo	2021/1/10	2021/1/13	租车一号	车辆世家	比亚迪汉	2000
U202311011	hb	2023/3/5	2023/3/6	车辆世家	天下租车	长安	800
U202306007	hk	2022/11/12	2022/11/15	完美租	车行天下	保时捷	3600
U202212003	ww	2020/7/5	2020/7/6	租车到家	租车壹号	长安	800
U202101002	wsa	2022/9/4	2020/9/5	神州租车	车辆世家	大众	600
U202306009	jy	2021/12/5	2021/12/8	车行世界	租车一号	保时捷911	4000
U202309010	gf	2023/1/5	2023/1/6	车行天下	天下租车	比亚迪唐	600
U202101006	bv	2023/4/2	2023/4/5	车辆世家	车行天下	比亚迪唐	1200
U202101006	bv	2023/5/30	2023/6/22	神州租车	租车世界	比亚迪宋	21600

会员编号:

会员姓名:

借车日期:

还车日期:

借车门店:

还车门店:

车辆类型:

查询

清空筛选条件

返回

管理员通过右侧的多条件查询查询相应订单，
以上就是全部的功能实现。

八、总结（300-500 字）

1. 系统的主要特色及创新点（创新点有单独的评分，注意此部分一定不能省略）
2. 遇到的困难及解决方法

3. 组内的分工，各自的工作量

1. 系统的主要特色及创新点：

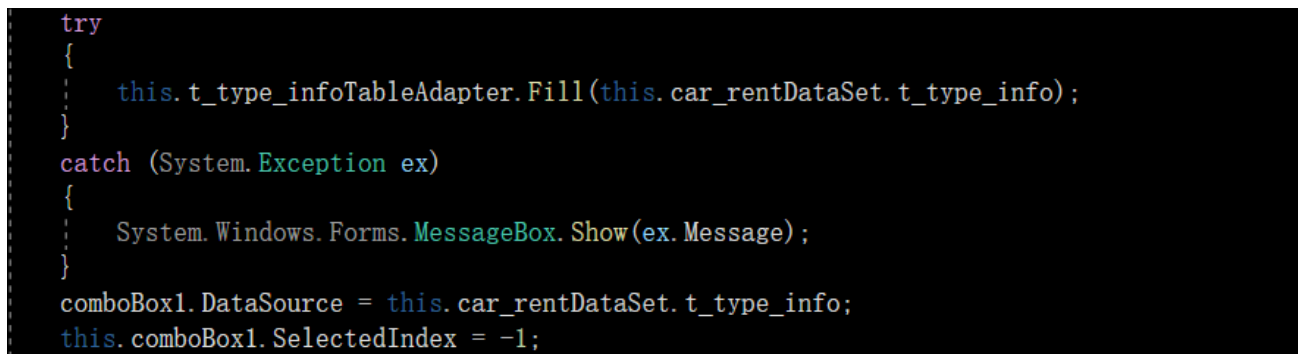
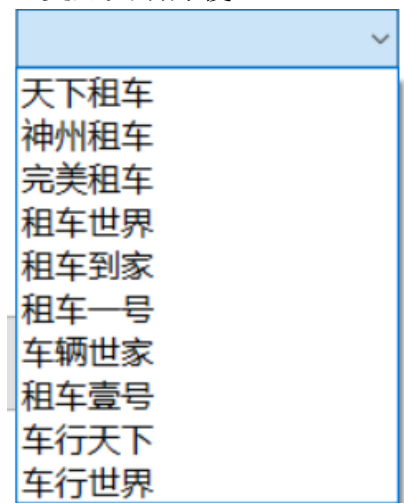
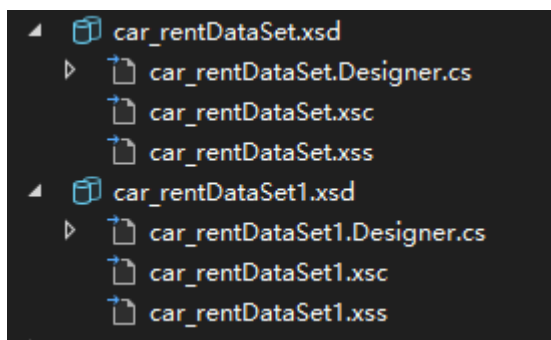
1) 系统功能方面：

能够实现现实应用中异地还车的复杂场景。

设计车辆所在门店表，记录车辆所在门店以及车辆是否正在租车状态，同时预约采用车型预约的方式，实现异地还车的同时，不会产生预约长时间不取车导致的车辆租赁停滞，能够最大限度提高车辆流动性以及系统的完整性。

2) 交互方面：

采用 C#语言进行设计，界面干净整洁、美丽漂亮，能够便捷地实现交互。同时，在数据交互方面：下拉框采用绑定数据源的形式，不是固定的文本类型，更加灵活方便。



3) 系统结构方面：

1、系统的插入存储采用 sql server 的存储过程，插入删除的速率提高，效率增大。


```

GO
CREATE proc return_car
@id VARCHAR(45),
@r_date DATE,
@payment FLOAT,
@over TINYINT
as
BEGIN
/*删除预约表数据*/
DELETE FROM t_car_res WHERE uid = @id;
/*设置rent表数据, 并获取变量值*/
/*由于门店信息与预约表相同 所以将还车日期, 是否逾期设置即可*/
UPDATE t_car_rent
SET t_car_rent.returndate = @r_date
WHERE uid = @id;
UPDATE t_car_rent
SET t_car_rent.overdue = @over
WHERE uid = @id;
/*添加用户订单数据*/
DECLARE @car_id VARCHAR(45);
DECLARE @rent_d DATE;
DECLARE @rent_s VARCHAR(45);
DECLARE @return_s VARCHAR(45);
select @car_id = carid from t_car_rent where uid = @id;
select @rent_d = rentdate from t_car_rent where uid = @id;
select @rent_s = rentstore from t_car_rent where uid = @id;
select @return_s = returnstore from t_car_rent where uid = @id;
insert into t_user_order values(@id,@car_id,@rent_d,@r_date,@rent_s,@return_s,@payment);
DELETE FROM t_car_rent WHERE uid = @id;
END
GO

```

2、根据查询的使用频度创建索引，加快查询速度。部分索引的显示：

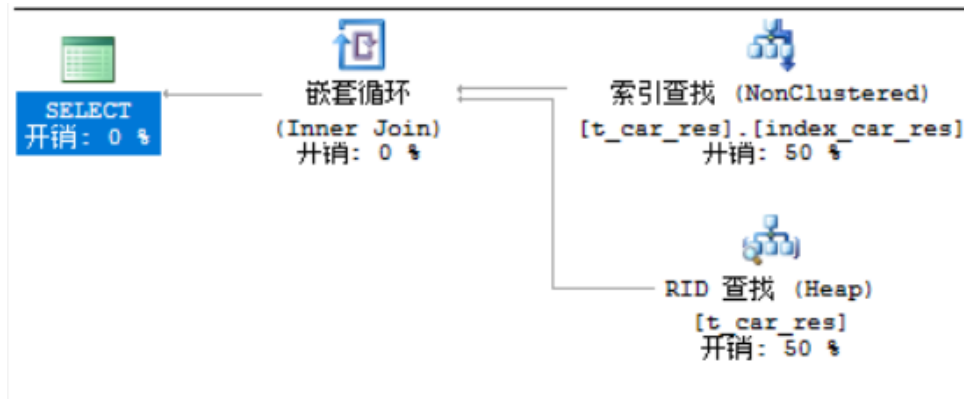
```

-- 创建唯一非聚集索引
create unique nonclustered
index index_pla_carid
on t_car_pla(carid)
with
(
    ignore_dup_key=on,
    statistics_norecompute=off
)
-- 创建索引
CREATE INDEX Index_pla_store
ON t_car_pla (carstoreid)

```

--表示创建唯一非聚集索引
--索引名称
--数据表名称（建立索引的列名）
--表示向唯一索引插入重复值会忽略重复值
--表示启用统计信息自动更新功能

使用 SQL SERVER 对查询性能进行解释，



查询都包含索引查找能够提高查询效率。

4) 具有支付功能，缴纳定金和账单金额时弹出微信二维码。

2. 遇到的困难及解决方法：

困难

1) 主要困难：预约与续租冲突：如果用户还未还车，想要续租会影响当前预约表，同时如果

用户逾期也会对下一位用户的使用造成影响。

2) 次要困难: C#语言从零学起。

解决方案

- 1) 与王庆老师积极讨论, 我们苦思冥想, 最终思考出能够解决问题的形式: 采用预约一定成功的方式, 预约所需车型、地点以及缴纳押金, 可以方便快速提车。若无该车型, 那么提车时选择其他车型, 这样在车型数量足够的情况下, 能够实现正常的车辆使用以及车辆流通。
- 2) 跟随网络的教程, 逐步搭建前端框架, 刻苦钻研 C#与数据库连接的操作, 最终完成系统的实现。

3. 组内的分工, 各自的工作量:

王 伟: 1) C#与 sql server 连接, C#基础架构。

2) 车辆负责人功能的 sql 与 C#实现。

姚攀峰: 1) 建立数据库与数据集, 会员功能的 sql 与 C#实现。

2) 下拉框内容添加数据源。

3) sql server 存储过程编写。

共 同: 数据流图, 功能实现讨论, 撰写答辩 ppt 和实验报告。