

**数据库系统原理实践报告**

|  |  |
| --- | --- |
| 项目名称： | 数据库系统原理实践 |
| 姓 名： | 饶鸿斌 |
| 专 业： | 计算机科学与技术 |
| 班 级： | CS1703 |
| 学 号： | U201714660 |
| 指导教师： | 丁晓峰 |

|  |  |
| --- | --- |
| 分数 |  |
| 教师签名 |  |

2020年 5 月 24 日

**教师评分页**

|  |  |
| --- | --- |
| 子目标 | 子目标评分 |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
|  |  |

目 录

[1 课程任务概述 1](#_Toc43125575)

[2 软件功能学习部分 3](#_Toc43125576)

[2.1 任务要求 3](#_Toc43125577)

[2.2 完成过程 3](#_Toc43125578)

[2.3任务总结 5](#_Toc43125579)

[3 Sql练习部分 5](#_Toc43125580)

[3.1 任务要求 5](#_Toc43125581)

[3.2 完成过程 5](#_Toc43125582)

[3.3任务总结 27](#_Toc43125583)

[4 数据库应用系统设计 28](#_Toc43125584)

[4.1系统设计目标 28](#_Toc43125585)

[4.2 需求分析 28](#_Toc43125586)

[4.3 总体设计 28](#_Toc43125587)

[4.4 数据库设计 30](#_Toc43125588)

[4.5 详细设计与实现 33](#_Toc43125589)

[4.6 系统测试 40](#_Toc43125590)

[4.7 系统设计与实现总结 46](#_Toc43125591)

[4 课程总结 46](#_Toc43125592)

[附录 47](#_Toc43125593)

# 1 课程任务概述

实验一：软件功能学习部分

1.了解并练习数据库管理系统软件的备份功能。

2.了解并练习在数据库上配置用户权限的操作。

实验二：Sql练习部分

1.建表

2.数据更新

3.查询

4.了解系统的查询性能分析功能（选做）

5.DBMS函数及存储过程和事务（选座）

**关于正文章节标题的排版要求：**

报告章标题称为一级标题，章内小节标题依次分为二级标题、三级标题等。一级标题的编号用数字1，2，…编制；二级标题的编号用1.1，1.2，…编制；三级标题的编号用1.1.1，1.2.1，… 编制；四级及以后各级标题可依此类推。建议标题不超过3级（如1.1.1），超出部分可根据需要使用(1)，①，A，a)，…等形式描述。

标题编号与标题文字之间均用空格隔开，如：“1 引言”、“2.1 需求分析”。报告正文的一级标题（章）须另起一页居中排版。

1级标题宋体小三号字加黑，单倍行距，段前、段后均12磅间隔。

2级标题宋体四号字加黑，1.25倍行距，段前6磅，段后0磅间隔。

3级标题宋体小四号字加黑，单倍行距，段前3磅、段后8磅间隔。

**关于正文中的图的排版要求：**

正文中所有插图要求图面整洁，布局合理，线条粗细均匀，圆弧连接光滑，尺寸标注规范。所有曲线、图表、线路图、流程图、程序框图、示意图等不准徒手画，必须按国家规定标准或工程要求采用计算机或手工绘制。

所有插图均应有图号和图名。图号按章编，如第2章的图为图2.1、图2.2、…，第3章的图为图3.1、图3.2、…等。图名是插图的名称，扼要概括图的内容，字数不宜太多。图号和图名写在图下方，并相对于图居中排版。少数图有图注，图注写在图下面且字号应比图号、图名的字小一号，图名和图注后面均不加标点符号。

所有插图均应在正文中予以引用。引用某插图时，一般写为“…见图x.y”或“图x.y是…”。正文中的插图均须安排在文中第一次引用到该图的正文下面，要求先见文，后见插图，且图一般不跨页绘制。

图中文字、图号和图名，统一采用小五号宋体。

**关于正文中的表的排版要求：**

表格由表号、表名、表头、表身等组成。表号按章编，如第2章的表为表2.1、表2.2、…，第3章的表为表3.1、表3.2、…等。表名是表格的名称，扼要概括表的内容，字数不宜太多。表号、表名放在表的正上方，相对于表体居中排版。表号及表名后不加标点。表头包括栏头、行头，与表身一起构成表格的主体。表中的竖称为栏，横格称为行。表身的内容，一般包括：数据、文字、公式和表图等。表内的数据对应位要对齐。少数表有表注，表注写在表下面且字号应比表号、表名的字小一号。

所有表格均应在正文中予以引用。引用某表格时，一般写为“…见表x.y”或“表x.y是…”。表格应尽量靠近正文的叙述，一般应先见文，后见表，表不跨节。表格允许转页。表格转页部分可以不写表号和表名，但要重复书写表头，并在表头右上角写“（续）”字标注。

表中文字、图号和图名，统一采用小五号宋体。

**关于正文中公式的排版要求：**

公式一般另行居中写，公式末不加标点。若公式前有文字，如例、解、证、假定等，文字顶格写，公式仍居中写。一行写不下时，公式允许转行。公式转行需处理得当，做到既意义正确，又使版面美观匀称。

公式要有编号，公式编号按章编，如第2章的公式为(2.1)、(2.2)、…，第3章的公式为(3.1)、(3.2)、…等。公式编号写在公式右侧行末顶边线，并加圆括号。

公式一般应在正文中予以引用，引用时以公式编号指示公式。正文中常有公式中表示量的符号说明，采用“式中”二字作为标志。一般可写成接排形式，如“式中，A指……；B指……”。

**关于实践报告的目录格式要求：**

目录是课程实验报告的纲要。正文的各级标题（一般最多取三级）、附录应编入目录，但目录本身不出现在其中。

目录要求层次清晰，含标题及对应的起始页号。目录的最后 “附录”无章节号。

课程实验报告正文、参考文献和附录页面，使用“1，2，3，… ”编连续页码。页码应标在页面的右下角。

目录中正文的各级标题名称、参考文献和附录及其对应的起始页号，务必与报告中正文的各级标题名称、附录及其对应的起始页号保持一致。

# 软件功能学习部分

## 2.1 任务要求

1）练习SQL Server或其他某个主流关系数据库管理系统软件的备份方式，要求要有通过数据库的软件功能进行的备份和通过文件形式的脱机备份。

2）练习在新增的数据库上增加用户并配置权限的操作。

## 2.2 完成过程

### 2.2.1 数据库备份

本次实验使用的是MySql数据库以及提供图形化界面的Navicat。

本次实验任务二部分的数据库名为lab1，我们使用该数据库来进行备份练习。在Navicat中点击备份，选中新建备份，依据提示进行备份。备份练习结果如图2.1所示：

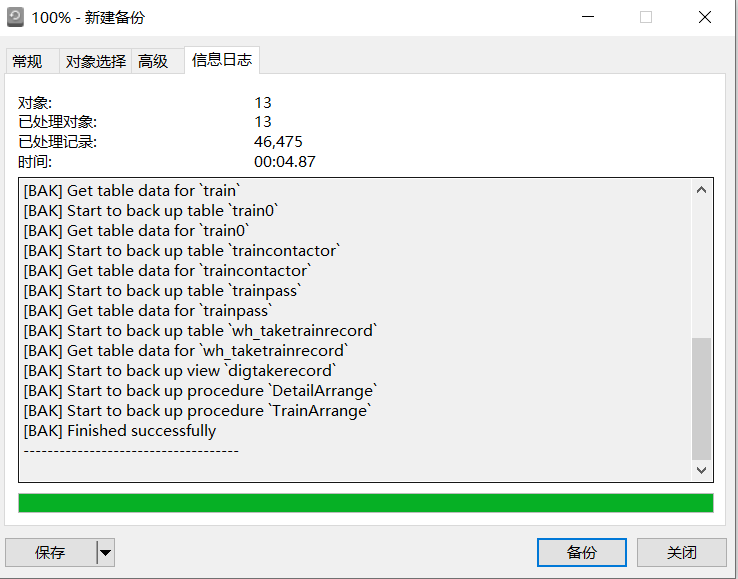


图2.1 新建备份结果

备份的信息将在Navicat中显示，显示的内容如图2.2所示：

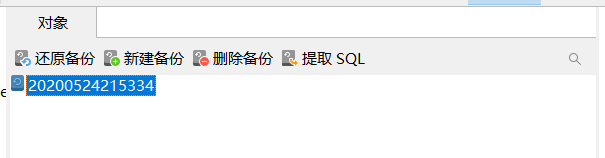


图2.2 备份结果显示

在新建备份后，右键点击还原备份即可还原备份数据，如图2.3所示：

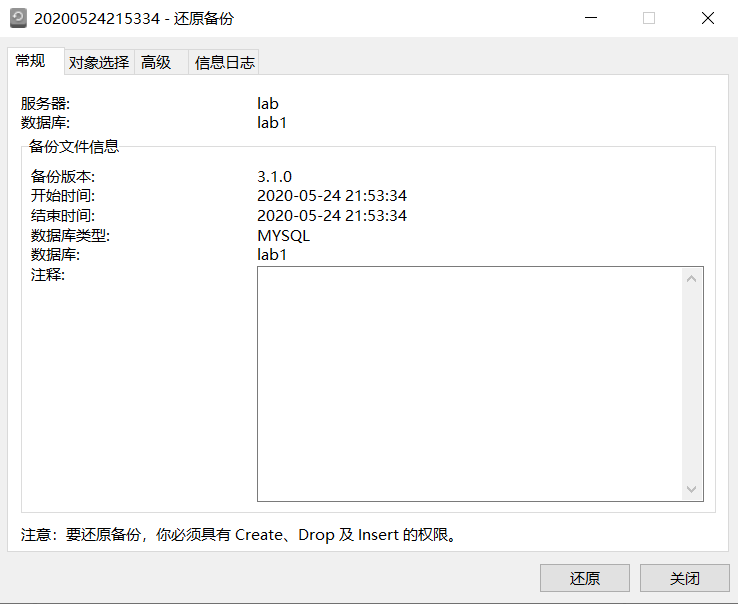


图2.3 还原备份

假设需要采用通过文件形式的脱机备份，则可以在对应目录下将整个数据库的文件拷贝下来。

### 2.2.2 用户权限配置

用户权限配置将在Navicat中完成，如图2.4所示：

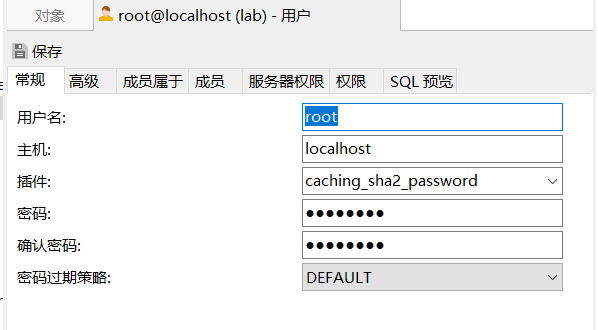


图2.4 用户权限配置

在图2.4对于的服务器权限和权限中可以针对用户进行相应的权限配置。

## 2.3任务总结

本次实验使用了图形化界面软件Navicat，所以难度不是很大。基本上主要集中对软件的使用上。

# 3 Sql练习部分

## 3.1 任务要求

1）创建与“疫期乘坐列车”相关的关系，包括主码和外码的说明。同时向表格中录入适当数量的实验数据，从而对相关的实验任务能够起到验证的作用。

2）数据更新。完成增删改操作、批处理操作、数据导入导出操作、创建视图操作、触发器实验等。

3）查询。用一条SQL语句完成16个小题中的查询需求。

4）了解系统的查询性能分析功能（选做）。

5）DBMS函数及存储过程和事务（选座）。

## 3.2 完成过程

### 3.2.1 建表

1）创建与“疫期乘坐列车”相关的关系。

创建表的sql语句如下所示：

-- 车站表【车站编号，车站名，所属城市】

-- Station (SID int, SName char(20), CityName char(20))

-- 其中，主码为车站编号。

create table station(

SID int,

sname char(20),

cityname char(22),

constraint PK\_station primary key(SID)

);

-- 车次表【列车流水号，发车日期，列车名称，起点站编号，终点站编号，开出时刻，终点时刻】

-- Train (TID int, SDate date, TName char(20), SStationID int, AStationID int, STime datetime, ATime datetime)

-- 其中，TID为主码，(列车名称，发车日期)为候选码；SStationID和AStationID都来源于车站表的SID。

create table train(

TID int,

SDate date,

TName char(20),

SStationID int,

AStationID int,

STime datetime,

ATime datetime,

constraint PK\_train primary key (TID),

constraint UK\_date\_tname unique(SDate,TName),

constraint FK\_train\_SStationId foreign key(SStationID) REFERENCES station(SID),

constraint FK\_train\_AStationId foreign key(AStationID) REFERENCES station(SID)

);

set FOREIGN\_KEY\_CHECKS =0;

drop table train;

set FOREIGN\_KEY\_CHECKS =1;

-- 建立部分车次表，以转存excel表数据

-- 增加的startID：起点站编号、destID：终点站编号;depart 出发时间；days:第几天到达

create table train0(

trainno char(11) primary key,

startID int,

start char(10),

destID int,

dest char(10),

depart time,

arrival time,

days int,

distance int

);

update train0 join station on train0.start = station.sname

set startid = station.SID;

update train0 join station on train0.dest = station.sname

set destid = station.SID;

create procedure TrainArrange()

begin

declare TID1 int;

declare SDate1 date;

declare TName1 char(20);

declare SStationID1 int;

declare AStationID1 int;

declare STime1 time;

declare ATime1 time;

declare d int;

declare arrivalDate Date;

declare done boolean default 0;

declare train\_cur cursor for select trainno,startID,destID,depart,arrival,days from train0 order by trainno;

declare continue handler for not FOUND set done =1;

open train\_cur;

fetch train\_cur into TName1,SStationID1,AStationID1,STime1,ATime1,d;

set TID1=1;

while not done do

set SDate1='2020-01-10';

while(SDate1<='2020-01-25') do

if(d<2) then set arrivalDate = SDate1;

else set arrivalDate = date\_add(SDate1,interval d-1 day);

end if;

-- 2020-01-23至2020-4-07武汉出发和武汉到达的车次停运

if not (

((arrivalDate>'2020-01-22' and arrivalDate<'2020-04-08') and

(AStationID1 in (select SID from station where cityname ='武汉'))) or

((SDate1>'2020-01-22' and SDate1 <'2020-04-08')and

(SStationID1 in (select SID from station where cityname ='武汉')))

)

then insert into train values(TID1,SDate1,TName1,SStationID1,AStationID1,

concat(SDate1, " ", STime1),

concat(arrivalDate, " ", ATime1));

set TID1 =TID1 +1;

end if;

set SDate1 =date\_add(SDate1,interval 1 day);

end while;

fetch train\_cur into TName1,SStationID1,AStationID1,STime1,ATime1,d;

end while;

close train\_cur;

end ;

drop procedure TrainArrange;

call TrainArrange();

-- 车程表【列车流水号，车站序号，车站编号，到达时刻，离开时刻】

-- TrainPass (TID int, SNo smallint, SID int, STime datetime, ATime datetime)

-- 其中，主码为(TID,SNo)。SID来源于车站表的SID。

create table trainpass(

TID int,

SNo smallint,

SID int,

STime datetime,

ATime datetime,

CONSTRAINT PK\_TrainPass primary key(TID,SNo),

CONSTRAINT FK\_TP\_TID FOREIGN key(TID) REFERENCES Train(TID),

CONSTRAINT FK\_TP\_SID FOREIGN key(SID) REFERENCES station(SID)

);

-- 中间表，导入部分车次的运行时刻表：车次，站次，···

create table timetable(

trainno char(11),

SNo smallint,

SID int,

sname char(20),

STime time,

ATime time,

D int,

PRIMARY key(trainno,SNo)

);

-- 填写SID车站的编号

update timetable join station on station.sname = timetable.sname

set timetable.SID = station.SID;

-- 遍历train表，每一行只取TID,SDtate和TName

-- 根据TNAME从timeable 中取出SNO,SID,ATIME,D,WTHER TRAIN0 = TNMAE对应的额遍历

-- 遍历每一个站点，从出发日期算出到站时间，离站时间然后插入

delimiter $$

create procedure DetailArrange()

begin

declare TID1,SNO1,SID1 INT;

declare SDate1 date;

declare TName1 char(20);

declare STime1 time;

declare ATime1 time;

declare d1 int;

declare arrivalDate Date;

declare departDate Date;

declare outer\_done boolean default 0;

declare inner\_done boolean default 0;

declare train\_cur cursor for select TID,SDate,TName from train order by TID;

declare tt\_cur cursor for

select SNO,SID,STime,ATime,D from timetable where trainno =TName1 order by SNO;

declare continue HANDLER for not found

begin

set outer\_done=1;

set inner\_done=1;

end;

open train\_cur;

fetch train\_cur into TID1,SDate1,TName1;

while not outer\_done do

set inner\_done =0;

open tt\_cur;

fetch tt\_cur into SNO1,SID1,STime1,ATime1,D1;

while not inner\_done do

if(D1<2) then set arrivalDate =SDate1;

else set arrivalDate = date\_add(SDate1,interval D1-1 day);

end if;

if STime1 <ATime1 then set departDate = arrivalDate;

else set departDate=date\_add(arrivalDate,interval 1 day);

end if;

if not ((arrivalDate>'2020-01-22' and arrivalDate<'2020-04-08')

and

(SID1 IN (SELECT SID FROM station where cityname='武汉'))) then

insert into trainpass values(TID1,SNO1,SID1,

concat(arrivalDate ," ", STime1),

concat(departDate ," ", ATime1));

end if;

fetch tt\_cur into SNO1,SID1,STime1,ATime1,D1;

END WHILE;

SET outer\_done=0;

close tt\_cur;

fetch train\_cur into TID1 ,SDate1,TName1;

end while;

close train\_cur;

end$$

drop procedure DetailArrange;

delimiter;

call DetailArrange();

-- 乘客表【乘客身份证号，姓名，性别，年龄】

-- Passenger (PCardID char(18), PName char(20), Sex bit, Age smallint)

-- 其中，主码为乘客身份证号；性别取值为0/1（“1”表示“男”，“0”表示“女”）。

create table Passenger (

PCardID char(18),

PName char(20),

Sex bit,

Age smallint,

CONSTRAINT PK\_passenger primary key(PCardID)

);

-- 键姓名表

-- sno为自增长列，

-- create table nmlist(

-- sno int primary key auto\_increment,

-- pid char(18),

-- nm char(10),

-- sex bit,

-- age smallint

-- );

-- 乘车记录表【记录编号，乘客身份证号，列车流水号，出发站编号，到达站编号，车厢号，席位排号，席位编号，席位状态】

-- TakeTrainRecord (RID int, PCardID char(18), TID int, SStationID int, AStationID int, CarrigeID smallint, SeatRow smallint，SeatNo char(1)，SStatus int)

-- 其中，主码、外码请依据应用背景合理定义。

-- CarrigeID若为空，则表示“无座”；

-- SeatNo只能取值为’A’、’B’、’C’、’E’、’F’，或为空值；

-- SStatus只能取值’0’（退票）、’1’（正常）、’2’（乘客没上车）。

create table TakeTrainRecord (

RID int,

PCardID char(18),

TID int,

SStationID int,

AStationID int,

CarrigeID smallint,

SeatRow smallint,

SeatNo char(1),

SStatus int,

CONSTRAINT PK\_TTR\_RID PRIMARY key(RID),

CONSTRAINT FK\_TTR\_PCardID foreign key (PCardID) REFERENCES passenger(PCardID),

CONSTRAINT FK\_TTR\_SStationID foreign key(SStationID) REFERENCES station(SID),

CONSTRAINT FK\_TTR\_AStationID foreign key(AStationID) REFERENCES station(SID),

CONSTRAINT CK\_TTR\_SeatNo check(SeatNo in('A','B','C','D','E','F')),

CONSTRAINT CK\_TTR\_SStatus check(SStatus in (0,1,2))

);

-- 诊断表【诊断编号，病人身份证号，诊断日期，诊断结果，发病日期】

-- DiagnoseRecord (DID int, PCardID char(18), DDay date, DStatus smallint, FDay date)

-- 其中，主码为DID；DStatus包括：1：新冠确诊；2：新冠疑似；3：排除新冠

create table DiagnoseRecord (

DID int,

PCardID char(18),

DDay date,

DStatus smallint,

FDay date,

CONSTRAINT PK\_DR\_DID PRIMARY key(DID),

CONSTRAINT CK\_DR\_DStatus check(DStatus in(1,2,3))

);

-- 乘客紧密接触者表【接触日期, 被接触者身份证号，状态，病患身份证号】

-- TrainContactor (CDate date, CCardID char(18), DStatus smallint, PCardID char(18))

-- 其中，主码为全码。DStatus包括：1：新冠确诊；2：新冠疑似；3：排除新冠

create table TrainContactor (

CDate date,

CCardID char(18),

DStatus smallint,

PCardID char(18),

CONSTRAINT PK\_TC\_all PRIMARY key (CDate,CCardID,DStatus,PCardID),

CONSTRAINT CK\_TC\_DStatus check(DStatus in(1,2,3))

);

在Navicat中运行对应的语句后，我们讲得到如图3.1所示的表：

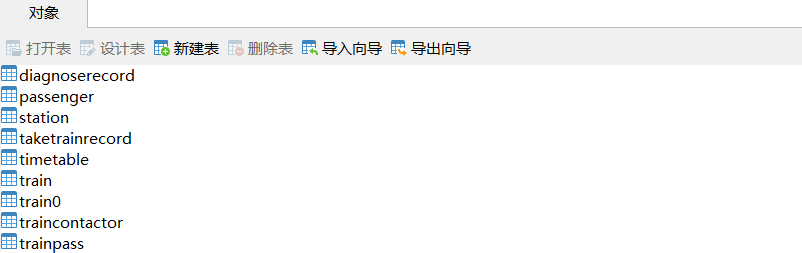


图3.1 数据库建表

其中train0、timeable为过渡表。

2）数据准备

利用Navicat的导入向导功能向各个表中导入老师所提供的数据，以passenger表为例，导入后passenger表的内容如图3.2所示：



图3.2 passenger表内容

其余表中的数据不再赘述。

### 3.2.2 数据更新

1）分别用一条sql语句完成对乘车记录表基本的增、删、改的操作

Sql语句如下所示：

INSERT INTO TakeTrainRecord VALUES(41,'65432319991017375X',1,1599,1621,7,2,'C',2);

UPDATE TakeTrainRecord SET SStatus = 1 where RID = 41;

DELETE FROM TakeTrainRecord WHERE RID = 41;

运行后结果如图3.3所示：



图3.3 对表的增、删、改结果

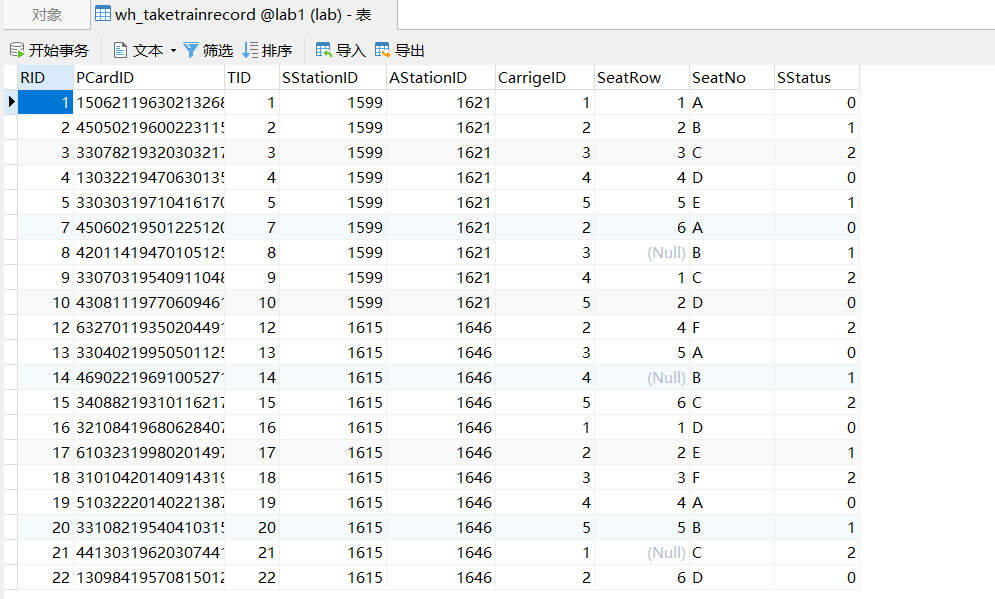
2）批处理操作

将乘车记录表中的从武汉出发的乘客的乘车记录插入到一个新表WH\_TakeTrainRecord中。

执行的sql语句如下所示：

CREATE TABLE WH\_TakeTrainRecord AS (SELECT \* FROM TakeTrainRecord where SStationID IN (SELECT SID FROM station WHERE cityname='武汉'));

执行后的结果如图3.4所示：

图3.4 wh\_taketrainrecord表内容

3）数据导入导出

通过查阅DBMS资料学习数据导入导出功能，并将任务2.1所建表格的数据所建表格的数据导出到操作系统文件，然后再将这些文件的数据导入到相应空表。

运行的语句如下所示：

SELECT \* FROM WH\_TakeTrainRecord INTO OUTFILE "D:/MySQL/MySQL Server 8.0/Uploads/wh\_taketrainrecord.txt";

执行后的结果如图3.5和图3.6所示：

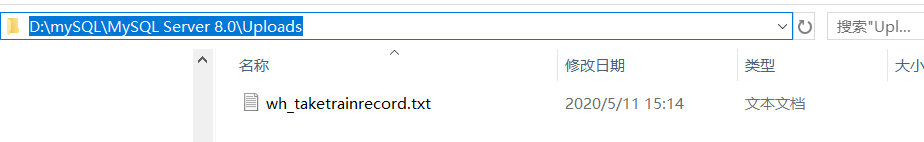


图3.5 数据导入导出结果

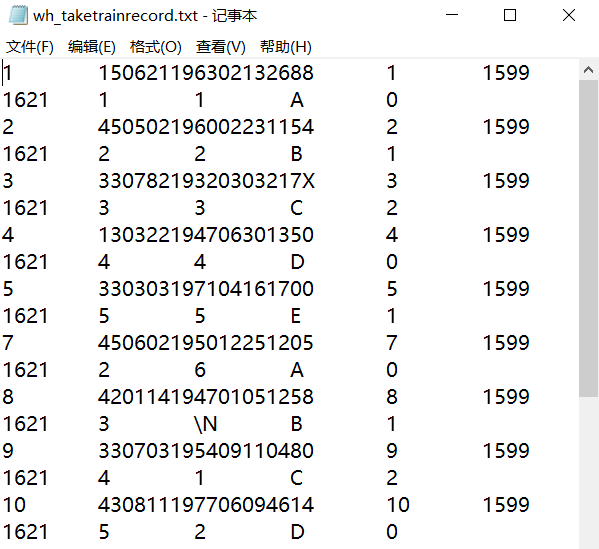


图3.6 数据导入导出结果

4）观察性实验

建立一个关系，但是不设置主码，然后向该关系中插入重复元组，然后观察在图形化交互界面中对已有数据进行删除和修改时所发生的现象。

执行的sql语句如下所示：

DROP TABLE IF EXISTS test;

CREATE TABLE test(

a int,

b int,

c int

);

INSERT INTO test VALUES (1,2,3);

INSERT INTO test VALUES (1,2,3);

INSERT INTO test VALUES (1,2,3);

#均发生了修改

SELECT \* FROM test;

DROP TABLE test;

运行结果如图3.7所示：



图3.7 对重复元组进行删除和修改结果

5）创建视图

创建一个新冠确诊病人的乘火车记录视图，其中的属性包括：身份证号、姓名、年龄、乘坐列车编号、发车日期、车厢号，席位排号，席位编号。按身份证号升序排序，如果身份证号一致，按发车日期降序排序（注意，如果病人买了票但是没坐车，不计入在内）。

执行的sql语句如下所示：

DROP VIEW IF EXISTS digTakeRecord;

CREATE VIEW digTakeRecord

AS SELECT passenger.PCardID,PName,Age,train.TID,SDate,CarrigeID,SeatRow,SeatNo

FROM passenger,train,taketrainrecord,diagnoserecord

WHERE passenger.PCardID=diagnoserecord.PCardID

AND passenger.PCardID=diagnoserecord.PCardID

AND train.TID=taketrainrecord.TID

AND diagnoserecord.DStatus=1

AND taketrainrecord.SStatus=1

ORDER BY passenger.PCardID,train.SDate DESC;

创建好的视图如图3.8所示：

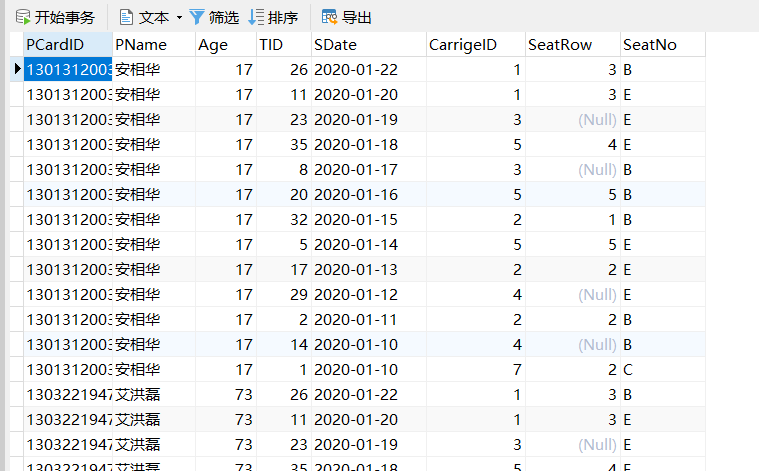


图3.8 视图digTakeRecord内容

6）触发器实验

编写一个触发器，用于实现以下完整性控制规则：

1) 当新增一个确诊患者时，若该患者在发病前14天内有乘车记录，则将其同排及前后排乘客自动加入“乘客紧密接触者表”，其中：接触日期为乘车日期。

2) 当一个紧密接触者被确诊为新冠时，从“乘客紧密接触者表”中修改他的状态为“1”。

执行的语句如下所示：

CREATE

TRIGGER trigger1 AFTER UPDATE ON DiagnoseRecord

FOR EACH ROW

BEGIN

declare Cdate date;

declare Cardid char(18);

declare Dstatus int;

declare Pcardid char(18);

set Dstatus =2;

set Pcardid = new.PCardID;

select TakeTrainRecord.PCardID into Cardid where TakeTrainRecord.TID in (select train.TID from TakeTrainRecord,train where TakeTrainRrcord.TID = train.ITD and

new.Pcardid = TakeTrainRecord.PCardID and

new.Dstatus =1 and

train.SDate>=date\_sub(new.FDay,interval 14 day)and

train.SDate<=new.FDay

);

if Cardid!=null then

insert into TrainContactor values(Cdate,Cardid,Dstatus,Pcardid);

END IF;

update TrainContactor set DStatus = new.DStatus where(TrainContactor.CCardID = new.PCardID);

END;

乘客记录表中插入一位新乘客

插入语句为：

insert into taketrainrecord values(42,110101201105182137,1,1599,1621,7,3,'A',1);

如图3.9所示：

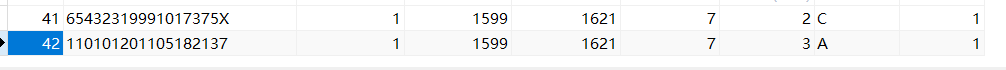


图3.9 新增乘客

让该乘客发病

插入语句为：

insert INTO diagnoserecord VALUES (43,110101201105182137,'2020-01-20',1,'2020-01-09');

原本的traincontactor表中没有数据。如图3.10所示：

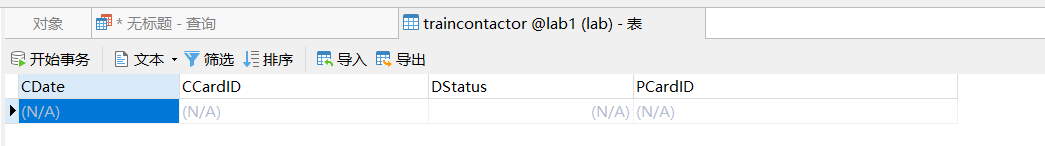


图3.10 traincontactor表中数据

依据上述的插入语句，traincontactor中将插入一条数据。如图3.11所示：

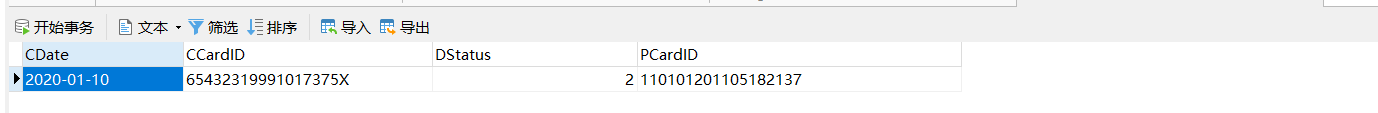


图3.11 traincontactor表中数据

### 3.2.3 查询

1）查询确诊者“张三”的在发病前14天内的乘车记录

查询语句如下所示：

select TakeTrainRecord.\*,passenger.PName from TakeTrainRecord,train,DiagnoseRecord,passenger where

TakeTrainRecord.TID = train.TID and

DiagnoseRecord.PCardID=TakeTrainRecord.PCardID and

passenger.PCardID =DiagnoseRecord.PCardID AND

passenger.PName= '张三' AND

DiagnoseRecord.DStatus = 1 AND

train.SDate <= DiagnoseRecord.FDay AND

train.SDate >= DATE\_SUB(DiagnoseRecord.FDay,INTERVAL 14 DAY);

查询结果如图3.12所示：

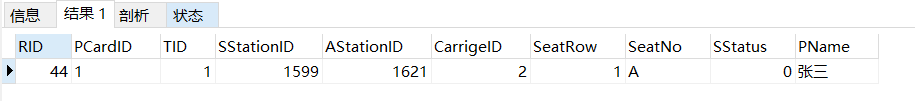


图3.12 查询结果

2）查询所有从城市“武汉”出发的乘客乘列车所到达的城市名

查询的sql语句如下：

select DISTINCT station.cityname from station,TakeTrainRecord where

station.SID = TakeTrainRecord.AStationID and

TakeTrainRecord.SStationID IN

(SELECT station.SID from station where station.cityname='武汉');

查询结果如图3.13所示：

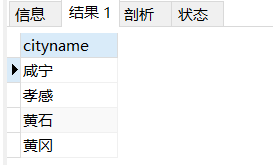


图3.13 查询结果

3）计算每位新冠患者从发病到确诊的时间间隔（天数）及患者身份信息，并将结果按照发病时间天数的降序排列

查询的sql语句如下：

select datediff (DiagnoseRecord.DDay,DiagnoseRecord.FDay)As DayInterval,passenger.\* from passenger,DiagnoseRecord where

passenger.PCardID =DiagnoseRecord.PCardID and

DiagnoseRecord.DStatus=1 ORDER BY DayInterval desc;

查询结果如图3.14所示：

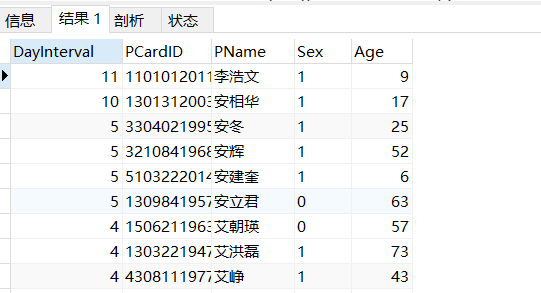


图3.14 查询结果

4）查询“2020-01-22”从“武汉”发出的所有列车

查询的sql语句如下：

select train.\* from train where train.SDate ='2020-01-22' and

train.SStationID in (select distinct SID FROM station where

station.cityname='武汉');

查询结果如图3.15所示：

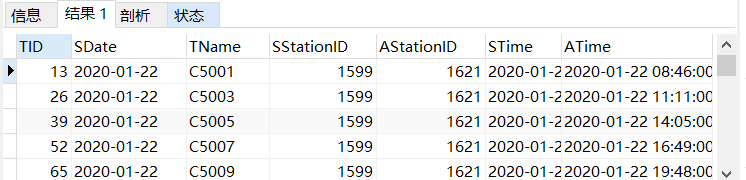


图3.15 查询结果

5）查询“2020-01-22”途经“武汉”的所有列车

查询的sql语句如下：

SELECT train.\*

FROM train,trainpass

WHERE train.TID=trainpass.TID AND train.SDate='2020-01-22'

AND trainpass.SID IN

(SELECT SID FROM station

WHERE CityName='武汉');

查询结果如图3.16所示：

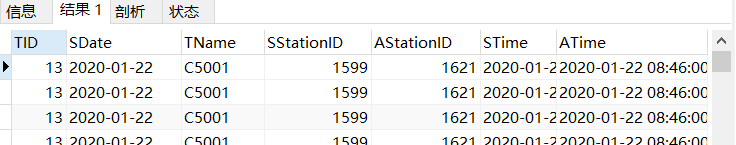


图3.16 查询结果

6）查询“2020-01-22”从武汉离开的所有乘客的身份证号、所到达的城市、到达日期

查询的sql语句如下：

select passenger.PCardID,station.cityname,train.ATime from passenger,TakeTrainRecord,station,train where passenger.PCardID=TakeTrainRecord.PCardID and

station.SID=TakeTrainRecord.AStationID and

train.TID=TakeTrainRecord.AStationID and

TakeTrainRecord.TID in

(select DISTINCT train.TID from train where train.SDate ='2020-01-22'and

train.SStationID in

(select distinct SID FROM station where

station.cityname='武汉'))

GROUP BY(cityname);

查询结果如图3.17所示：

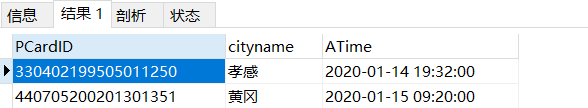


图3.17 查询结果

7）统计“2020-01-22” 从武汉离开的所有乘客所到达的城市及达到各个城市的武汉人员数

查询的sql语句如下：

select station.cityname,count(cityname) as number from passenger,taketrainrecord,station,train where

passenger.PCardID = taketrainrecord.PCardID and

station.SID = taketrainrecord.AStationID and

train.TID = taketrainrecord.TID and

taketrainrecord.TID in

(select DISTINCT train.TID from train where train.SDate ='2020-01-22'and

train.SStationID in

(select distinct SID FROM station where

station.cityname='武汉'))

GROUP BY(cityname);

查询结果如图3.18所示：



图3.18 查询结果

8）查询2020年1月到达武汉的所有人员

查询的sql语句如下：

select distinct passenger.\* from passenger,taketrainrecord,station,train where passenger.PCardID = taketrainrecord.PCardID and

taketrainrecord.AStationID in

(select distinct SID FROM station where

station.cityname='武汉')

and taketrainrecord.TID = train.TID and

train.ATime<'2020-02-01';

查询结果如图3.19所示：

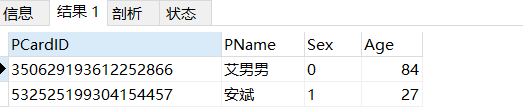


图3.19 查询结果

9）查询2020年1月乘车途径武汉的外地人员（身份证非“420”开头）

查询的sql语句如下：

select distinct passenger.\* from passenger,train,trainpass,taketrainrecord where

passenger.PCardID=taketrainrecord.PCardID and

trainpass.TID=train.TID and

left(passenger.PCardID,3)!='420' and

trainpass.SID in

(select distinct SID FROM station where

station.cityname='武汉') and

taketrainrecord.TID=train.TID and

train.STime>='2020-01-01' and

train.STime<'2020-02-01';

查询结果如图3.20所示：



图3.20 查询结果

10）统计“2020-01-22”乘坐过‘G107’号列车的新冠患者在火车上的密切接触乘客人数（每位新冠患者的同车厢人员都算同车密切接触）

查询的sql语句如下：

select count(\*) as counts from taketrainrecord where

(taketrainrecord.TID,taketrainrecord.CarrigeID)in

(select taketrainrecord.TID,taketrainrecord.CarrigeID from

passenger,taketrainrecord,train,diagnoserecord where

taketrainrecord.TID=train.TID and

train.ATime='2020-01-22' and

train.TName='G007' and

passenger.PCardID=taketrainrecord.PCardID and

diagnoserecord.PCardID=passenger.PCardID and

diagnoserecord.DStatus=1);

查询结果如图3.21所示：



图3.21 查询结果

11）查询一趟列车的一节车厢中有3人及以上乘客被确认患上新冠的列车名、出发日期，车厢号

查询的sql语句如下：

select train.TName,train.SDate,taketrainrecord.CarrigeID from

train,taketrainrecord,diagnoserecord where

train.TID=taketrainrecord.TID and

taketrainrecord.PCardID=diagnoserecord.PCardID and

diagnoserecord.DStatus=1

GROUP BY train.TName,train.SDate,taketrainrecord.CarrigeID having count(\*)>=3;

查询结果如图3.22所示：



图3.22 查询结果

12）查询没有感染任何周边乘客的新冠乘客的身份证号、姓名、乘车日期

查询的sql语句如下：

SELECT DISTINCT passenger.PCardID,PName,CDate

FROM passenger,traincontactor

WHERE passenger.PCardID=traincontactor.PCardID AND CCardID NOT IN

(SELECT DISTINCT CCardID FROM traincontactor WHERE DStatus=1);

查询结果如图3.23所示：

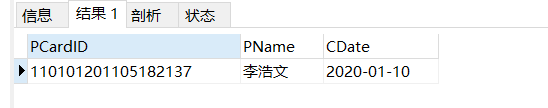


图3.23 查询结果

13）查询到达 “北京”、或“上海”，或“广州”（即终点站）的列车名，要求where子句中除了连接条件只能有一个条件表达式

查询的sql语句如下：

select DISTINCT train.TName from train where

train.AStationID in

(select station.SID from station where

station.cityname='北京' or station.cityname='上海' or

station.cityname='广州');

查询结果如图3.24所示：

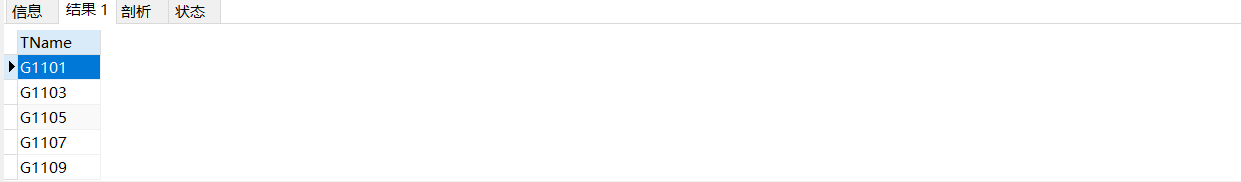


图3.24 查询结果

14）查询“2020-01-22”从“武汉站”出发，然后当天换乘另一趟车的乘客身份证号和首乘车次号，结果按照首乘车次号降序排列，同车次则按照乘客身份证号升序排列

查询的sql语句如下：

select train.TID,passenger.PCardID from train,passenger,taketrainrecord WHERE

passenger.PCardID = taketrainrecord.PCardID and

taketrainrecord.TID = train.TID and

train.TID in

(select train.TID from train where train.SDate='2020-01-22' and train.SStationID in

(select DISTINCT SID from station where

station.sname='武汉'))and

((taketrainrecord.AStationID,passenger.PCardID)in

(select taketrainrecord.SStationID,passenger.PCardID from train,passenger,taketrainrecord where

passenger.PCardID=taketrainrecord.PCardID and

taketrainrecord.TID = train.TID and train.TID in

(select train.TID from train where train.SDate='2020-01-22')))

ORDER BY train.TID desc,PCardID;

15）查询所有新冠患者的身份证号，姓名及其2020年以来所乘坐过的列车名、发车日期，要求即使该患者未乘坐过任何列车也要列出来

查询的sql语句如下：

select temp.PCardID,PName,Tname,SDate from

(select passenger.PCardID,PName from

passenger,diagnoserecord where

passenger.PCardID=diagnoserecord.PCardID and

diagnoserecord.DStatus=1)

temp left join

(select TName,SDate,PCardID from taketrainrecord,train where

taketrainrecord.TID=train.TID and

train.SDate>='2020-01-01')

temp1 on temp.PCardID = temp1.PCardID ORDER BY temp.PCardID,temp1.SDate desc;

查询结果如图3.25所示：

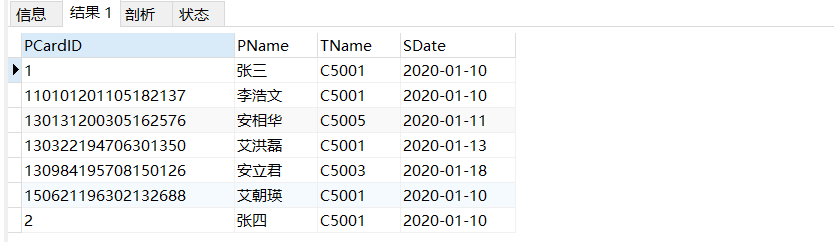


图3.25 查询结果

16）查询所有发病日期相同而且确诊日期相同的病患统计信息，包括：发病日期、确诊日期和患者人数，结果按照发病日期降序排列的前提下再按照确诊日期降序排列

查询的sql语句如下：

select FDay,DDay,count(\*) as total from

diagnoserecord where DStatus =1 group by

FDay,DDay ORDER BY FDay desc,DDay desc;

查询结果如图3.26所示：

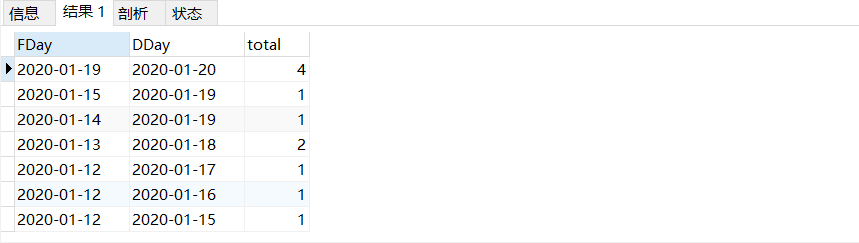


图3.26 查询结果

## 3.3任务总结

本次实验难度较大，任务量也较大。通过本次任务，我加深了对sql语句的运用。一开始只能做前面几个简单的查询语句，触发器也写的问题很大。然后通过查询之前的ppt和百度相关的知识点，一点点扣，最终把全部的任务都完成了。可以说，本次任务没有十分难的地方，都是平时学习到的知识点。但是，在写的时候需要我们小心，不要出错。例如空格、逗号、分号等遗漏错误，这是很不应该的。

# 4 数据库应用系统设计

## 4.1系统设计目标

自行选择所擅长的DBMS软件以及数据库应用系统（客户端程序或者网站）的程序开发工具，参考后面的题目例子，拟定一个自己感兴趣的数据库应用系统题目，完成该小型数据库应用系统的设计与实现工作。主要内容包括：需求调研与分析、总体设计、数据库设计、详细设计与实现、测试等环节的工作。

选择题目：汽车租借信息系统

## 4.2 需求分析

### 4.2.1 任务文档要求

采用B/S或C/S模式实现一个汽车租借信息系统。完成用户、车辆、经手员工、租借情况、车辆损毁情况、交通违规罚款等信息的管理。

要求：

1）实现不同权限的浏览和更新。

2）能够根据车辆使用情况计算押金退还金额。

3）能查询客户的租借历史记录，并进行信誉度评价，进行会员制和非会员制的客户管理。

4）能够管理车辆报修信息；

5）能够生成租借公司的日、月、季度、年财务报表。

### 4.2.2 需求分析内容

概述：租车信息管理系统是一个同时面向用户和租车公司的一个信息管理系统。

功能需求：

1. 作为用户，可以输入自己的账号和密码进入到租车信息管理系统来查询自己的租借记录等信息。同时可以进行租借汽车和归还汽车的操作。
2. 作为租车公司，可以添加汽车、客户和员工的信息，同时可以在后台看到全部的租借信息。同时还可以以图表的信息查看收益情况。

## 4.3 总体设计

依据系统的功能，基本实现模块如下所示：



图4-1 系统模块实现图

系统的各项功能模块的具体功能如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | 功能名称 | 功能描述 | 解决需求 |
| 1 | JDBC | 使 Java 客户端程序能与后端的 MySQL 链接进行操作。 | 满足前后端链接的驱动需 求。 |
| 2 | 后端数据库 | 使用 MySQL 进行对数据的管 理。 | 保证了数据存储的性能高 效，满足性能需求。 |
| 3 | 前端展示 | 使用动态的图表来展示数据结 果和财报等统计信息 | 满足数据管理统计等增值 功能需求。 |

系统的基本流程图如下所示：



图4-2 系统基本流程图

## 4.4 数据库设计

该系统的主要ER图如下图所示：



图4-3 核心数据E-R图模型

下表为数据库中所存储的关系模式和具体的数据类型：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 键信息 | 非空 | 备注 |
| ID | Int | 主键 | √ | 汽车id号 |
| BRAND | Varchar |  | √ | 汽车品牌 |
| RENT\_PAY | Int |  | √ | 汽车租金 |
| STATUS | Int |  | √ | 汽车状态 |
| PLEDGE\_PAY | Int |  | √ | 汽车押金 |
| VALID | Int |  | √ | 是否租用：0表示未租用，1表示租用 |
| CUSTOMERID | Varchar |  |  | 汽车租借客户ID |
| LASTLOGIN | Datetime |  |  | 汽车租借时间 |

表4-1 车辆信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 键信息 | 非空 | 备注 |
| ID | Int | 主键 | √ | 员工id号 |
| NAME | Varchar |  | √ | 员工姓名 |
| TITLE | Int |  | √ | 员工头衔 |
| PASSWD | Varchar |  | √ | 登录密码 |
| LASTLOGIN | Datetime |  |  | 员工登录时间 |

表4-2 员工信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 键信息 | 非空 | 备注 |
| ID | Int | 主键 | √ | 顾客id号 |
| NAME | Varchar |  | √ | 顾客姓名 |
| MEMBER | Int |  | √ | 是否会员：0表示会员，1表示非会员 |
| PASSWD | Varchar |  | √ | 登录密码 |
| LASTLOGIN | Datetime |  |  | 顾客登录时间 |

表4-3 顾客信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 键信息 | 非空 | 备注 |
| INFOID | Int | 主键 | √ | 日志id号 |
| CARID | Int |  | √ | 租/还汽车ID号 |
| CUSTOMERID | Int |  | √ | 租/还客户ID号 |
| STUFFID | Int |  | √ | 租/还员工ID号 |
| RENT\_TIME | Datetime |  | √ | 租/还时间 |
| REVENT | Int |  | √ | 租用或归还：1表示租用，2表示归还 |
| DETAIL | Varchar |  |  | 此数据暂时无用 |
| RENT\_COST | Int |  | √ | 租借花费 |

表4-4 租车日志表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 数据类型 | 键信息 | 非空 | 备注 |
| PID | Int | 主键 | √ | 利润id号 |
| PEVENT | Varchar |  | √ | 租用或归还：1表示租用，0表示归还 |
| PDATETIME | Int |  | √ | 记录时间 |
| PPROFIT | Double |  | √ | 单次租车利润 |

表4-5 利润表

## 4.5 详细设计与实现

### 4.5.1开发环境

操作系统：windows10专业版

开发语言：java+javafx+css

IDE：Intellij IDEA2020.2.2

数据库：MySQL

### 4.5.2系统业务流程图

系统业务流程图分为顾客业务流程图和员工业务流程图。分别如下所示：



图4-4 顾客业务流程图



图4-5 员工业务流程图

### 4.5.3关键技术和算法说明

（1）登录实现：

算法设计：

1. 首先判断用户是否输入完整账号和密码，如果不是全部输入就发出提醒
2. 执行SQL查询，判断用户密码和账号是否对应。如果密码输入错误则提醒用户。

（2）数据库模块设计：

将针对数据库的增删查改同其他的业务分割开来，形成数据层和具体业务层之间的交互。数据库采用单例模式，防止内存的浪费。同时根据具体的业务需要来扩充数据库管理类的内部方法，方便后续的扩展。

public static synchronized DBConnector getInstance() {

if (instance == null)

instance = new DBConnector();

return instance;

}

（3）数据添加和删除设计：

算法设计：

1. 首先判断添加的数据的ID编号是否已经存在，存在的话就提醒使用者。确保ID是主键。
2. 通过读入输入在textview的数据来插入到相对应的数据库表中。
3. 删除数据将选中的数据在对应的数据表中查询，然后进行删除操作

数据的添加和删除，包括汽车信息、顾客信息、员工信息和日志信息均在服务端完成。

（4）租车和还车设计：

算法设计：

1. 租车时，首先筛选出可以被租借的汽车的信息，加载到待选框中，供顾客进行选择。
2. 还车时，首选筛选出该顾客租借的全部汽车的信息，然后将选中的汽车在数据库对应的表中进行修改，将汽车的租借人和租借时间去掉，并将汽车修改为可租借状态。同时进行日志的登记。

租车和还车操作均在客户端中完成。

（5）日志设计：

算法设计：在顾客进行租车和还车时，日志会记录每一次该操作。

（6）利润设计：

通过计算一个顾客的租车和还车来计算出该顾客的具体费用。

### 4.5.4界面设计

登录界面：



图4-6 登录界面图

顾客登录后的主页：



图4-7 顾客登录主页图

顾客的租车界面：



图4-8 顾客登录租车界面图

顾客还车界面图：



图4-9 顾客登录还车界面图

员工登录主页图：



图4-10 员工登录主页图

员工登录显示客户信息图：



图4-11 员工登录主页客户信息显示图

员工登录显示员工信息图：



图4-12 员工登录主页员工信息显示图

员工登录显示日志信息图：

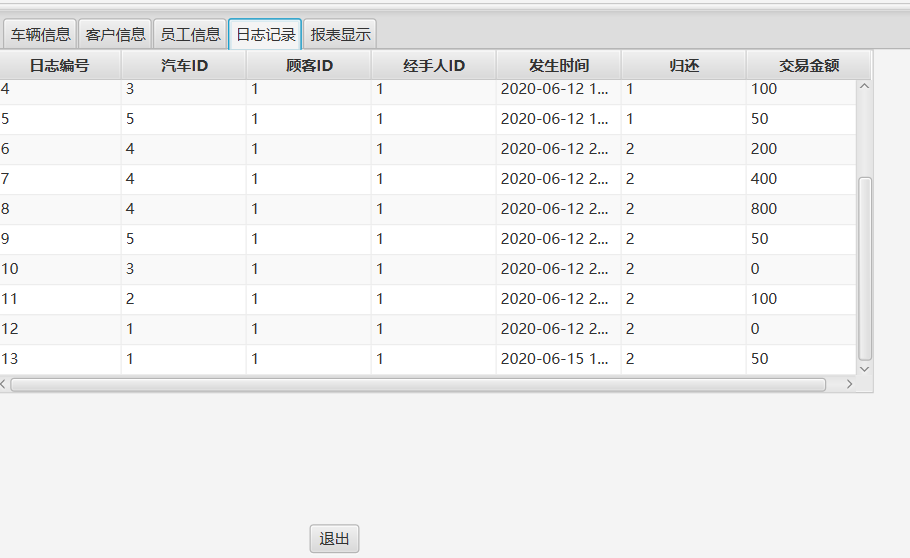


图4-13 员工登录主页日志信息显示图

员工登录显示报表信息图：



图4-14 员工登录主页报表信息显示图

## 4.6 系统测试

（1）登录验证测试：

1、输入错误的密码：



图4-15 输入错误密码

（2）添加汽车、员工、顾客信息测试：

添加前的数据显示如上述界面所显示一样。

添加数据：



图4-16 添加汽车数据

添加后：



图4-17 添加汽车数据后显示图

添加客户信息：



图4-18 添加顾客数据

添加后显示图：



图4-19 添加顾客数据后显示图

添加员工信息：

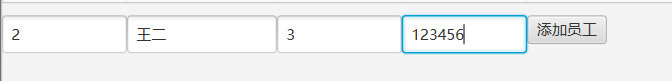


图4-20 添加员工数据

添加后显示图：



图4-21 添加员工数据后显示图

（3）客户端租借还车操作和利润界面：

当前利润界面：

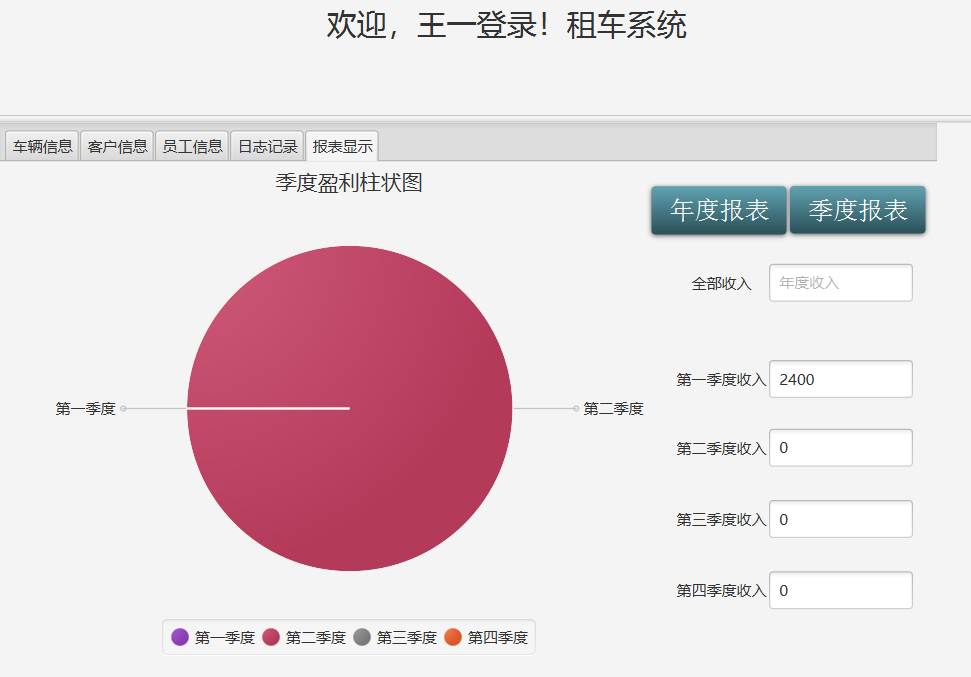


图4-22 利润界面显示图

租借汽车界面：



图4-23 租借汽车显示图

租借汽车成功后，还车界面显示：



图4-24 归还汽车显示图

还车成功后，界面显示：



图4-25 归还汽车成功显示图

此时利润表显示：

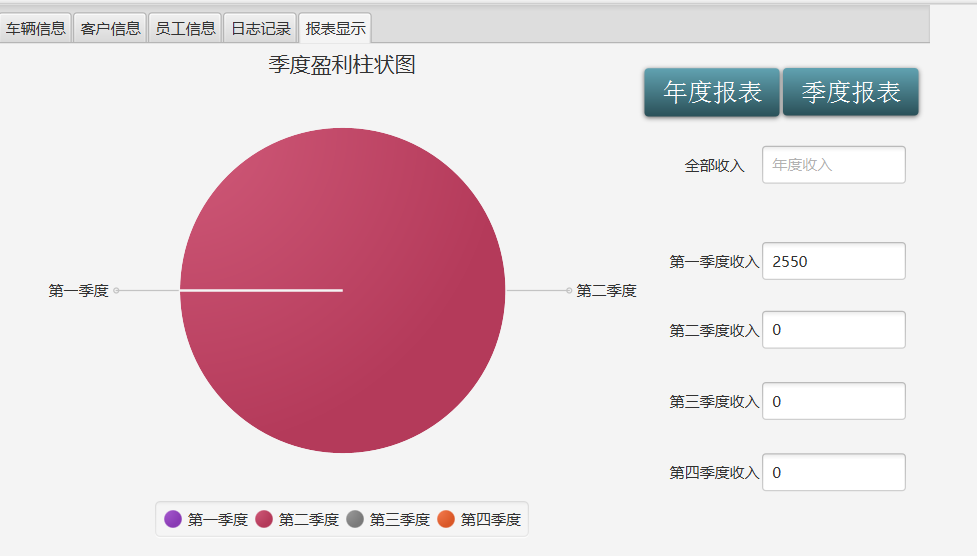


图4-26 归还汽车成功利润显示图

日志表显示成功租借和归还汽车日志：



图4-27 归还汽车成功日志显示图

（4）服务端删除汽车、顾客、员工操作：

删除前的汽车、顾客、员工信息如上所示；

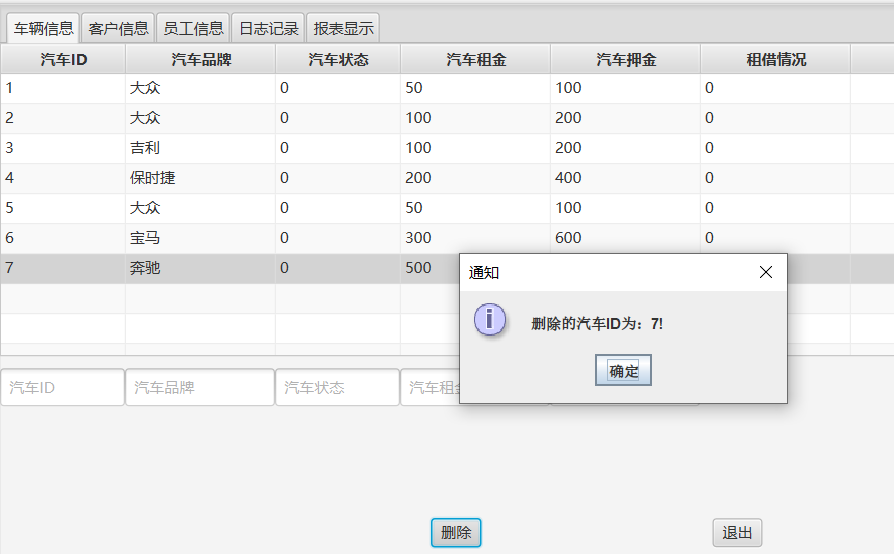


图4-28 删除汽车成功日志显示图



图4-29 删除顾客成功日志显示图



图4-30 删除员工成功日志显示图

## 4.7 系统设计与实现总结

整个系统设计分为如下几个部分：

1. 第一部分：完成数据库的设计

首先撰写文档，设计数据库表类型、ER 图、数据流图等，然后在 MySQL 中建立 相关的表和建立完整性约束。

1. 第二部分：系统架构设计

在纸上设计好了整个系统的流程，规划好了要写哪些模块，哪些文件。

1. 第三部分：前端开发

整个设计是基于javafx设计的，在选择显示UI组件时，参考了网上的一些设计方案。在设计利润的图形显示时，采用了多种效果：饼状图、柱状图等，最后选中饼状图。

1. 第四部分：内部逻辑实现

内部逻辑实现时，按照既定的方案按部就班的进行逻辑业务的代码编写，难度不是很大。

# 4 课程总结

本次课设可以说是时间紧、任务重的一个课设。幸好之前在上java课的时候学习了一些javafx的知识，所以就索性使用javafx来进行课设的设计。整体的框架的搭建也参考了java课程中的第二个大实验，故不是很复杂。

起初在设计数据库的时候，没有画E-R图，设计出的数据库十分的繁杂，不够清晰和简洁。数据之间的关系也不是的混乱。在做出E-R图并进行优化后，最后设计出该数据库。整体的数据库的设计上不是很复杂，但是还是确保了可扩展性，便于后续的继续开发。

总结整个课设的开发过程，我觉得真的让人学习最快的方法就是不断的去上网查询，遇到问题马上解决问题，无论是看博客，还是查文档，真的可以迅速学会很多东西。

感谢课设，感谢老师的指导，也感谢自己这两个星期的付出。

# 附录

1、数据库设计：

/\*采用B/S或C/S模式实现一个汽车租借信息系统。完成用户、车辆、经手员工、租借情况、车辆损毁情况、交通违规罚款等信息的管理。

要求：

1）实现不同权限的浏览和更新。

2）能够根据车辆使用情况计算押金退还金额。

3）能查询客户的租借历史记录，并进行信誉度评价，进行会员制和非会员制的客户管理。

4）能够管理车辆报修信息；

5）能够生成租借公司的日、月、季度、年财务报表。

\*/

/\*ID:编号、cusomterID是租车人的号、BRAND是汽车品牌、RENT\_pay是租金、status是车况、PLEDGE——PAY是押金、valid表示是否可以租借、lastlogin是上次租车时间\*/

DROP TABLE IF EXISTS `carinfo`;

CREATE TABLE `carinfo` (

`ID` int(11) NOT NULL,

`BRAND` varchar(11) NOT NULL,

`RENT\_PAY` int(11) NOT NULL,

`STATUS` int(11) NOT NULL,

`PLEDGE\_PAY` int(11) NOT NULL,

`VALID` int(11) NOT NULL,

`CUSTOMERID` varchar(11) DEFAULT NULL,

`LASTLOGIN` datetime DEFAULT NULL,

PRIMARY KEY (`ID`),

KEY `K5\_idx` (`CUSTOMERID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='车辆信息表';

insert into carinfo values(1,'大众',50,0,100,0,'1',now());

DROP TABLE IF EXISTS `customerinfo`;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `customerinfo` (

`ID` int(11) NOT NULL,

`NAME` varchar(11) NOT NULL,

`MEMBER` tinyint(1) NOT NULL,

`PASSWD` varchar(11) NOT NULL,

`LOGINTIME` datetime DEFAULT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='顾客信息表';

insert into customerinfo values(1,'张一',0,'123456',now());

DROP TABLE IF EXISTS `employeeinfo`;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `employeeinfo` (

`ID` int(11) NOT NULL,

`NAME` varchar(11) NOT NULL,

`TITLE` int(11) NOT NULL,

`PASSWD` varchar(11) NOT NULL,

`LOGINTIME` datetime DEFAULT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='员工信息表';

inset into employeeinfo values(1,'王一',1,'123456',now());

DROP TABLE IF EXISTS `rentdiary`;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `rentdiary` (

`INFOID` int(11) NOT NULL,

`CARID` int(11) NOT NULL,

`CUSTOMERID` int(11) NOT NULL,

`STUFFID` int(11) NOT NULL,

`RENT\_TIME` datetime NOT NULL,

`REVENT` int(11) NOT NULL,

`DETAIL` varchar(255) DEFAULT NULL,

`RENT\_COST` int(11) NOT NULL,

PRIMARY KEY (`INFOID`),

KEY `K1` (`CARID`),

KEY `K3` (`CUSTOMERID`),

KEY `K3\_idx` (`STUFFID`),

CONSTRAINT `K1` FOREIGN KEY (`CARID`) REFERENCES `carinfo` (`ID`),

CONSTRAINT `K2` FOREIGN KEY (`CUSTOMERID`) REFERENCES `customerinfo` (`ID`),

CONSTRAINT `K3` FOREIGN KEY (`STUFFID`) REFERENCES `employeeinfo` (`ID`),

CONSTRAINT `diary\_chk\_1` CHECK ((`INFOID` > 0))

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='日记';

DROP TABLE IF EXISTS `profit`;

SET character\_set\_client = utf8mb4 ;

CREATE TABLE `profit` (

`PID` int(11) NOT NULL,

`PEVENT` int(11) NOT NULL,

`PDATETIME` datetime NOT NULL,

`PPROFIT` double NOT NULL,

PRIMARY KEY (`PID`),

CONSTRAINT `profit\_chk\_1` CHECK ((`PID` > 0))

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='利润表';

2、App.java

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.SplitPane;

import javafx.stage.Stage;

import java.io.IOException;

import java.sql.SQLException;

public class App extends Application {

public static void main(String[] args) {

launch(args);

}

public void start(Stage stage) throws Exception {

try

{

DBConnector.getInstance().connectionDatabase("localhost","car\_rent",3306,"root","123456");

stage.setTitle("汽车租借系统");

Parent root\_login= FXMLLoader.load(getClass().getResource("login.fxml"));

Scene scene = new Scene(root\_login,800,600);

stage.setScene(scene);

stage.show();

}

catch (SQLException e)

{

System.out.println("打开数据库失败！\n");

e.printStackTrace();

}

}

}

3、main,java

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Scene;

import javafx.scene.control.SplitPane;

import javafx.stage.Stage;

import java.sql.SQLException;

public class main {

public static void main(String[] args) {

Application.launch(App.class,args);

}

}

4、loginCotroller.java

import javafx.beans.value.ChangeListener;

import javafx.beans.value.ObservableValue;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.PasswordField;

import javafx.stage.Stage;

import javax.swing.\*;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.ResourceBundle;

public class loginController implements Initializable {

@FXML

public ComboBox<String> Combo\_account,Combo\_type;

@FXML

private PasswordField text\_password;

@FXML

private Button loginButton;

ObservableList<String> ename\_stuff = FXCollections.observableArrayList();

ObservableList<String> cname\_customer = FXCollections.observableArrayList();

ArrayList<String> eid\_stuff,cid\_customer;

@Override

public void initialize(URL url, ResourceBundle resourceBundle) {

eid\_stuff = new ArrayList<>();

cid\_customer = new ArrayList<>();

Combo\_type.getItems().addAll("顾客","员工");

Combo\_type.getSelectionModel().select(0);

try{

DBConnector connector = DBConnector.getInstance();

Connection connection = connector.getConnection();

String sql = "select ID,NAME from customerinfo";

Statement statement =connection.createStatement();

ResultSet resultSet =statement.executeQuery(sql);

String CID,CNAME;

while(resultSet.next())

{

CID=resultSet.getString("ID");

CNAME=resultSet.getString("NAME");

cname\_customer.add(CNAME);

cid\_customer.add(CID);

}

sql="select ID,NAME from employeeinfo";

resultSet=statement.executeQuery(sql);

String EID,ENAME;

while(resultSet.next())

{

EID=resultSet.getString("ID");

ENAME=resultSet.getString("NAME");

ename\_stuff.add(ENAME);

eid\_stuff.add(EID);

}

resultSet.close();

statement.close();

} catch (SQLException throwables) {

throwables.printStackTrace();

}

Combo\_account.setItems(cname\_customer);

Combo\_account.getSelectionModel().select(0);

cname\_customer.removeAll();

Combo\_type.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<String>() {

@Override

public void changed(ObservableValue<? extends String> observableValue, String s, String t1) {

int index = Combo\_type.getSelectionModel().getSelectedIndex();

if(index==0)

{

Combo\_account.setItems(cname\_customer);

}

else if(index==1)

{

Combo\_account.setItems(ename\_stuff);

}

Combo\_account.getSelectionModel().select(0);

}

});

}

@FXML

public void onLoginClick() throws SQLException, IOException {

if(checkPasswordAndUsername()!=true)

return;

DBConnector connector = DBConnector.getInstance();

int type=Combo\_type.getSelectionModel().getSelectedIndex();

String password = text\_password.getText();

int account\_index=Combo\_account.getSelectionModel().getSelectedIndex();

String chooseId=type==0?cid\_customer.get(account\_index):eid\_stuff.get(account\_index);

//登录密码正确

if(connector.CheckPassword(type,Integer.parseInt(chooseId),password)==true)

{

System.out.println("密码验证成功！\n");

String path=null;

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

if(type==0)//顾客

{

path="customer.fxml";

CustomerController.setCustomer\_name(Combo\_account.getValue());

CustomerController.setCustomer\_id(chooseId);

}

else if(type==1)

{

path="stuff.fxml";

StuffController.setStuff\_name(Combo\_account.getValue());

StuffController.setStuff\_id(chooseId);

}

else return;

connector.updateTimecustomerOremployee(type,Integer.parseInt(chooseId));

text\_password.clear();

Stage tempStage = (Stage)loginButton.getScene().getWindow();

tempStage.close();

Parent root = FXMLLoader.load(getClass().getResource(path));

Scene scene = new Scene(root,1000,800);

scene.setRoot(root);

Stage stage = new Stage();

stage.setScene(scene);

stage.show();

}

else

{

JOptionPane.showMessageDialog(null, "密码和账号不匹配！");

}

}

private boolean checkPasswordAndUsername()

{

if(Combo\_account.getValue()==null||Combo\_account.getValue().toString().isEmpty()==true)

{

JOptionPane.showMessageDialog(null,"账号不可为空！");

return false;

}

if(Combo\_type.getValue()==null||Combo\_type.getValue().toString().isEmpty()==true)

{

JOptionPane.showMessageDialog(null,"类型不可为空！");

return false;

}

if(text\_password.getText().equals(" "))

{

JOptionPane.showMessageDialog(null,"密码不可为空！");

return false;

}

return true;

}

}

5.StuffController.java

import emtity.StuffCar;

import emtity.StuffCustomer;

import emtity.StuffDiary;

import emtity.StuffPerson;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.event.EventHandler;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.chart.\*;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.control.cell.TextFieldTableCell;

import javafx.scene.input.MouseEvent;

import javafx.stage.Stage;

import javax.swing.\*;

import java.io.IOException;

import java.net.URL;

import java.sql.SQLException;

import java.util.ResourceBundle;

public class StuffController implements Initializable {

@FXML

Button btn\_exit1;

@FXML

Label label\_title;

@FXML

TextField add\_cmember1,add\_cmember2,add\_cmember3,add\_cmember4,add\_cmember5;

@FXML

private TableColumn<StuffCar,String>stuff\_carid,stuff\_carbrand,stuff\_carstatus,stuff\_carrent,stuff\_carpledge,stuff\_carvalid;

@FXML

private TableColumn<StuffCustomer,String>stuff\_cusid,stuff\_cuspasswd,stuff\_cusname,stuff\_cusmoral,stuff\_cusmember;

@FXML

private TableColumn<StuffPerson,String> stuffid,stuffname,stufftitle,stuffpasswd;

@FXML

private TableColumn<StuffDiary,String>stuff\_did,stuff\_dcarid,stuff\_dcusid,stuff\_dstuffid,stuff\_dtime,stuff\_devent,stuff\_dcost;

@FXML

private TableView<StuffDiary> table\_diary;

@FXML

private TableView<StuffCar> table\_car;

@FXML

private TableView<StuffPerson> table\_stuff;

@FXML

private TableView<StuffCustomer> table\_customer;

@FXML

TextField add\_carid,add\_carbrand,add\_carstatus,add\_carrent,add\_carpledge;

@FXML

TextField add\_cmid,add\_cmname,add\_cmpasswd,add\_cmoral,add\_cmember;

@FXML

TextField add\_smid,add\_smname,add\_smpasswd,add\_smtitle;

@FXML

TextField add\_did,add\_dcarid,add\_dcusid,add\_dstuffid,add\_dtime,add\_devent,add\_dcost;

@FXML

private PieChart pieChart=new PieChart();

ObservableList<StuffCar> StuffCarlist = FXCollections.observableArrayList();

ObservableList<StuffCustomer>StuffCustomerlist = FXCollections.observableArrayList();

ObservableList<StuffDiary>StuffDiarylist = FXCollections.observableArrayList();

ObservableList<StuffPerson>StuffPersonlist=FXCollections.observableArrayList();

ObservableList<PieChart.Data> pieChartData = FXCollections.observableArrayList();

DBConnector connector;

public static String stuff\_id;

public static String stuff\_name;

public static void setStuff\_id(String id)

{

stuff\_id=id;

}

public static void setStuff\_name(String name)

{

stuff\_name=name;

}

@Override

public void initialize(URL url, ResourceBundle resourceBundle) {

connector= DBConnector.getInstance();

label\_title.setText("欢迎，"+stuff\_name+"登录！租车系统");

try {

this.show\_carinfo();

} catch (SQLException throwables) {

throwables.printStackTrace();

}

return;

}

@FXML

//载入汽车信息

public void show\_carinfo() throws SQLException {

stuff\_carid.setCellValueFactory(new PropertyValueFactory<StuffCar,String>("Carid"));

stuff\_carbrand.setCellValueFactory(new PropertyValueFactory<StuffCar,String>("Carbrand"));

stuff\_carrent.setCellValueFactory(new PropertyValueFactory<StuffCar,String>("Carrent\_pay"));

stuff\_carstatus.setCellValueFactory(new PropertyValueFactory<StuffCar,String>("Carstatus"));

stuff\_carpledge.setCellValueFactory(new PropertyValueFactory<StuffCar,String>("Carpledge\_pay"));

stuff\_carvalid.setCellValueFactory(new PropertyValueFactory<StuffCar,String>("Carvalid"));

//对如下单元格进行编辑

stuff\_carstatus.setCellFactory(TextFieldTableCell.forTableColumn());

stuff\_carpledge.setCellFactory(TextFieldTableCell.forTableColumn());

stuff\_carrent.setCellFactory(TextFieldTableCell.forTableColumn());

stuff\_carrent.setOnEditCommit(new EventHandler<TableColumn.CellEditEvent<StuffCar, String>>() {

@Override

public void handle(TableColumn.CellEditEvent<StuffCar, String> stuffCarStringCellEditEvent) {

stuffCarStringCellEditEvent.getTableView().getItems().get(stuffCarStringCellEditEvent.getTablePosition().getRow()).setCarrent\_pay(stuffCarStringCellEditEvent.getNewValue());

}

});

stuff\_carpledge.setOnEditCommit(new EventHandler<TableColumn.CellEditEvent<StuffCar, String>>() {

@Override

public void handle(TableColumn.CellEditEvent<StuffCar, String> stuffCarStringCellEditEvent) {

stuffCarStringCellEditEvent.getTableView().getItems().get(stuffCarStringCellEditEvent.getTablePosition().getRow()).setCarpledge\_pay(stuffCarStringCellEditEvent.getNewValue());

}

});

stuff\_carstatus.setOnEditCommit(new EventHandler<TableColumn.CellEditEvent<StuffCar, String>>() {

@Override

public void handle(TableColumn.CellEditEvent<StuffCar, String> stuffCarStringCellEditEvent) {

stuffCarStringCellEditEvent.getTableView().getItems().get(stuffCarStringCellEditEvent.getTablePosition().getRow()).setCarstatus(stuffCarStringCellEditEvent.getNewValue());

}

});

add\_carid.setPromptText("汽车ID");

add\_carbrand.setPromptText("汽车品牌");

add\_carstatus.setPromptText("汽车状态");

add\_carrent.setPromptText("汽车租金");

add\_carpledge.setPromptText("汽车押金");

//加载汽车信息

StuffCarlist.clear();

StuffCarlist=connector.getAllCarinfo(StuffCarlist);

table\_car.setItems(StuffCarlist);

}

@FXML

public void onExitClick()

{

Stage nextSage=(Stage)btn\_exit1.getScene().getWindow();

nextSage.close();

Parent root = null;

try

{

root = FXMLLoader.load(getClass().getResource("login.fxml"));

}catch (IOException e)

{

e.printStackTrace();

}

Scene scene =new Scene(root,1000,800);

scene.setRoot(root);

Stage stage = new Stage();

stage.setScene(scene);

stage.show();

}

@FXML

//将选中的汽车下架

public void onCarDeleteClick() throws SQLException {

StuffCar stuffCar=table\_car.getSelectionModel().getSelectedItem();

if(stuffCar==null)

{

JOptionPane.showMessageDialog(null,"请选择要删除的信息行！");

return;

}

connector.deleteReturnCar(stuffCar.getCarid(),stuffCar.getCarvalid());

this.show\_carinfo();//刷新

}

@FXML

//添加新上架的汽车

public void onCarAddClick() throws SQLException {

String addcarid=add\_carid.getText();

String addcarbrand=add\_carbrand.getText();

String addcarstatu=add\_carstatus.getText();

String addcarrent=add\_carrent.getText();

String addcarpledge=add\_carpledge.getText();

String addcarvalid="0";

if(addcarid.isEmpty()||addcarbrand.isEmpty()||addcarpledge.isEmpty()||addcarrent.isEmpty()||addcarstatu.isEmpty())

{

JOptionPane.showMessageDialog(null,"请输入完整信息！");

return;

}

connector.addCarinfo(addcarid,addcarbrand,addcarstatu,addcarrent,addcarpledge,addcarvalid);

System.out.println(addcarid);

System.out.println(addcarbrand);

System.out.println(addcarpledge);

add\_carid.clear();

add\_carbrand.clear();

add\_carpledge.clear();

add\_carrent.clear();

add\_carstatus.clear();

this.show\_carinfo();

}

//载入客户信息

@FXML

public void show\_customerinfo() throws SQLException {

stuff\_cusid.setCellValueFactory(new PropertyValueFactory<StuffCustomer,String>("Cusid"));

stuff\_cusname.setCellValueFactory(new PropertyValueFactory<StuffCustomer,String>("Cusname"));

stuff\_cuspasswd.setCellValueFactory(new PropertyValueFactory<StuffCustomer,String>("Cuspasswd"));

stuff\_cusmoral.setCellValueFactory(new PropertyValueFactory<StuffCustomer,String>("Cusmoral"));

stuff\_cusmember.setCellValueFactory(new PropertyValueFactory<StuffCustomer,String>("Cusmember"));

stuff\_cusname.setCellFactory(TextFieldTableCell.forTableColumn());

stuff\_cusmember.setCellFactory(TextFieldTableCell.forTableColumn());

stuff\_cuspasswd.setCellFactory(TextFieldTableCell.forTableColumn());

stuff\_cusmember.setOnEditCommit(new EventHandler<TableColumn.CellEditEvent<StuffCustomer, String>>() {

@Override

public void handle(TableColumn.CellEditEvent<StuffCustomer, String> stuffCustomerStringCellEditEvent) {

stuffCustomerStringCellEditEvent.getTableView().getItems().get(stuffCustomerStringCellEditEvent.getTablePosition().getRow()).setCusmember(stuffCustomerStringCellEditEvent.getNewValue());

}

});

stuff\_cusname.setOnEditCommit(new EventHandler<TableColumn.CellEditEvent<StuffCustomer, String>>() {

@Override

public void handle(TableColumn.CellEditEvent<StuffCustomer, String> stuffCustomerStringCellEditEvent) {

stuffCustomerStringCellEditEvent.getTableView().getItems().get(stuffCustomerStringCellEditEvent.getTablePosition().getRow()).setCusname(stuffCustomerStringCellEditEvent.getNewValue());

}

});

stuff\_cuspasswd.setOnEditCommit(new EventHandler<TableColumn.CellEditEvent<StuffCustomer, String>>() {

@Override

public void handle(TableColumn.CellEditEvent<StuffCustomer, String> stuffCustomerStringCellEditEvent) {

stuffCustomerStringCellEditEvent.getTableView().getItems().get(stuffCustomerStringCellEditEvent.getTablePosition().getRow()).setCuspasswd(stuffCustomerStringCellEditEvent.getNewValue());

}

});

add\_cmid.setPromptText("客户ID");

add\_cmname.setPromptText("客户姓名");

add\_cmpasswd.setPromptText("登录密码");

add\_cmember.setPromptText("会员说明");

add\_cmoral.setPromptText("信誉评级");

StuffCustomerlist.clear();

StuffCustomerlist=connector.getAllcustomerinfo();

table\_customer.setItems(StuffCustomerlist);

}

@FXML

//删除客户信息

public void onCustomerDeleteClick() throws SQLException {

StuffCustomer stuffCustomer = table\_customer.getSelectionModel().getSelectedItem();

if(stuffCustomer==null)

{

JOptionPane.showMessageDialog(null,"请选择要删除的信息行！");

return;

}

connector.deleteCusotomerinfo(stuffCustomer.getCusid());

this.show\_customerinfo();

}

//添加客户信息

public void onCustomerAddClick() throws SQLException {

String addcmid=add\_cmid.getText();

String addcmname=add\_cmname.getText();

String addcmpasswd=add\_cmpasswd.getText();

String addmember =add\_cmember.getText();

if(addcmid.isEmpty()||addcmname.isEmpty()||addcmpasswd.isEmpty())

{

JOptionPane.showMessageDialog(null,"请输入完整信息！");

return;

}

connector.addCustomerinfo(addcmid,addcmname,addmember,addcmpasswd);

this.show\_customerinfo();

}

@FXML

//加载员工信息

public void show\_stuffinfo() throws SQLException {

stuffid.setCellValueFactory(new PropertyValueFactory<StuffPerson,String>("Stuffid"));

stuffname.setCellValueFactory(new PropertyValueFactory<StuffPerson,String>("Stuffname"));

stufftitle.setCellValueFactory(new PropertyValueFactory<StuffPerson,String>("Stufftitle"));

stuffpasswd.setCellValueFactory(new PropertyValueFactory<StuffPerson,String>("Stuffpasswd"));

add\_smid.setPromptText("员工ID");

add\_smname.setPromptText("员工姓名");

add\_smtitle.setPromptText("员工头衔");

add\_smpasswd.setPromptText("员工密码");

StuffPersonlist.clear();

StuffPersonlist=connector.getAllstuffinfo(StuffPersonlist);

table\_stuff.setItems(StuffPersonlist);

}

@FXML

public void onStuffDeleteClick() throws SQLException {

StuffPerson stuffPerson = table\_stuff.getSelectionModel().getSelectedItem();

if(stuffPerson==null)

{

JOptionPane.showMessageDialog(null,"请选择要删除的信息行！");

return;

}

connector.deleteStuffinfo(stuffPerson.getStuffid());

this.show\_stuffinfo();

}

@FXML

public void onStuffAddClick() throws SQLException {

String stuffid=add\_smid.getText();

String stuffname=add\_smname.getText();

String stufftitle=add\_smtitle.getText();

String stuffpasswd=add\_smpasswd.getText();

if(stuffid.isEmpty()||stuffname.isEmpty()||stufftitle.isEmpty()||stuffpasswd.isEmpty())

{

JOptionPane.showMessageDialog(null,"请输入完整信息！");

return;

}

connector.addStuffinfo(stuffid,stuffname,stufftitle,stuffpasswd);

add\_smid.clear();

add\_smname.clear();

add\_smpasswd.clear();

add\_smtitle.clear();

}

@FXML

public void show\_diaryinfo() throws SQLException {

stuff\_did.setCellValueFactory(new PropertyValueFactory<StuffDiary,String>("Did"));

stuff\_dcarid.setCellValueFactory(new PropertyValueFactory<StuffDiary,String>("Dcarid"));

stuff\_dcost.setCellValueFactory(new PropertyValueFactory<StuffDiary,String>("Dcost"));

stuff\_dcusid.setCellValueFactory(new PropertyValueFactory<StuffDiary,String>("Dcusid"));

stuff\_devent.setCellValueFactory(new PropertyValueFactory<StuffDiary,String>("Devent"));

stuff\_dstuffid.setCellValueFactory(new PropertyValueFactory<StuffDiary,String>("Dstuffid"));

stuff\_dtime.setCellValueFactory(new PropertyValueFactory<StuffDiary,String>("Dtime"));

StuffDiarylist.clear();

StuffDiarylist=connector.getAlldirayinfo(StuffDiarylist);

table\_diary.setItems(StuffDiarylist);

// add\_did.setPromptText("日志ID");

// add\_dcarid.setPromptText("汽车ID");

// add\_dcusid.setPromptText("顾客ID");

// add\_dstuffid.setPromptText("员工ID");

//add\_dtime.setPromptText("日期");

//add\_devent.setPromptText("事件（是否归还）");

//add\_dcost.setPromptText("交易金额");

}

@FXML

public void onDiaryDeleteClick() throws SQLException {

StuffDiary stuffDiary = table\_diary.getSelectionModel().getSelectedItem();

if(stuffDiary==null)

{

JOptionPane.showMessageDialog(null,"请选择要删除的信息行！");

return;

}

connector.deleteDiaryinfo(stuffDiary.getDid(),stuffDiary.getDevent());

this.show\_diaryinfo();

}

@FXML

public void onDiaryAddClick() throws SQLException {

String did=add\_did.getText();

String dcarid=add\_dcarid.getText();

String dcusid=add\_dcusid.getText();

String dstuffid=add\_dstuffid.getText();

String dtime=add\_dtime.getText();

String devent=add\_devent.getText();

String dcost=add\_dcost.getText();

if(did.isEmpty()||dcarid.isEmpty()||dcusid.isEmpty()||dstuffid.isEmpty()||dtime.isEmpty()||devent.isEmpty()||dcost.isEmpty()) {

JOptionPane.showMessageDialog(null, "请输入完整信息！");

return;

}

connector.addDiaryinfo(did,dcarid,dcusid,dstuffid,dtime,devent,dcost);

this.show\_diaryinfo();

add\_dcarid.clear();

add\_dcost.clear();

add\_dcusid.clear();

add\_devent.clear();

add\_did.clear();

add\_dstuffid.clear();

add\_dtime.clear();

}

@FXML

public void show\_tableinfo() throws SQLException {

pieChartData.clear();

pieChart.getData().clear();

pieChartData = connector.getprofitinfo();

pieChart.setData(pieChartData);

pieChart.setTitle("汽车类别分类饼状图");

pieChart.setClockwise(true);

pieChart.setLabelLineLength(50);

pieChart.setLabelsVisible(true);

pieChart.setStartAngle(180);

final Label caption = new Label("");

caption.setStyle("-fx-font: 24 arial;");

for (final PieChart.Data data : pieChart.getData()) {

data.getNode().addEventHandler(MouseEvent.MOUSE\_PRESSED,

e -> {

double total = 0;

for (PieChart.Data d : pieChart.getData()) {

total += d.getPieValue();

}

caption.setTranslateX(e.getSceneX());

caption.setTranslateY(e.getSceneY());

String text = String.format("%.1f%%", 100 \* data.getPieValue() / total);

caption.setText(text);

pieChart.setAccessibleText(text);

}

);

}

}

@FXML

public void get\_year() throws SQLException {

if(pieChart!=null)

pieChart.getData().clear();

pieChartData.clear();

pieChartData=connector.getprofit\_year(pieChartData);

pieChart.setData(pieChartData);

pieChart.setTitle("年度盈利额");

add\_cmember1.setText(String.valueOf(connector.getprofitall()));

pieChart.setClockwise(true);

pieChart.setLabelLineLength(50);

pieChart.setLabelsVisible(true);

pieChart.setStartAngle(180);

}

@FXML

public void get\_season() throws SQLException {

int []num=new int[4];

num=connector.getprofit\_season(num);

if(pieChart!=null)

pieChart.getData().clear();

pieChartData.add(new PieChart.Data("第一季度",num[0]));

add\_cmember2.setText(String.valueOf(num[0]));

pieChartData.add(new PieChart.Data("第二季度",num[1]));

add\_cmember3.setText(String.valueOf(num[1]));

pieChartData.add(new PieChart.Data("第三季度",num[2]));

add\_cmember4.setText(String.valueOf(num[2]));

pieChartData.add(new PieChart.Data("第四季度",num[3]));

add\_cmember5.setText(String.valueOf(num[3]));

int i=0;

for (PieChart.Data data : pieChartData) {

data.getNode().setStyle(

"-fx-pie-color: #ffd700+i;"

);

i++;

}

pieChart.setData(pieChartData);

pieChart.setTitle("季度盈利柱状图");

pieChart.setClockwise(true);

pieChart.setLabelLineLength(50);

pieChart.setLabelsVisible(true);

pieChart.setStartAngle(180);

}

}

6.CustomerController.java

import emtity.CustomerPerson;

import emtity.customerReturnCar;

import javafx.beans.value.ChangeListener;

import javafx.beans.value.ObservableValue;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.fxml.FXML;

import javafx.fxml.FXMLLoader;

import javafx.fxml.Initializable;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.stage.Stage;

import javax.swing.\*;

import java.io.IOException;

import java.net.URL;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.ResourceBundle;

public class CustomerController implements Initializable{

@FXML

public ComboBox<String> combo\_brand,combo\_statu,combo\_rent,combo\_pledge,combo\_id,combo\_sname;

@FXML

private Button btn\_exit;

@FXML

Label label\_title;

@FXML

private TableColumn<CustomerPerson, String> person\_id,person\_cname,person\_member,person\_dlrq,person\_moral;//对应顾客信息的列

@FXML

private TableColumn<customerReturnCar,String> car\_id,car\_brand,car\_status,car\_rent,car\_pledge,car\_valid;//对应租车信息的列

@FXML

TableView<CustomerPerson> table\_personainfo;//顾客信息列表

@FXML

TableView<customerReturnCar> tableCustomerReturnCar;//还车列表

DBConnector connector;

ObservableList<String> ob\_brand = FXCollections.observableArrayList();//汽车品牌

ObservableList<String> ob\_statu = FXCollections.observableArrayList();//汽车状况

ObservableList<String> ob\_rent = FXCollections.observableArrayList();//汽车租金

ObservableList<String> ob\_pledge = FXCollections.observableArrayList();//汽车押金

ObservableList<String> ob\_id = FXCollections.observableArrayList();

ObservableList<String> ob\_sname = FXCollections.observableArrayList();

ObservableList<CustomerPerson> personList = FXCollections.observableArrayList();

ObservableList<customerReturnCar> ReturnCarList = FXCollections.observableArrayList();

ArrayList<String>bandList,statuList,rentList,pledgeList,IdList,NameList;

public static String customer\_id;

public static String customer\_name;

public static void setCustomer\_id(String id)

{

customer\_id=id;

}

public static void setCustomer\_name(String name)

{

customer\_name=name;

}

@Override

public void initialize(URL url, ResourceBundle resourceBundle) {

connector=DBConnector.getInstance();

bandList = new ArrayList<>();

statuList = new ArrayList<>();

rentList = new ArrayList<>();

pledgeList = new ArrayList<>();

IdList = new ArrayList<>();

NameList = new ArrayList<>();

label\_title.setText("欢迎，"+customer\_name+"登录！租车系统");

//载入信息

try{

ob\_brand=connector.getBandFromCarinfo();

combo\_brand.setItems(ob\_brand);

ob\_sname =connector.getNAMEfromemployeeinfo();

combo\_sname.setItems(ob\_sname);

combo\_sname.getSelectionModel().select(0);

}catch (SQLException e)

{

e.printStackTrace();

}

combo\_brand.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<String>() {

@Override

public void changed(ObservableValue<? extends String> observableValue, String s, String t1) {

try {

String brand = combo\_brand.getSelectionModel().getSelectedItem();

connector.getInfoFromCarinfo(brand,ob\_id,ob\_statu,ob\_pledge,ob\_rent);

combo\_id.setItems(ob\_id);

combo\_id.getSelectionModel().select(0);

} catch (SQLException throwables) {

throwables.printStackTrace();

}

}

});

//确定了汽车的牌号，其租金、押金和车况就一一确定了。

combo\_id.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<String>() {

@Override

public void changed(ObservableValue<? extends String> observableValue, String s, String t1) {

int tempid = combo\_id.getSelectionModel().getSelectedIndex();

String tempId = combo\_id.getSelectionModel().getSelectedItem();

ObservableList<String> ob\_statu\_temp = FXCollections.observableArrayList();

ObservableList<String> ob\_rent\_temp = FXCollections.observableArrayList();

ObservableList<String> ob\_pledge\_temp = FXCollections.observableArrayList();

if(tempId==null)return;

if(tempid>=0)

{

ob\_statu\_temp.add(ob\_statu.get(tempid));

ob\_pledge\_temp.add(ob\_pledge.get(tempid));

ob\_rent\_temp.add(ob\_rent.get(tempid));

combo\_statu.setItems(ob\_statu\_temp);

combo\_pledge.setItems(ob\_pledge\_temp);

combo\_rent.setItems(ob\_rent\_temp);

combo\_rent.getSelectionModel().select(0);

combo\_pledge.getSelectionModel().select(0);

combo\_statu.getSelectionModel().select(0);

}

}

});

}

//退出返回到登录界面

@FXML

public void onExitClick()

{

Stage nextSage=(Stage)btn\_exit.getScene().getWindow();

nextSage.close();

Parent root = null;

try

{

root = FXMLLoader.load(getClass().getResource("login.fxml"));

}catch (IOException e)

{

e.printStackTrace();

}

Scene scene =new Scene(root,800,600);

scene.setRoot(root);

Stage stage = new Stage();

stage.setScene(scene);

stage.show();

}

//租车

@FXML

public void onComfirmClick()

{

String temp\_stuff = combo\_sname.getSelectionModel().getSelectedItem();

String temp\_status = combo\_statu.getSelectionModel().getSelectedItem();

String temp\_rent = combo\_rent.getSelectionModel().getSelectedItem();

String temp\_brand = combo\_brand.getSelectionModel().getSelectedItem();

String temp\_id = combo\_id.getSelectionModel().getSelectedItem();

String temp\_pledge = combo\_pledge.getSelectionModel().getSelectedItem();

int temp\_index = combo\_sname.getSelectionModel().getSelectedIndex();

if(temp\_brand==null||temp\_brand.isEmpty()||temp\_stuff==null||temp\_stuff.isEmpty()||temp\_id==null||temp\_id.isEmpty())

{

JOptionPane.showMessageDialog(null,"信息不全！请填写完整信息");

return;

}

try

{

int stuffid = connector.getIDfromemployeeinfo(temp\_index);

System.out.println(stuffid);

connector.updateDiary(Integer.parseInt(temp\_id),Integer.parseInt(customer\_id),stuffid,Integer.parseInt(temp\_rent));

connector.updatecarinfo(Integer.parseInt(temp\_id),Integer.parseInt(customer\_id),Integer.parseInt(temp\_rent));

JOptionPane.showMessageDialog(null,"租车成功！");

combo\_id.getSelectionModel().clearSelection();

combo\_brand.getSelectionModel().clearSelection();

combo\_rent.getSelectionModel().clearSelection();

combo\_statu.getSelectionModel().clearSelection();

combo\_pledge.getSelectionModel().clearSelection();

}

catch (SQLException E)

{

E.printStackTrace();

}

}

@FXML

public void onClearClick()

{

combo\_statu.getSelectionModel().clearSelection();

combo\_brand.getSelectionModel().clearSelection();

combo\_id.getSelectionModel().clearSelection();

combo\_pledge.getSelectionModel().clearSelection();

combo\_rent.getSelectionModel().clearSelection();

combo\_sname.getSelectionModel().clearSelection();

}

//载入个人信息

@FXML

public void getpersoninfo() {

try{

person\_id.setCellValueFactory(new PropertyValueFactory<CustomerPerson,String>("customer\_id"));

person\_cname.setCellValueFactory(new PropertyValueFactory<CustomerPerson,String>("customer\_name"));

person\_member.setCellValueFactory(new PropertyValueFactory<CustomerPerson,String>("customer\_member"));

person\_dlrq.setCellValueFactory(new PropertyValueFactory<CustomerPerson,String>("customer\_dlrq"));

person\_moral.setCellValueFactory(new PropertyValueFactory<CustomerPerson,String>("customer\_moral"));

personList.clear();

personList=connector.getPersoninfo(Integer.valueOf(customer\_id),personList);

}catch (SQLException e)

{

e.printStackTrace();

}

finally {

table\_personainfo.setItems(personList);

}

}

//载入当前客户租车的信息

@FXML

public void returnCar() throws SQLException {

car\_id.setCellValueFactory(new PropertyValueFactory<customerReturnCar,String>("CarId"));

car\_brand.setCellValueFactory(new PropertyValueFactory<customerReturnCar,String>("CarBrand"));

car\_pledge.setCellValueFactory(new PropertyValueFactory<customerReturnCar,String>("CarPledge"));

car\_status.setCellValueFactory(new PropertyValueFactory<customerReturnCar,String>("CarStatus"));

car\_rent.setCellValueFactory(new PropertyValueFactory<customerReturnCar,String>("CarRent"));

car\_valid.setCellValueFactory(new PropertyValueFactory<customerReturnCar,String>("CarValid"));

ReturnCarList.clear();

ReturnCarList=connector.getCarinfo(Integer.valueOf(customer\_id));

tableCustomerReturnCar.setItems(ReturnCarList);

}

//还车

@FXML

public void onReturncarClick() throws SQLException {

customerReturnCar temp\_return\_car = tableCustomerReturnCar.getSelectionModel().getSelectedItem();

if(temp\_return\_car==null)

{

JOptionPane.showMessageDialog(null,"请选中你要进行退号操作的操作行！");

return;

}

connector.DeleteReturnedCar(Integer.valueOf(customer\_id),temp\_return\_car);

}

}

7.数据库管理类：DBConnector.java

import emtity.\*;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.collections.ObservableSet;

import javafx.scene.chart.PieChart;

import javafx.scene.chart.XYChart;

import javax.management.ObjectName;

import javax.print.DocFlavor;

import javax.swing.\*;

import java.nio.file.LinkOption;

import java.security.interfaces.RSAKey;

import java.sql.\*;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

class DbUtils {

public static void closeConn(Connection conn) {

if (conn != null) {

try {

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

conn = null;

}

}

public static void closeStat(PreparedStatement stat) {

if (stat != null) {

try {

stat.close();

} catch (SQLException e) {

e.printStackTrace();

}

stat = null;

}

}

public static void closeCursor(ResultSet cursor) {

if (cursor != null) {

try {

cursor.close();

} catch (SQLException e) {

e.printStackTrace();

}

cursor = null;

}

}

public static void closeResource(ResultSet cursor, PreparedStatement stat) {

closeCursor(cursor);

closeStat(stat);

}

}

public class DBConnector {

private static DBConnector instance;

private Connection connection;

private PreparedStatement Pstatement;

private DBConnector() {

//将mysql驱动注册到DriverManager中去

/\*

Class.forName() 方法要求JVM查找并加载指定的类到内存中；

\*/

try {

Class.forName("com.mysql.cj.jdbc.Driver");

} catch (ClassNotFoundException e) {

System.out.println("找不到驱动类，加载失败！");

e.printStackTrace();

}

}

//单例模式

public static synchronized DBConnector getInstance() {

if (instance == null)

instance = new DBConnector();

return instance;

}

//创建数据库的连接

public void connectionDatabase(String hostname, String DBname, Integer port, String username, String password) throws SQLException {

String url = "jdbc:mysql://" + hostname + ":" + port + "/" + DBname + "?useUnicode=true&characterEncoding=UTF-8&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC";

connection = DriverManager.getConnection(url, username, password);

}

public Connection getConnection() {

return connection;

}

//检查用户登录是否正确

public boolean CheckPassword(int type, int chooseId, String password) throws SQLException {

String customer\_passwd = "select PASSWD from customerinfo where ID=?";

String employee\_passwd = "select PASSWD from employeeinfo where ID=?";

if (type == 0) {

Pstatement = connection.prepareStatement(customer\_passwd);

Pstatement.setInt(1, chooseId);

} else if (type == 1) {

Pstatement = connection.prepareStatement(employee\_passwd);

Pstatement.setInt(1, chooseId);

} else

return false;

ResultSet resultSet = Pstatement.executeQuery();

if (resultSet.next()) {

String passwd = resultSet.getString("PASSWD");

//密码正确，更新登录时间

if (passwd.equals(password)) {

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

if (type == 0) {

String updatecustomerinfo = "update customerinfo set LOGINTIME = "

+ "'" + df.format(new Date()) + "' where id=?";

Pstatement = connection.prepareStatement(updatecustomerinfo);

Pstatement.setInt(1, chooseId);

} else if (type == 1) {

String updateemployeeinfo = "update employeeinfo set LOGINTIME = "

+ "'" + df.format(new Date()) + "' where id=?";

Pstatement = connection.prepareStatement(updateemployeeinfo);

Pstatement.setInt(1, chooseId);

} else return false;

Pstatement.executeUpdate();

} else {

return false;

}

}

return true;

}

//获得汽车品牌列表

public ObservableList getBandFromCarinfo() throws SQLException {

String sql = "select distinct BRAND from CARINFO";

Pstatement = connection.prepareStatement(sql);

ResultSet resultSet = Pstatement.executeQuery();

String brand;

ObservableList<String> brandList = FXCollections.observableArrayList();

while (resultSet.next()) {

brand = resultSet.getString("BRAND");

brandList.add(brand);

}

return brandList;

}

//获得员工姓名列表

public ObservableList getNAMEfromemployeeinfo() {

ObservableList<String> nameList = FXCollections.observableArrayList();

try {

String sql = "select NAME from EMPLOYEEINFO ";

Pstatement = connection.prepareStatement(sql);

ResultSet resultSet = Pstatement.executeQuery();

String name;

while (resultSet.next()) {

name = resultSet.getString("NAME");

nameList.add(name);

}

} catch (SQLException e) {

e.printStackTrace();

}

return nameList;

}

public void getInfoFromCarinfo(String band, ObservableList id, ObservableList statu, ObservableList pledge, ObservableList rent) throws SQLException {

String sql = "select ID,BRAND,RENT\_PAY,STATUS,PLEDGE\_PAY,VALID FROM CARINFO WHERE BRAND=? AND VALID=0";

Pstatement = connection.prepareStatement(sql);

Pstatement.setString(1, band);

ResultSet resultSet = Pstatement.executeQuery();

id.clear();

statu.clear();

pledge.clear();

rent.clear();

while (resultSet.next()) {

id.add(resultSet.getString("ID"));

statu.add(resultSet.getString("STATUS"));

rent.add(resultSet.getString("RENT\_PAY"));

pledge.add(resultSet.getString("PLEDGE\_PAY"));

}

}

//根据排序进行查找id

public int getIDfromemployeeinfo(int index) throws SQLException {

String sql = "select ID FROM EMPLOYEEINFO ORDER BY ID LIMIT ?,1";

Pstatement = connection.prepareStatement(sql);

Pstatement.setInt(1, index);

ResultSet resultSet = Pstatement.executeQuery();

resultSet.next();

int id = resultSet.getInt("ID");

return id;

}

//更新登录时间

public void updateTimecustomerOremployee(int type, int id) throws SQLException {

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String sql;

if (type == 0) {

sql = "update customerinfo set LOGINTIME = "

+ "'" + df.format(new Date()) + "' where id= ?";

} else if (type == 1) {

sql = "update employeeinfo set LOGINTIME = "

+ "'" + df.format(new Date()) + "' where id= ?";

} else

return;

Pstatement = connection.prepareStatement(sql);

Pstatement.setInt(1, id);

Pstatement.executeUpdate();

}

//更新日志，载入租车信息

public void updateDiary(int carid, int loginid, int stuffid, int rent) throws SQLException {

/\* String sql = "select count(\*) as number from rentdiary";

Pstatement = connection.prepareStatement(sql);

ResultSet resultSet = Pstatement.executeQuery();

resultSet.next();

int number = resultSet.getInt("number");

number--;

String sql2 = "select INFOID from rentdiary ORDER BY INFOID limit ?,1";

Pstatement = connection.prepareStatement(sql2);

Pstatement.setInt(1, number);

resultSet = Pstatement.executeQuery();

resultSet.next();

int infoid = resultSet.getInt("INFOID");

infoid++;\*/

String sql = "select max(infoid) as number from rentdiary";

Pstatement = connection.prepareStatement(sql);

ResultSet resultSet = Pstatement.executeQuery();

resultSet.next();

int infoid = resultSet.getInt("number");

infoid++;

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String sql3 = "insert into rentdiary(INFOID,CARID,CUSTOMERID,STUFFID,RENT\_TIME,REVENT,RENT\_COST) VALUES(?,?,?,?" + ",'" + df.format(new Date()) + "'," + "1,?)";

Pstatement = connection.prepareStatement(sql3);

Pstatement.setInt(1, infoid);

Pstatement.setInt(2, carid);

Pstatement.setInt(3, loginid);

Pstatement.setInt(4, stuffid);

Pstatement.setInt(5, rent);

Pstatement.execute();

}

//租车后，更新汽车信息，同时将汽车计入收入表中

public void updatecarinfo(int carid, int loginid, int rent) throws SQLException {

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String sql = "update carinfo set VALID=1 where id=?";

String sql2 = "update carinfo set LASTLOGIN='" + df.format(new Date()) + "' where id=?";

String sql3 = "update carinfo set CUSTOMERID=? where id=?";

//String sql4 = "select count(\*) as num from profit";

String sql4="select max(pid) pid from profit";

Pstatement = connection.prepareStatement(sql);

Pstatement.setInt(1, carid);

Pstatement.executeUpdate();

Pstatement = connection.prepareStatement(sql2);

Pstatement.setInt(1, carid);

Pstatement.executeUpdate();

Pstatement = connection.prepareStatement(sql3);

Pstatement.setInt(1, loginid);

Pstatement.setInt(2, carid);

Pstatement.executeUpdate();

Pstatement = connection.prepareStatement(sql4);

ResultSet resultSet = Pstatement.executeQuery();

resultSet.next();

int profitid = resultSet.getInt("pid");

profitid++;

String sql5 = "insert into profit values("+profitid+","+"1"+",'"+df.format(new Date())+"',"+rent+")";

Pstatement = connection.prepareStatement(sql5);

Pstatement.setInt(1, profitid);

Pstatement.setInt(2,1);

Pstatement.setDouble(3, rent);

Pstatement.execute();

}

//客户端的客户信息录入

public ObservableList getPersoninfo(int infoid, ObservableList list) throws SQLException {

String id = null, name = null, member = null, dlrq = null, moral = null;

String sql = "select ID,NAME,MEMBER,LOGINTIME FROM customerinfo WHERE ID = ?";

Pstatement = connection.prepareStatement(sql);

Pstatement.setInt(1, infoid);

ResultSet resultSet = Pstatement.executeQuery();

while (resultSet.next()) {

id = resultSet.getString("ID");

name = resultSet.getString("NAME");

String member\_temp = resultSet.getString("MEMBER");

dlrq = resultSet.getString("LOGINTIME");

//获得是否为会员

if (member\_temp.equals("0")) {

member = "否";

} else

member = "是";

}

String sql2 = "select COUNT(REVENT) AS NUM from rentdiary WHERE customerid=?";

Pstatement = connection.prepareStatement(sql2);

Pstatement.setInt(1, infoid);

resultSet = Pstatement.executeQuery();

int temp\_num = 0;

while (resultSet.next()) {

temp\_num = resultSet.getInt("NUM");

}

switch (temp\_num) {

case 0:

moral = "优秀";

break;

case 1:

case 2:

moral = "良好";

break;

case 3:

case 4:

moral = "一般";

break;

default:

moral = "较差";

break;

}

list.add(new CustomerPerson(id, name, member, moral, dlrq));

return list;

}

//客户端汽车信息录入

public ObservableList getCarinfo(int infoid) throws SQLException {

ObservableList<customerReturnCar> list = FXCollections.observableArrayList();

String valid = null, rent = null, id = null, brand = null, status = null, pledge = null, cid = null, lastlogin = null;

String sql = "select \* from carinfo where CUSTOMERID = ?";

Pstatement = connection.prepareStatement(sql);

Pstatement.setInt(1, infoid);

ResultSet resultSet = Pstatement.executeQuery();

while (resultSet.next()) {

id = resultSet.getString("ID");

brand = resultSet.getString("BRAND");

rent = resultSet.getString("RENT\_PAY");

status = resultSet.getString("STATUS");

pledge = resultSet.getString("PLEDGE\_PAY");

String tempvalid = resultSet.getString("VALID");

if (tempvalid.equals("0"))

valid = "否";

else

valid = "是";

System.out.println(id);

System.out.println(brand);

list.add(new customerReturnCar(id, brand, rent, status, pledge, valid));

}

return list;

}

public void DeleteReturnedCar(int loginid, customerReturnCar customerReturnCar) throws SQLException {

//获得租车花费，花费的金额根据租车的时间来计算

String sql = "select lastlogin from carinfo where ID = ?";

Pstatement = connection.prepareStatement(sql);

Pstatement.setInt(1, Integer.parseInt(customerReturnCar.getCarId()));

ResultSet resultSet = Pstatement.executeQuery();

resultSet.next();

String date\_lastlogin = resultSet.getString("lastlogin");

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String currenttime = df.format(new Date());

String sql2 = "select sum(rent\_cost) as num4 from rentdiary where CARID = ? and RENT\_TIME BETWEEN '"+date\_lastlogin+"' AND '"+currenttime+"'";

Pstatement = connection.prepareStatement(sql2);

Pstatement.setInt(1, Integer.parseInt(customerReturnCar.getCarId()));

resultSet = Pstatement.executeQuery();

resultSet.next();

int data\_cost = resultSet.getInt("num4");//租车花费

int addtional\_pay = 0;//额外的费用

int return\_pay = 0;//返回的费用

int pledge\_pay = Integer.parseInt(customerReturnCar.getCarPledge());//租车押金

if (data\_cost > pledge\_pay)//租车的花费大于押金

{

return\_pay = 0;

addtional\_pay = data\_cost - pledge\_pay;

} else {

return\_pay = data\_cost - pledge\_pay;

addtional\_pay = 0;

}

//车损定责没写

//下面针对rentdiary进行补充，将还车信息输入其中

String sql3 = "select max(infoid) as number from rentdiary";

Pstatement = connection.prepareStatement(sql3);

resultSet = Pstatement.executeQuery();

resultSet.next();

int data\_infoID = resultSet.getInt("number");

data\_infoID++;

String sql5 = "select STUFFID FROM rentdiary where infoid = ?";

Pstatement = connection.prepareStatement(sql5);

Pstatement.setInt(1,Integer.parseInt(customerReturnCar.getCarId()));

resultSet = Pstatement.executeQuery();

resultSet.next();

int data\_stuffid = resultSet.getInt("STUFFID");

String sql6 = "insert into rentdiary(infoid,carid,customerid,stuffid,rent\_time,revent,rent\_cost) values (?,?,?,?,'"+currenttime+"',?,?)";

Pstatement = connection.prepareStatement(sql6);

Pstatement.setInt(1, data\_infoID);

Pstatement.setInt(2, Integer.parseInt(customerReturnCar.getCarId()));

Pstatement.setInt(3, loginid);

Pstatement.setInt(4, data\_stuffid);

Pstatement.setInt(5, 2);//2表示还车

Pstatement.setInt(6, data\_cost);

Pstatement.execute();

//更新收入表

//当客户需要支付一定费用时

if (return\_pay != pledge\_pay) {

String sql7 = "select max(pid) as number from profit";

Pstatement = connection.prepareStatement(sql7);

resultSet = Pstatement.executeQuery();

resultSet.next();

int data\_pid = resultSet.getInt("number");

data\_pid++;

int profit\_income = pledge\_pay - return\_pay + addtional\_pay;

String sql9 = "insert into profit(PID,PEVENT,PDATETIME,PPROFIT) VALUES(?,?,'"+currenttime+"',?)";

Pstatement = connection.prepareStatement(sql9);

Pstatement.setInt(1, data\_pid);

Pstatement.setInt(2, 2);

Pstatement.setInt(3,profit\_income);

Pstatement.execute();

}

//修改汽车信息表

String sql10 = "update carinfo set lastlogin=null,valid=0 where id=?";

Pstatement = connection.prepareStatement(sql10);

Pstatement.setInt(1, Integer.parseInt(customerReturnCar.getCarId()));

Pstatement.execute();

//输出信息

if (addtional\_pay == 0) {

JOptionPane.showMessageDialog(null, "还车成功！退还结算后的押金：" + return\_pay);

return;

} else {

JOptionPane.showMessageDialog(null, "还车成功！还需补交金额：" + addtional\_pay);

return;

}

}

//服务端加载全部汽车的信息

public ObservableList getAllCarinfo(ObservableList list ) throws SQLException {

String sql = "select \* from carinfo order by id";

Pstatement = connection.prepareStatement(sql);

ResultSet resultSet = Pstatement.executeQuery();

while (resultSet.next()) {

list.add(new StuffCar(resultSet.getString("ID"), resultSet.getString("BRAND"), resultSet.getString("rent\_pay"), resultSet.getString("status"), resultSet.getString("pledge\_pay"), resultSet.getString("valid")));

}

return list;

}

//服务端下架汽车信息行

public void deleteReturnCar(String carid, String valid) throws SQLException {

if (valid.equals("1")) {

JOptionPane.showMessageDialog(null, "选中的车尚没有归还，无法下架！");

return;

}

String sql = "delete from carinfo where ID = ?";

Pstatement = connection.prepareStatement(sql);

Pstatement.setInt(1, Integer.parseInt(carid));

JOptionPane.showMessageDialog(new JFrame().getContentPane(),

"删除的汽车ID为：" + carid + "!", "通知", JOptionPane.INFORMATION\_MESSAGE);

Pstatement.execute();

}

//服务端上架汽车信息

public void addCarinfo(String carid, String carbrand, String carstatus, String carrent, String carpledge, String carvalid) throws SQLException {

String sql = "select ID from carinfo where id = ?";

Pstatement = connection.prepareStatement(sql);

Pstatement.setInt(1,Integer.parseInt(carid));

ResultSet resultSet=Pstatement.executeQuery();

if(resultSet.next())

{

JOptionPane.showMessageDialog(null,"请勿插入重复的汽车ID！");

return;

}

String sql1="insert into carinfo(ID,BRAND,RENT\_PAY,STATUS,PLEDGE\_PAY,VALID) VALUES (?,?,?,?,?,?)";

Pstatement=connection.prepareStatement(sql1);

Pstatement.setInt(1,Integer.parseInt(carid));

Pstatement.setString(2,carbrand);

Pstatement.setInt(3,Integer.parseInt(carrent));

Pstatement.setInt(4,Integer.parseInt(carstatus));

Pstatement.setInt(5,Integer.parseInt(carpledge));

Pstatement.setInt(6,Integer.parseInt(carvalid));

Pstatement.execute();

}

//服务端加载全部顾客信息

public ObservableList getAllcustomerinfo() throws SQLException {

ObservableList<StuffCustomer> list = FXCollections.observableArrayList();

String sql="select \* from customerinfo order by id";

Pstatement=connection.prepareStatement(sql);

ResultSet resultSet= Pstatement.executeQuery();

while(resultSet.next()) {

String moral = null;

String sql2 = "select count(revent) as num from rentdiary where customerid = ?";

PreparedStatement temp = connection.prepareStatement(sql2);

temp.setInt(1, resultSet.getInt("ID"));

ResultSet resultSet1 = temp.executeQuery();

resultSet1.next();

int num = resultSet1.getInt("num");

switch (num) {

case 0:

moral = "优秀";

break;

case 1:

case 2:

moral = "良好";

break;

case 3:

case 4:

moral = "一般";

break;

default:

moral = "较差";

break;

}

list.add(new StuffCustomer(String.valueOf(resultSet.getInt("ID")), resultSet.getString("NAME"), resultSet.getString("PASSWD"), moral, resultSet.getString("member")));

}

return list;

}

//服务端删除客户信息

public void deleteCusotomerinfo(String cusid) throws SQLException {

String sql ="select id from carinfo where customerid = ?";

Pstatement=connection.prepareStatement(sql);

Pstatement.setInt(1,Integer.parseInt(cusid));

ResultSet resultSet=Pstatement.executeQuery();

if(resultSet.next())

{

JOptionPane.showMessageDialog(null,"该客户正在租用汽车，不能删除！！");

return;

}

String sql2="delete from customerinfo where id = ?";

Pstatement=connection.prepareStatement(sql2);

Pstatement.setInt(1,Integer.parseInt(cusid));

Pstatement.execute();

}

//服务端添加客户信息

public void addCustomerinfo(String cusid,String name,String member,String passwd) throws SQLException {

String sql="select id from customerinfo where id = ?";

Pstatement=connection.prepareStatement(sql);

Pstatement.setInt(1,Integer.parseInt(cusid));

ResultSet resultSet=Pstatement.executeQuery();

if(resultSet.next())

{

JOptionPane.showMessageDialog(null,"请勿插入重复的客户ID！");

return;

}

String sql2="insert into customerinfo(id,name,member,passwd) values (?,?,?,?)";

Pstatement=connection.prepareStatement(sql2);

Pstatement.setInt(1,Integer.parseInt(cusid));

Pstatement.setString(2,name);

Pstatement.setInt(3,Integer.parseInt(member));

Pstatement.setString(4,passwd);

Pstatement.execute();

}

//服务端加载全部员工信息

public ObservableList getAllstuffinfo(ObservableList list) throws SQLException {

String sql="select \* from employeeinfo order by id";

Pstatement=connection.prepareStatement(sql);

ResultSet resultSet=Pstatement.executeQuery();

while(resultSet.next())

{

list.add(new StuffPerson(resultSet.getString("ID"),resultSet.getString("NAME"),resultSet.getString("title"),resultSet.getString("passwd")));

}

return list;

}

public void deleteStuffinfo(String stuffid) throws SQLException {

String sql="delete from employeeinfo where id = ?";

Pstatement=connection.prepareStatement(sql);

Pstatement.setInt(1,Integer.parseInt(stuffid));

Pstatement.execute();

}

public void addStuffinfo(String stuffid,String stuffname,String stufftitle,String stuffpasswd) throws SQLException {

String sql="select \* from employeeinfo where id =?";

Pstatement=connection.prepareStatement(sql);

Pstatement.setInt(1,Integer.parseInt(stuffid));

ResultSet resultSet=Pstatement.executeQuery();

if(resultSet.next())

{

JOptionPane.showMessageDialog(null,"请勿插入重复的员工ID！");

return;

}

String sql2="insert into employeeinfo(id,name,title,passwd) values (?,?,?,?)";

Pstatement=connection.prepareStatement(sql2);

Pstatement.setInt(1,Integer.parseInt(stuffid));

Pstatement.setString(2,stuffname);

Pstatement.setInt(3,Integer.parseInt(stufftitle));

Pstatement.setString(4,stuffpasswd);

Pstatement.execute();

}

public ObservableList getAlldirayinfo(ObservableList list) throws SQLException {

String sql="select \* from rentdiary order by infoid";

Pstatement=connection.prepareStatement(sql);

ResultSet resultSet=Pstatement.executeQuery();

while(resultSet.next())

{

list.add(new StuffDiary(resultSet.getString("INFOID"),resultSet.getString("CARID"),resultSet.getString("CUSTOMERID"),resultSet.getString("STUFFID"),resultSet.getString("RENT\_TIME"),resultSet.getString("REVENT"),resultSet.getString("rent\_cost")));

}

return list;

}

public void deleteDiaryinfo(String infoid,String event) throws SQLException {

if(event.equals("租车")||event.equals("还车"))

{

JOptionPane.showMessageDialog(null,"此时汽车正在租车或还车阶段，无法删除日志！");

return;

}

String sql="delete from rentdiary where infoid = ?";

Pstatement=connection.prepareStatement(sql);

Pstatement.setInt(1,Integer.parseInt(infoid));

Pstatement.execute();

}

public void addDiaryinfo(String infoid,String cusid,String stuffid,String carid,String time,String event,String cost) throws SQLException {

String sql="select \* from rentdiary where infoid =?";

Pstatement=connection.prepareStatement(sql);

Pstatement.setInt(1,Integer.parseInt(infoid));

ResultSet resultSet=Pstatement.executeQuery();

if(resultSet.next())

{

JOptionPane.showMessageDialog(null,"请勿插入重复的日志ID！");

return;

}

String sql2="insert into rentdiary(infoid,carid,customerid,stuffid,rent\_time,revent,rent\_cost) values (?,?,?,?,?,?,?)";

Pstatement=connection.prepareStatement(sql2);

Pstatement.setInt(1,Integer.parseInt(infoid));

Pstatement.setInt(2,Integer.parseInt(carid));

Pstatement.setInt(3,Integer.parseInt(cusid));

Pstatement.setInt(4,Integer.parseInt(stuffid));

Pstatement.setString(5,time);

Pstatement.setString(6,event);

Pstatement.setString(7,cost);

Pstatement.execute();

}

//获得每个季度的收入

public int[] getprofit\_season(int[]a) throws SQLException {

String sql="select sum(pprofit) as sum1 from profit where month(pdatetime)>=1 and month(pdatetime)<=3 order by month(pdatetime)";

String sql2="select sum(pprofit) as sum2 from profit where month(pdatetime)>3 and month(pdatetime)<=6 order by month(pdatetime)";

String sql3="select sum(pprofit) as sum3 from profit where month(pdatetime)>6 and month(pdatetime)<=9 order by month(pdatetime)";

String sql4="select sum(pprofit) as sum4 from profit where month(pdatetime)>9 and month(pdatetime)<=12 order by month(pdatetime)";

Pstatement=connection.prepareStatement(sql);

ResultSet resultSet=Pstatement.executeQuery();

resultSet.next();

a[0]=resultSet.getInt("sum1");

Pstatement=connection.prepareStatement(sql2);

resultSet=Pstatement.executeQuery();

resultSet.next();

a[1]=resultSet.getInt("sum2");

Pstatement=connection.prepareStatement(sql3);

resultSet=Pstatement.executeQuery();

resultSet.next();

a[2]=resultSet.getInt("sum3");

Pstatement=connection.prepareStatement(sql4);

resultSet=Pstatement.executeQuery();

resultSet.next();

a[3]=resultSet.getInt("sum4");

return a;

}

public ObservableList getprofit\_year(ObservableList list) throws SQLException {

String sql="SELECT year(pdatetime) as num,sum(pprofit) as sum from profit group by year(pdatetime) order by year(pdatetime)";

Pstatement=connection.prepareStatement(sql);

ResultSet resultSet=Pstatement.executeQuery();

while(resultSet.next())

{

list.add(new PieChart.Data(resultSet.getString("num"),resultSet.getInt("sum")));

}

return list;

}

public ObservableList getprofitinfo() throws SQLException {

ObservableList<PieChart.Data> list = FXCollections.observableArrayList();

String sql="select BRAND,COUNT(BRAND) AS NUM FROM CARINFO GROUP BY BRAND";

Pstatement=connection.prepareStatement(sql);

ResultSet resultSet=Pstatement.executeQuery();

while(resultSet.next())

{

list.add(new PieChart.Data(resultSet.getString("brand"),resultSet.getInt("NUM")));

}

return list;

}

public double getprofitall() throws SQLException {

String sql="select sum(pprofit) AS NUM FROM profit";

Pstatement=connection.prepareStatement(sql);

ResultSet resultSet=Pstatement.executeQuery();

return resultSet.getDouble("NUM");

}

}

8.实体类：StuffPerson.java

package emtity;

import javafx.beans.property.SimpleStringProperty;

public class StuffPerson {

private final SimpleStringProperty Stuffid;

private final SimpleStringProperty Stuffname;

private final SimpleStringProperty Stufftitle;

private final SimpleStringProperty Stuffpasswd;

public StuffPerson(String Stuffid,String Stuffname,String Stufftitle,String Stuffpasswd)

{

this.Stuffid = new SimpleStringProperty(Stuffid);

this.Stuffname = new SimpleStringProperty(Stuffname);

this.Stufftitle = new SimpleStringProperty(Stufftitle);

this.Stuffpasswd=new SimpleStringProperty(Stuffpasswd);

}

public String getStuffid(){

return this.Stuffid.get();

}

public void setStuffid(String ci) {

this.Stuffid.set(ci);

}

public String getStuffname(){

return this.Stuffname.get();

}

public void setStuffname(String ci) {

this.Stuffname.set(ci);

}

public String getStufftitle(){

return this.Stufftitle.get();

}

public void setStufftitle(String ci) {

this.Stufftitle.set(ci);

}

public String getStuffpasswd(){

return this.Stuffpasswd.get();

}

public void setStuffpasswd(String ci) {

this.Stuffpasswd.set(ci);

}

}

9.实体类：StuffDiary.java

package emtity;

import javafx.beans.property.SimpleStringProperty;

public class StuffDiary {

private final SimpleStringProperty Did;

private final SimpleStringProperty Dcarid;

private final SimpleStringProperty Dcusid;

private final SimpleStringProperty Dstuffid;

private final SimpleStringProperty Dtime;

private final SimpleStringProperty Devent;

private final SimpleStringProperty Dcost;

public StuffDiary(String Did,String Dcarid,String Dcusid,String Dstuffid,String Dtime,String Devent,String Dcost)

{

this.Did=new SimpleStringProperty(Did);

this.Dcarid=new SimpleStringProperty(Dcarid);

this.Dcusid=new SimpleStringProperty(Dcusid);

this.Dstuffid=new SimpleStringProperty(Dstuffid);

this.Dtime=new SimpleStringProperty(Dtime);

this.Devent=new SimpleStringProperty(Devent);

this.Dcost=new SimpleStringProperty(Dcost);

}

public String getDcost(){

return this.Dcost.get();

}

public void setDcost(String ci) {

this.Dcost.set(ci);

}

public String getDid(){

return this.Did.get();

}

public void setDid(String ci) {

this.Did.set(ci);

}

public String getDcarid(){

return this.Dcarid.get();

}

public void setDcarid(String ci) {

this.Dcarid.set(ci);

}

public String getDcusid(){

return this.Dcusid.get();

}

public void setDcusid(String ci) {

this.Dcusid.set(ci);

}

public String getDstuffid(){

return this.Dstuffid.get();

}

public void setDstuffid(String ci) {

this.Dstuffid.set(ci);

}

public String getDtime(){

return this.Dtime.get();

}

public void setDtime(String ci) {

this.Dtime.set(ci);

}

public String getDevent(){

return this.Devent.get();

}

public void setDevent(String ci) {

this.Devent.set(ci);

}

}

10.实体类：StuffCar.java

package emtity;

import javafx.beans.property.SimpleStringProperty;

public class StuffCar {

private final SimpleStringProperty Carid;

private final SimpleStringProperty Carbrand;

private final SimpleStringProperty Carrent\_pay;

private final SimpleStringProperty Carstatus;

private final SimpleStringProperty Carpledge\_pay;

private final SimpleStringProperty Carvalid;

public StuffCar(String Carid, String Carbrand, String Carrent, String Carstatus, String Carpledge, String Carvalid) {

this.Carid = new SimpleStringProperty(Carid);

this.Carbrand = new SimpleStringProperty(Carbrand);

this.Carrent\_pay = new SimpleStringProperty(Carrent);

this.Carstatus = new SimpleStringProperty(Carstatus);

this.Carpledge\_pay = new SimpleStringProperty(Carpledge);

this.Carvalid = new SimpleStringProperty(Carvalid);

}

public String getCarbrand() {

return Carbrand.get();

}

public String getCarid() {

return Carid.get();

}

public String getCarpledge\_pay() {

return Carpledge\_pay.get();

}

public String getCarrent\_pay() {

return Carrent\_pay.get();

}

public String getCarstatus() {

return Carstatus.get();

}

public String getCarvalid() {

return Carvalid.get();

}

public void setCarbrand(String s)

{

this.Carbrand.set(s);

}

public void setCarrent\_pay(String s)

{

this.Carrent\_pay.set(s);

}

public void setCarstatus(String s)

{

this.Carstatus.set(s);

}

public void setCarpledge\_pay(String s)

{

this.Carpledge\_pay.set(s);

}

public void setCarvalid(String s)

{

this.Carvalid.set(s);

}

}

11.实体类：StuffCustomer.java

package emtity;

import javafx.beans.property.SimpleStringProperty;

public class StuffCustomer {

private final SimpleStringProperty Cusid;

private final SimpleStringProperty Cusname;

private final SimpleStringProperty Cusmoral;

private final SimpleStringProperty Cuspasswd;

private final SimpleStringProperty Cusmember;

public StuffCustomer(String Cusid,String Cusname,String passwd,String Cusmoral,String m)

{

this.Cusid = new SimpleStringProperty(Cusid);

this.Cusname = new SimpleStringProperty(Cusname);

this.Cuspasswd = new SimpleStringProperty(passwd);

this.Cusmoral=new SimpleStringProperty(Cusmoral);

this.Cusmember=new SimpleStringProperty(m);

}

public String getCusid() {

return Cusid.get();

}

public String getCusmember() {

return Cusmember.get();

}

public void setCusmember(String s){

this.Cusmember.set(s);

}

public String getCusname() {

return Cusname.get();

}

public String getCuspasswd() {

return Cuspasswd.get();

}

public String getCusmoral() {

return Cusmoral.get();

}

public void setCusid(String s)

{

this.Cusid.set(s);

}

public void setCusname(String s)

{

this.Cusname.set(s);

}

public void setCusmoral(String s)

{

this.Cusmoral.set(s);

}

public void setCuspasswd(String s)

{

this.Cuspasswd.set(s);

}

}

12.实体类：CustomerPerson.java

package emtity;

import javafx.beans.property.SimpleStringProperty;

//顾客的个人信息

public class CustomerPerson {

private final SimpleStringProperty customer\_id;

private final SimpleStringProperty customer\_name;

private final SimpleStringProperty customer\_member;

private final SimpleStringProperty customer\_dlrq;

private final SimpleStringProperty customer\_moral;

public CustomerPerson(String ci,String cname,String cm,String cmoral,String dlrq) {

this.customer\_id = new SimpleStringProperty(ci);

this.customer\_name = new SimpleStringProperty(cname);

this.customer\_member = new SimpleStringProperty(cm);

this.customer\_dlrq = new SimpleStringProperty(dlrq);

this.customer\_moral=new SimpleStringProperty(cmoral);

}

public String getCustomer\_dlrq() {

return customer\_dlrq.get();

}

public String getCustomer\_id() {

return customer\_id.get();

}

public String getCustomer\_member() {

return customer\_member.get();

}

public String getCustomer\_moral() {

return customer\_moral.get();

}

public String getCustomer\_name() {

return customer\_name.get();

}

public void setCustomer\_id(String s)

{

this.customer\_id.set(s);

}

public void setCustomer\_name(String s)

{

this.customer\_name.set(s);

}

public void setCustomer\_member(String s)

{

this.customer\_member.set(s);

}

public void setCustomer\_dlrq(String s)

{

this.customer\_dlrq.set(s);

}

public void setCustomer\_moral(String s)

{

this.customer\_moral.set(s);

}

}

13.实体类：customerReturnCar.java

package emtity;

import javafx.beans.property.SimpleStringProperty;

public class customerReturnCar {

private final SimpleStringProperty CarId;//汽车牌号

private final SimpleStringProperty CarBrand;//汽车品牌

private final SimpleStringProperty CarRent;//租金

private final SimpleStringProperty CarStatus;//汽车状况

private final SimpleStringProperty CarPledge;//押金

private final SimpleStringProperty CarValid;//是否占用

public customerReturnCar(String id,String brand,String rent,String status,String pledge,String valid)

{

this.CarRent = new SimpleStringProperty(rent);

this.CarPledge=new SimpleStringProperty(pledge);

this.CarId=new SimpleStringProperty(id);

this.CarBrand=new SimpleStringProperty(brand);

this.CarValid=new SimpleStringProperty(valid);

this.CarStatus=new SimpleStringProperty(status);

}

public String getCarBrand() {

return CarBrand.get();

}

public String getCarId() {

return CarId.get();

}

public String getCarPledge() {

return CarPledge.get();

}

public String getCarRent() {

return CarRent.get();

}

public String getCarStatus() {

return CarStatus.get();

}

public String getCarValid() {

return CarValid.get();

}

public void setCarId(String id)

{

this.CarId.set(id);

}

public void setCarBrand(String s)

{

this.CarBrand.set(s);

}

public void setCarRent(String s)

{

this.CarRent.set(s);

}

public void setCarStatus(String statuss)

{

this.CarStatus.set(statuss);

}

public void setCarPledge(String s)

{

this.CarPledge.set(s);

}

public void setCarValid(String s)

{

this.CarValid.set(s);

}

}

14.customer.fxml

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.paint.\*?>

<?import javafx.scene.text.\*?>

<?import javafx.scene.control.\*?>

<?import java.lang.\*?>

<?import javafx.scene.layout.\*?>

<SplitPane fx:id="split\_login" dividerPositions="0.15829145728643215" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" onMouseEntered="#getpersoninfo" orientation="VERTICAL" prefHeight="500.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="CustomerController">

<items>

<AnchorPane fx:id="anchorpane\_up" minHeight="0.0" minWidth="0.0" prefHeight="100.0" prefWidth="160.0">

<children>

<Label fx:id="label\_title" layoutX="119.0" layoutY="13.0" prefHeight="27.0" prefWidth="425.0" text="顾客登陆系统">

<font>

<Font name="Arial" size="24.0" />

</font>

</Label>

</children></AnchorPane>

<AnchorPane fx:id="anchorpane\_down" minHeight="0.0" minWidth="0.0" prefHeight="100.0" prefWidth="160.0">

<children>

<TabPane fx:id="tabpane\_reg" layoutY="1.0" prefHeight="400.0" prefWidth="598.0" tabClosingPolicy="UNAVAILABLE">

<tabs>

<Tab fx:id="tab\_personalinformation" text="个人信息">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">

<children>

<Button fx:id="btn\_exit1" layoutX="260.0" layoutY="250.0" mnemonicParsing="false" onAction="#onExitClick" text="退出">

</Button>

<TableView fx:id="table\_personainfo" prefHeight="200.0" prefWidth="598.0">

<columns>

<TableColumn fx:id="person\_id" prefWidth="80.0" text="顾客ID号" />

<TableColumn fx:id="person\_cname" editable="false" prefWidth="130.0" text="顾客姓名" />

<TableColumn fx:id="person\_member" prefWidth="80.0" text="是否会员" />

<TableColumn fx:id="person\_moral" prefWidth="128.0" text="信誉等级" />

<TableColumn fx:id="person\_dlrq" prefWidth="182.0" text="登陆时间" />

</columns>

</TableView>

</children>

</AnchorPane>

</content>

</Tab>

<Tab fx:id="tab\_reg" text="租车">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">

<children>

<Button fx:id="btn\_ok" layoutX="117.0" layoutY="250.0" mnemonicParsing="false" onAction="#onComfirmClick" text="确定">

<font>

<Font name="System Bold" size="15.0" />

</font>

</Button>

<Button fx:id="btn\_clear" layoutX="250.0" layoutY="250.0" mnemonicParsing="false" onAction="#onClearClick" text="清除">

<font>

<Font name="System Bold" size="15.0" />

</font>

</Button>

<Button fx:id="btn\_exit" layoutX="407.0" layoutY="250.0" mnemonicParsing="false" onAction="#onExitClick" text="退出">

<font>

<Font name="System Bold" size="15.0" />

</font>

</Button>

<Label fx:id="label\_brand" layoutX="15.0" layoutY="34.0" text="汽车品牌">

<font>

<Font name="Arial Bold" size="15.0" />

</font>

</Label>

<Label fx:id="label\_id" layoutX="300.0" layoutY="34.0" text="汽车牌号">

<font>

<Font name="Arial Bold" size="15.0" />

</font>

</Label>

<Label fx:id="label\_rent" layoutX="15.0" layoutY="94.0" text="汽车租金">

<font>

<Font name="Arial Bold" size="15.0" />

</font>

</Label>

<Label fx:id="label\_pledge" layoutX="300.0" layoutY="94.0" text="汽车押金">

<font>

<Font name="Arial Bold" size="15.0" />

</font>

</Label>

<Label fx:id="label\_statu" layoutX="15.0" layoutY="156.0" text="车况等级">

<font>

<Font name="Arial Bold" size="15.0" />

</font>

</Label>

<Label fx:id="label\_sname" layoutX="300.0" layoutY="156.0" text="经手员工">

<font>

<Font name="Arial Bold" size="15.0" />

</font>

</Label>

<ComboBox fx:id="combo\_brand" layoutX="85.0" layoutY="21.0" prefHeight="42.0" prefWidth="100.0" />

<ComboBox fx:id="combo\_id" layoutX="370.0" layoutY="21.0" prefHeight="42.0" prefWidth="100.0" />

<ComboBox fx:id="combo\_rent" layoutX="85.0" layoutY="80.0" prefHeight="43.0" prefWidth="100.0" />

<ComboBox fx:id="combo\_pledge" layoutX="370.0" layoutY="82.0" prefHeight="41.0" prefWidth="100.0" />

<ComboBox fx:id="combo\_statu" layoutX="85.0" layoutY="144.0" prefHeight="40.0" prefWidth="100.0" />

<ComboBox fx:id="combo\_sname" layoutX="370.0" layoutY="143.0" prefHeight="43.0" prefWidth="100.0" />

</children>

</AnchorPane>

</content>

</Tab>

<Tab fx:id="tab\_reutn" onSelectionChanged="#returnCar" text="还车">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">

<children>

<Button fx:id="btn\_ok3" layoutX="207.0" layoutY="250.0" mnemonicParsing="false" onAction="#onReturncarClick" text="还车">

</Button>

<Button fx:id="btn\_exit2" layoutX="332.0" layoutY="250.0" mnemonicParsing="false" onAction="#onExitClick" text="退出">

</Button>

<TableView fx:id="tableCustomerReturnCar" prefHeight="200.0" prefWidth="598.0">

<columns>

<TableColumn fx:id="car\_id" prefWidth="100.0" text="汽车ID" />

<TableColumn fx:id="car\_brand" editable="false" prefWidth="103.0" text="汽车品牌" />

<TableColumn fx:id="car\_status" prefWidth="91.0" text="车况" />

<TableColumn fx:id="car\_rent" prefWidth="96.0" text="租金" />

<TableColumn fx:id="car\_pledge" prefWidth="81.0" text="押金" />

<TableColumn fx:id="car\_valid" prefWidth="129.0" text="是否占用" />

</columns>

</TableView>

</children>

</AnchorPane>

</content>

</Tab>

</tabs>

</TabPane>

</children>

</AnchorPane>

</items>

</SplitPane>

15.login.fxml

<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.\*?>

<?import javafx.scene.text.\*?>

<?import javafx.scene.control.\*?>

<?import javafx.scene.layout.\*?>

<?import javafx.scene.text.Font?>

<SplitPane fx:id="split\_login" dividerPositions="0.15829145728643215" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" orientation="VERTICAL" prefHeight="500.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="loginController">

<items>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="100.0" prefWidth="160.0">

<children>

<Label fx:id="logintitle" layoutX="221.0" layoutY="17.0" prefHeight="42.0" prefWidth="203.0" text="租 车 系 统">

<font>

<Font name="Arial Bold" size="36.0" />

</font>

</Label>

</children></AnchorPane>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="400.0" prefWidth="160.0">

<children>

<Button fx:id="loginButton" layoutX="289.0" layoutY="249.0" mnemonicParsing="false" onAction="#onLoginClick" prefHeight="42.0" prefWidth="93.0" text="登录">

</Button>

<Label fx:id="label\_account" layoutX="146.0" layoutY="99.0" text="账号">

<font>

<Font name="Arial Bold" size="18.0" />

</font>

</Label>

<Label fx:id="label\_pass" layoutX="146.0" layoutY="158.0" text="密码">

<font>

<Font name="Arial Bold" size="18.0" />

</font>

</Label>

<PasswordField fx:id="text\_password" layoutX="198.0" layoutY="152.0" prefHeight="28.0" prefWidth="274.0" />

<ComboBox fx:id="Combo\_account" layoutX="198.0" layoutY="94.0" prefHeight="28.0" prefWidth="274.0" />

<TitledPane animated="false" layoutX="45.0" layoutY="-31.0" text="untitled" />

<Label fx:id="label\_account1" layoutX="117.0" layoutY="38.0" text="登录类型">

<font>

<Font name="Arial Bold" size="18.0" />

</font>

</Label>

<ComboBox fx:id="Combo\_type" layoutX="198.0" layoutY="34.0" prefHeight="28.0" prefWidth="274.0" />

</children></AnchorPane>

</items>

</SplitPane>

16.stuff.fxml

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.chart.\*?>

<?import javafx.scene.paint.\*?>

<?import javafx.scene.text.\*?>

<?import javafx.scene.control.\*?>

<?import java.lang.\*?>

<?import javafx.scene.layout.\*?>

<?import javafx.scene.chart.PieChart?>

<?import javafx.scene.Group?>

<?import javafx.scene.chart.BarChart?>

<SplitPane fx:id="split\_login" dividerPositions="0.15829145728643215" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" orientation="VERTICAL" prefHeight="580.0" prefWidth="750.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="StuffController">

<items>

<AnchorPane fx:id="anchorpane\_up" minHeight="0.0" minWidth="0.0" prefHeight="100.0" prefWidth="750.0">

<children>

<Label fx:id="label\_title" layoutX="262.0" layoutY="34.0" text="租车行内部系统">

<font>

<Font size="24.0" />

</font></Label>

</children>

</AnchorPane>

<AnchorPane fx:id="anchorpane\_down" minHeight="0.0" minWidth="0.0" prefHeight="580.0" prefWidth="750.0">

<children>

<TabPane fx:id="tabpane\_reg" layoutY="1.0" prefHeight="580.0" prefWidth="750.0" tabClosingPolicy="UNAVAILABLE">

<tabs>

<Tab fx:id="tab\_carinfo" onSelectionChanged="#show\_carinfo" text="车辆信息">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="580.0" prefWidth="750.0">

<children>

<Button fx:id="btn\_exit" layoutX="570.0" layoutY="380.0" onAction="#onExitClick" text="退出" />

<Button fx:id="btn\_deletecar" layoutX="345" layoutY="380.0" onAction="#onCarDeleteClick" text="删除" />

<Button fx:id="add\_car" layoutX="560.0" layoutY="260.0" onAction="#onCarAddClick" text="添加" />

<TextField fx:id="add\_carid" layoutY="260.0" prefHeight="30" prefWidth="100" />

<TextField fx:id="add\_carbrand" layoutX="100.0" layoutY="260.0" prefHeight="30" prefWidth="120" />

<TextField fx:id="add\_carstatus" layoutX="220.0" layoutY="260.0" prefHeight="30" prefWidth="100" />

<TextField fx:id="add\_carrent" layoutX="320.0" layoutY="260.0" prefHeight="30" prefWidth="120" />

<TextField fx:id="add\_carpledge" layoutX="440.0" layoutY="260.0" prefHeight="30" prefWidth="120" />

<TableView fx:id="table\_car" editable="true" prefHeight="250.0" prefWidth="750.0">

<columns>

<TableColumn fx:id="stuff\_carid" editable="true" prefWidth="100.0" text="汽车ID" />

<TableColumn fx:id="stuff\_carbrand" editable="true" prefWidth="120.0" text="汽车品牌" />

<TableColumn fx:id="stuff\_carstatus" editable="true" prefWidth="100.0" text="汽车状态" />

<TableColumn fx:id="stuff\_carrent" editable="true" prefWidth="120.0" text="汽车租金" />

<TableColumn fx:id="stuff\_carpledge" editable="true" prefWidth="120.0" text="汽车押金" />

<TableColumn fx:id="stuff\_carvalid" editable="false" prefWidth="120.0" text="租借情况" />

</columns>

</TableView>

</children>

</AnchorPane>

</content>

</Tab>

<Tab fx:id="tab\_customerinfo" onSelectionChanged="#show\_customerinfo" text="客户信息">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="580.0" prefWidth="750.0">

<children>

<Button fx:id="btn\_exit1" layoutX="470.0" layoutY="380.0" onAction="#onExitClick" text="退出" />

<Button fx:id="btn\_delete11" layoutX="155.0" layoutY="380.0" onAction="#onCustomerDeleteClick" text="删除客户信息" />

<Button fx:id="add\_cm" layoutX="635.0" layoutY="310.0" onAction="#onCustomerAddClick" prefHeight="30.0" prefWidth="111.0" text="添加客户信息" />

<TableView fx:id="table\_customer" editable="true" layoutX="-2.0" layoutY="2.0" prefHeight="309.0" prefWidth="750.0">

<columns>

<TableColumn fx:id="stuff\_cusid" editable="true" prefWidth="123.0" sortType="DESCENDING" text="客户ID" />

<TableColumn fx:id="stuff\_cusname" editable="true" prefWidth="138.0" text="客户名" />

<TableColumn fx:id="stuff\_cuspasswd" editable="true" prefWidth="98.0" text="登录密码" />

<TableColumn fx:id="stuff\_cusmoral" editable="true" prefWidth="162.0" text="信誉评级" />

<TableColumn fx:id="stuff\_cusmember" editable="true" prefWidth="113.0" text="是否会员" />

</columns>

</TableView>

<TextField fx:id="add\_cmid" layoutY="310.0" prefHeight="30" prefWidth="124.0" />

<TextField fx:id="add\_cmname" layoutX="124.0" layoutY="310.0" prefHeight="30" prefWidth="141.0" />

<TextField fx:id="add\_cmpasswd" layoutX="260.0" layoutY="310.0" prefHeight="30" prefWidth="105.0" />

<TextField fx:id="add\_cmoral" layoutX="357.0" layoutY="310.0" prefHeight="30" prefWidth="170.0" />

<TextField fx:id="add\_cmember" layoutX="521.0" layoutY="310.0" prefHeight="30" prefWidth="115.0" />

</children>

</AnchorPane>

</content>

</Tab>

<Tab fx:id="tab\_stuffinfo" onSelectionChanged="#show\_stuffinfo" text="员工信息">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="580.0" prefWidth="750.0">

<children>

<Button fx:id="btn\_exit3" layoutX="470.0" layoutY="380.0" onAction="#onExitClick" text="退出" />

<Button fx:id="btn\_stuffdelete" layoutX="320.0" layoutY="380.0" onAction="#onStuffDeleteClick" text="删除员工" />

<Button fx:id="btn\_stuffadd" layoutX="420.0" layoutY="260.0" onAction="#onStuffAddClick" text="添加员工" />

<TableView fx:id="table\_stuff" editable="true" prefHeight="250.0" prefWidth="750.0">

<columns>

<TableColumn fx:id="stuffid" editable="true" prefWidth="100.0" text="员工ID" />

<TableColumn fx:id="stuffname" editable="true" prefWidth="120.0" text="员工姓名" />

<TableColumn fx:id="stufftitle" editable="true" prefWidth="100.0" text="员工级别" />

<TableColumn fx:id="stuffpasswd" editable="true" prefWidth="100.0" text="员工密码" />

</columns>

</TableView>

<TextField fx:id="add\_smid" layoutY="260.0" prefHeight="30" prefWidth="100" />

<TextField fx:id="add\_smname" layoutX="100.0" layoutY="260.0" prefHeight="30" prefWidth="120" />

<TextField fx:id="add\_smtitle" layoutX="220.0" layoutY="260.0" prefHeight="30" prefWidth="100" />

<TextField fx:id="add\_smpasswd" layoutX="320.0" layoutY="260.0" prefHeight="30" prefWidth="100" />

</children>

</AnchorPane>

</content>

</Tab>

<Tab fx:id="tab\_diaryinfo" onSelectionChanged="#show\_diaryinfo" text="日志记录">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="377.0" prefWidth="626.0">

<children>

<Button fx:id="btn\_exit4" layoutX="250.0" layoutY="380.0" onAction="#onExitClick" text="退出" />

<TableView fx:id="table\_diary" editable="true" layoutX="-1.0" prefHeight="275.0" prefWidth="702.0">

<columns>

<TableColumn fx:id="stuff\_did" editable="true" prefWidth="100.0" text="日志编号" />

<TableColumn fx:id="stuff\_dcarid" editable="true" prefWidth="100.0" text="汽车ID" />

<TableColumn fx:id="stuff\_dcusid" editable="true" prefWidth="100.0" text="顾客ID" />

<TableColumn fx:id="stuff\_dstuffid" editable="true" prefWidth="100.0" text="经手人ID" />

<TableColumn fx:id="stuff\_dtime" editable="true" prefWidth="100.0" text="发生时间" />

<TableColumn fx:id="stuff\_devent" editable="true" prefWidth="100.0" text="归还" />

<TableColumn fx:id="stuff\_dcost" editable="true" prefWidth="100.0" text="交易金额" />

</columns>

</TableView>

</children>

</AnchorPane>

</content>

</Tab>

<Tab fx:id="tab\_chart" onSelectionChanged="#show\_tableinfo" text="报表显示">

<content>

<AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="580.0" prefWidth="750.0" stylesheets="stuff.css">

<children>

<PieChart id="pieChart" fx:id="pieChart" layoutX="30" layoutY="0" prefHeight="400" prefWidth="500" />

<Button fx:id="BATCHAR\_YEAR" layoutX="522.0" layoutY="20.0" mnemonicParsing="false" onAction="#get\_year" text="年度报表" />

<Button fx:id="BATCHAR\_season" layoutX="633.0" layoutY="20.0" mnemonicParsing="false" onAction="#get\_season" text="季度报表" />

<PieChart id="getyear" fx:id="get\_year" layoutX="30" layoutY="0" prefHeight="400" prefWidth="500" />

<PieChart id="getseason" fx:id="get\_season" layoutX="30" layoutY="0" />

<TextField fx:id="add\_cmember1" layoutX="616.0" layoutY="82.0" prefHeight="30" prefWidth="115.0" promptText="年度收入" />

<TextField fx:id="add\_cmember2" layoutX="616.0" layoutY="214.0" prefHeight="30" prefWidth="115.0" promptText="第一季度收入" />

<TextField fx:id="add\_cmember3" layoutX="616.0" layoutY="159.0" prefHeight="30" prefWidth="115.0" promptText="第二季度收入" />

<TextField fx:id="add\_cmember4" layoutX="616.0" layoutY="328.0" prefHeight="30" prefWidth="115.0" promptText="第三季度收入" />

<TextField fx:id="add\_cmember5" layoutX="616.0" layoutY="271.0" prefHeight="30" prefWidth="115.0" promptText="第四季度收入" />

<Label fx:id="label\_title1" layoutX="554.0" layoutY="90.0" text="全部收入">

<font>

<Font size="12.0" />

</font>

</Label>

<Label fx:id="label\_title11" layoutX="542.0" layoutY="167.0" text="第一季度收入">

<font>

<Font size="12.0" />

</font>

</Label>

<Label fx:id="label\_title111" layoutX="542.0" layoutY="222.0" text="第二季度收入">

<font>

<Font size="12.0" />

</font>

</Label>

<Label fx:id="label\_title112" layoutX="542.0" layoutY="279.0" text="第三季度收入">

<font>

<Font size="12.0" />

</font>

</Label>

<Label fx:id="label\_title113" layoutX="542.0" layoutY="336.0" text="第四季度收入">

<font>

<Font size="12.0" />

</font>

</Label>

</children>

</AnchorPane>

</content>

</Tab>

</tabs>

</TabPane>

</children>

</AnchorPane>

</items>

</SplitPane>

17.stuff.css

.root{

display:block;

}

.root{

-fx-font-family: "微软雅黑";

-fx-background-size: stretch stretch;

-fx-background-position: center center;

}

.button {

-fx-text-fill: white;

-fx-font-size: 20px;

-fx-font-family: "Arial";

-fx-font-weight: bold;

-fx-background-color: linear-gradient(#61a2b1, #2A5058);

-fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );

}

.button:hover {

-fx-background-color: lightskyblue;

}

#pieChart

.default-color0.chart-pie {

-fx-pie-color: #dd3a2e;

}

.default-color1.chart-pie {

-fx-pie-color: #ff9e2c;

}

.default-color2.chart-pie {

-fx-pie-color: #ffdd40;

}

.default-color3.chart-pie {

-fx-pie-color: #9abd57;

}

.default-color4.chart-pie {

-fx-pie-color: #2bc9f7;

}

.default-color5.chart-pie {

-fx-pie-color: #1cabbb;

}

.default-color6.chart-pie {

-fx-pie-color: #2b73f7;

}

.default-color7.chart-pie {

-fx-pie-color: #9760f1;

}

.root {

-fx-font-family: "微软雅黑";

-fx-background-size: stretch stretch;

-fx-background-position: center center;

}