****

**《数据库实训》 课程报告**

**(2019-2019 学年第2 学期)**

**题目:火车票销售系统**

**提交日期： 2019 年 8 月 24 日**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学 院** | **软件** | | **专业班级** | **软件工程** | |
| **课程名称** | **数据库实训** | | **任课教师** | **曾兵** | |
| **人数** | **1** | |  |  | |
| **学 号** | | **学生姓名** | | | **分 数** |
| **201730321094** | | **赖迪霖** | | |  |
| **教师评语：**  **教师签名：** | | | | | |

目录

**需求分析**

1.1 用户需求说明

1.1.1 数据需求

信息需求，即在系统中需要处理哪些数据。根据对火车票销售系统的分析，本系统的信息需求如表1-1所示。

表1-1信息输入输出

|  |  |  |
| --- | --- | --- |
| 信息输入 | 信息处理 | 信息输出 |
| 用户输入所需订购的车票信息 | （1）检索系统数据库，  （2）更新系统中的订票信息 | 符合用户需求的车次及相关  信息  操作反馈（如订票成功等） |
| 用户输入个人信息 | 将新的用户信息存入系统数据库 | 操作反馈（如注册成功或修改信息成功等） |
| 用户输入交易的订单号 | （1）检索系统数据库  （2）记录，删除用户请求的订票  （3）记录，更新系统数据库 | 符合用户需求的订票信息  操作反馈（如退票成功等） |
| 用户输入自己的身份证和密码 | 用户登录系统 | 操作反馈（如登陆成功等） |

1.1.2 事务需求

1.查询

分为对车票信息的查询和客户对已订车票信息的查询。要求：

1）对车次的查询，可以按发车时间、出发地与目的站进行查询；

2）车票信息包括：车票号，火车号、出发站、目的站、出发日期、出发时间、到达时间、票价、座位类型、车票类型、余量。

3）车票座位类型的选择查询。

4）车票信息只允许用户查询，不能修改。

2.订票

通过查询系统，客户根据自己的需求找到满意的车次，通过小程序订票确定已预订选中的车票。要求：

订票记录应包括：订单号、车票号、发车日期、订购日期、客户身份证号。

3.退票

可退票，通过查询系统，客户可以根据自己的客户号码找到自己的订票信息，通过退票模块退去已购车票。

1.2 系统需求说明

为了保证系统能够长期、安全、稳定、可靠、高效的运行，本系统应该满足以下的性能需求。

1.准确性和及时性

系统处理的准确性和及时性是系统的必要性能。系统应能及时而且准确的根据用户权限及所输入的信息做出响应。由于本系统的查询功能对于整个系统的功能和性能完成举足轻重。作为系统的很多数据来源，而车票的数量和时间又影响用户的决策活动，其准确性和及时性很大程度上决定了系统的成败。在系统开发过程中，必须采用一定的方法保证系统的准确性和及时性。

2.易用性

本系统是直接面对用户的，而用户往往对计算机并不是非常熟悉。这就要求系统能够提供良好的用户接口，易用的人机交互界面。要实现这一点，就要求系统应该尽量使用用户熟悉的术语和中文信息的界面，从而保证系统的易用性。

3.安全性

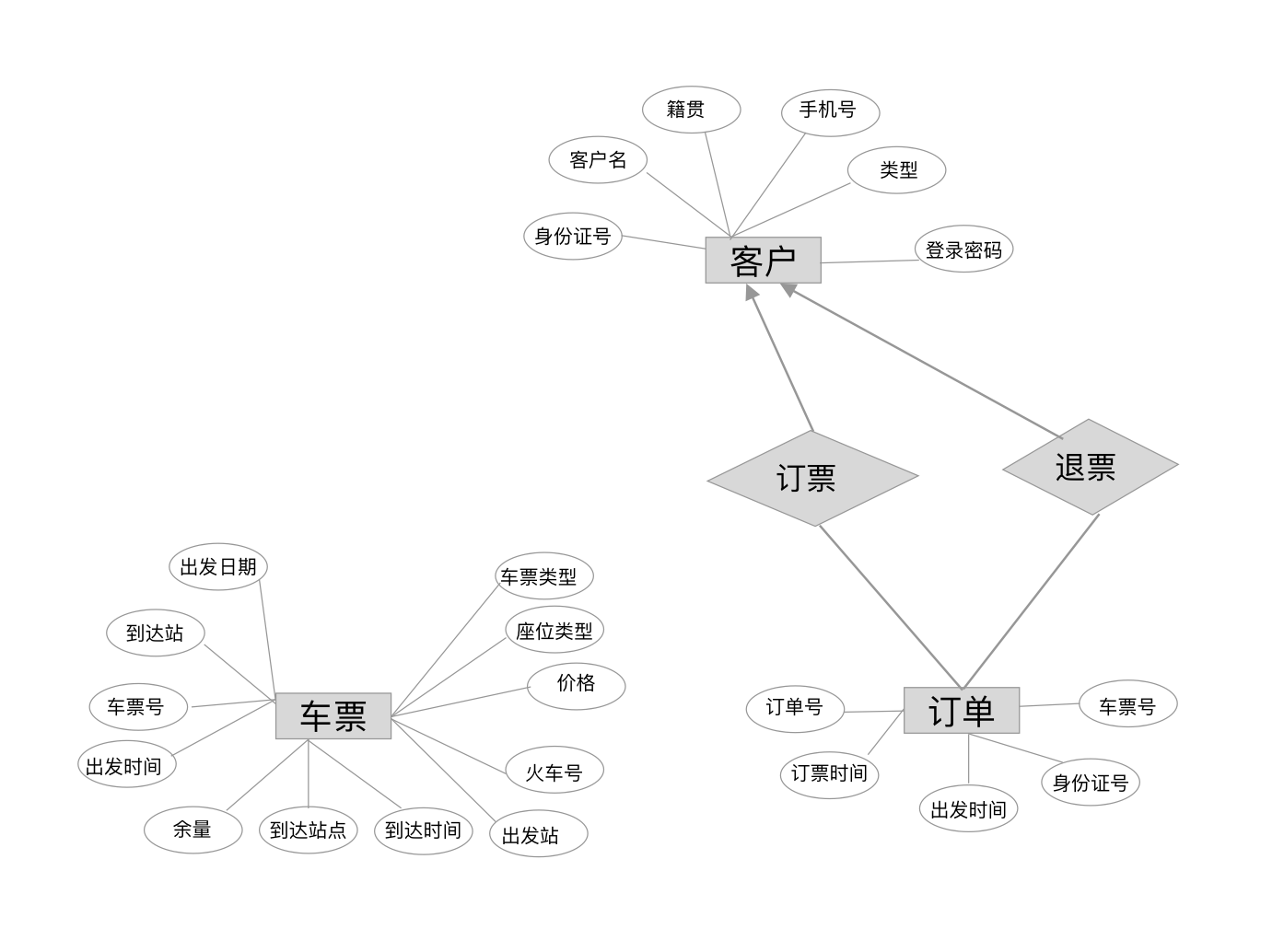
网上订票系统中涉及到的数据是客运公司相当重要的信息，系统要保证用户的权限，对于车次等信息用户只享有查询服务，不得更改；系统还要提供方便的手段供系统维护人员进行数据备份、日常安全管理、以及系统意外崩溃时数据的恢复等工作。同时系统还要保证对数据库进行及时更新，保证数据一致性。

**数据库设计**

1. 关系模式的设计:
2. 分析系统存在的实体、属性、以及实体间的联系.

|  |  |  |
| --- | --- | --- |
| **实体** | **联系** | **实体** |
| 用户 | 购买 | 车票 |
| 管理员 | 修改 | 车票 |
| 用户 | 退订 | 车票 |

1. 画出ER模型图



1. 把ER模型图转换为关系表（关系表格式见下）, 说明转换的依据.

表1 client表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 备注 |
| id | bigint | 18 | 用户id，主码 |
| name | varchar | 50 | 用户名 |
| hometown | varchar | 18 | 籍贯 |
| telphone | varchar | 11 | 电话 |
| type | varchar | 18 | 学生票或成人票 |
| password | varchar | 20 | 用户登录密码 |

表2 ticket表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 备注 |
| ticket\_id | varchar | 18 | 车票号，主码 |
| start\_station | varchar | 11 | 出发站，非空 |
| arrive\_station | varchar | 16 | 到达站，非空 |
| arrive\_time | timestamp |  | 到达时间 |
| go\_date | date |  | 出发日期 |
| go\_time | time |  | 出发时间 |
| price | varchar | 11 | 价格 |
| train\_number | varchar | 10 | 车票号 |
| remain | int | 11 | 余量 |
| seat\_type | varchar | 18 | 硬座或软座 |
| type | varchar | 18 | 学生票或成人票 |

表3 orders表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 备注 |
| order\_number | varchar | 18 | 订单编号，主码 |
| order\_time | timestamp |  | 订票时间 |
| go\_date | timestamp |  | 出发时间 |
| client\_id | bigint | 20 | 用户id，外键 |
| ticket\_id | varchar | 18 | 票id，外键 |

1. 设计主、外码, 完整性约束, 并论证它们的合理性.
2. client表的主码是client\_id，它标识唯一的客户
3. ticket表的主码是ticket\_id, 它标识唯一的车票
4. orders表的主码是order\_number,它标识唯一的订单；外码是client\_id和ticket\_id
5. 非空约束：orders中的go\_date,ticket中的start\_station等，用于保证数据非空
6. 描述每一个基本表关系，证明它属于什么范式。 探讨并证明它们是否可以进一步优化, 如果可以, 那么给出优化后的结果.
7. 订票表

a.因为订票中的任何一个非主属性都不传递依赖于它的任何一个候选码关键字，所以它属于3NF

b.因为客户和订单是一对多的关系，所以优化后，订票表可删除，然后client表中的属性加到订单表中，合成新的表。

1. 退票表

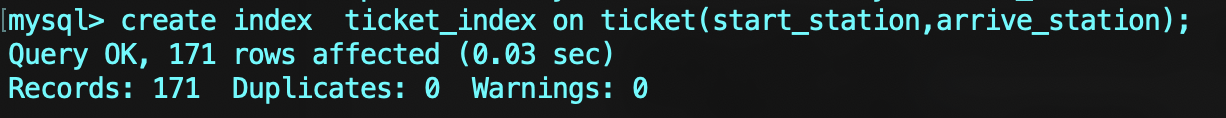
a.因为退票中的任何一个非主属性都不传递依赖于它的任何一个候选码关键字，所以它属于3NF

b.因为客户和退票是一对多的关系，所以优化后，退票表可删除，然后client表中的属性加到订单表中，合成新的表。

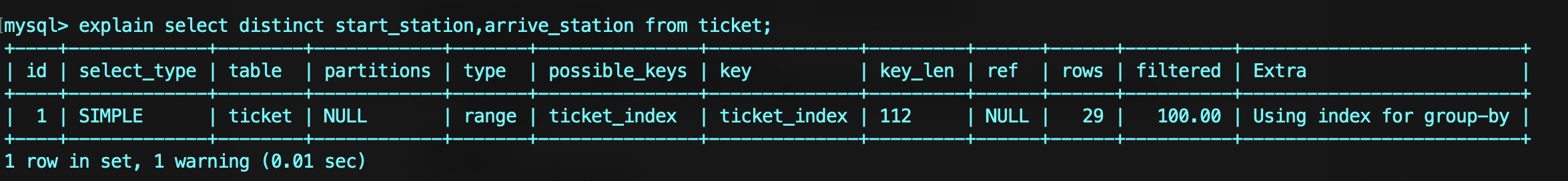
1. 索引设计:

设计索引, 结合物理存贮结构和查询处理过程, 论证它们的合理性.

（1）创建出发站和终点站的索引



（2）检验索引的正确性



合理性验证：

物理结构：非聚簇索引、稠密索引

优点

1.大大加快数据的检索速度

2.在使用分组和排序子句进行数据检索时，可以显著减少查询中分组和排序的时间。

缺点

1.索引需要占物理空间。

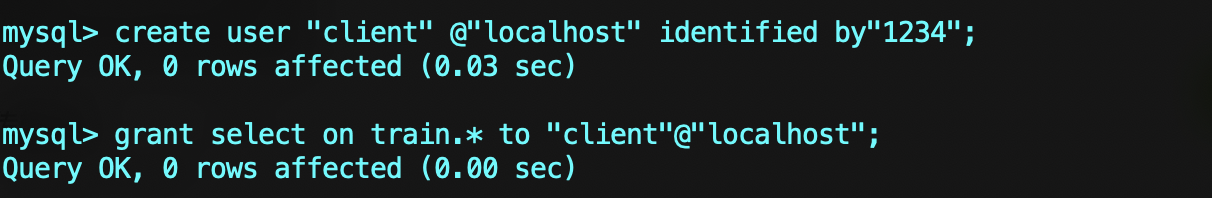
2.当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，降低了数据的维护速度。;

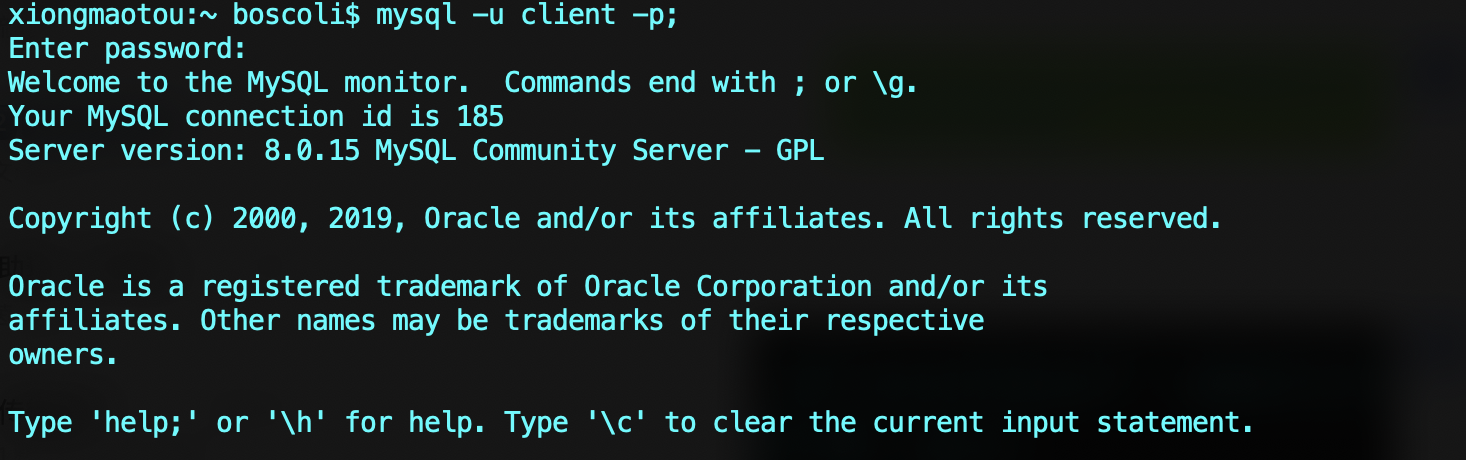
1. 安全性控制

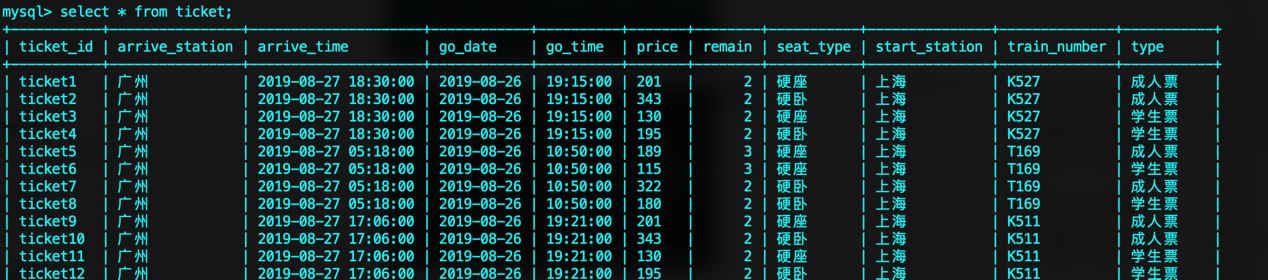
在本火车票售票管理系统中，我们建立了两种类型的安全机制，系统安全和数据安全。在系统安全的建设方面，不允许以游客身份访问本系统，所有用户都必须注册并登陆，登录时会验证用户名和密码。只有两者匹配时，才可访问本系统。在系统安全方面实现了对方法的封装，能保证系统和数据两方面的安全。在数据安全方面， 数据库对象的访问和使用有严格的控制，其中的某些表只有具有特定权限才可以访问。

1. 设计数据库用户和角色, 并分配权限. 论证设计的合理性.

（1）创建数据库普通用户并赋予只能查看表的权限

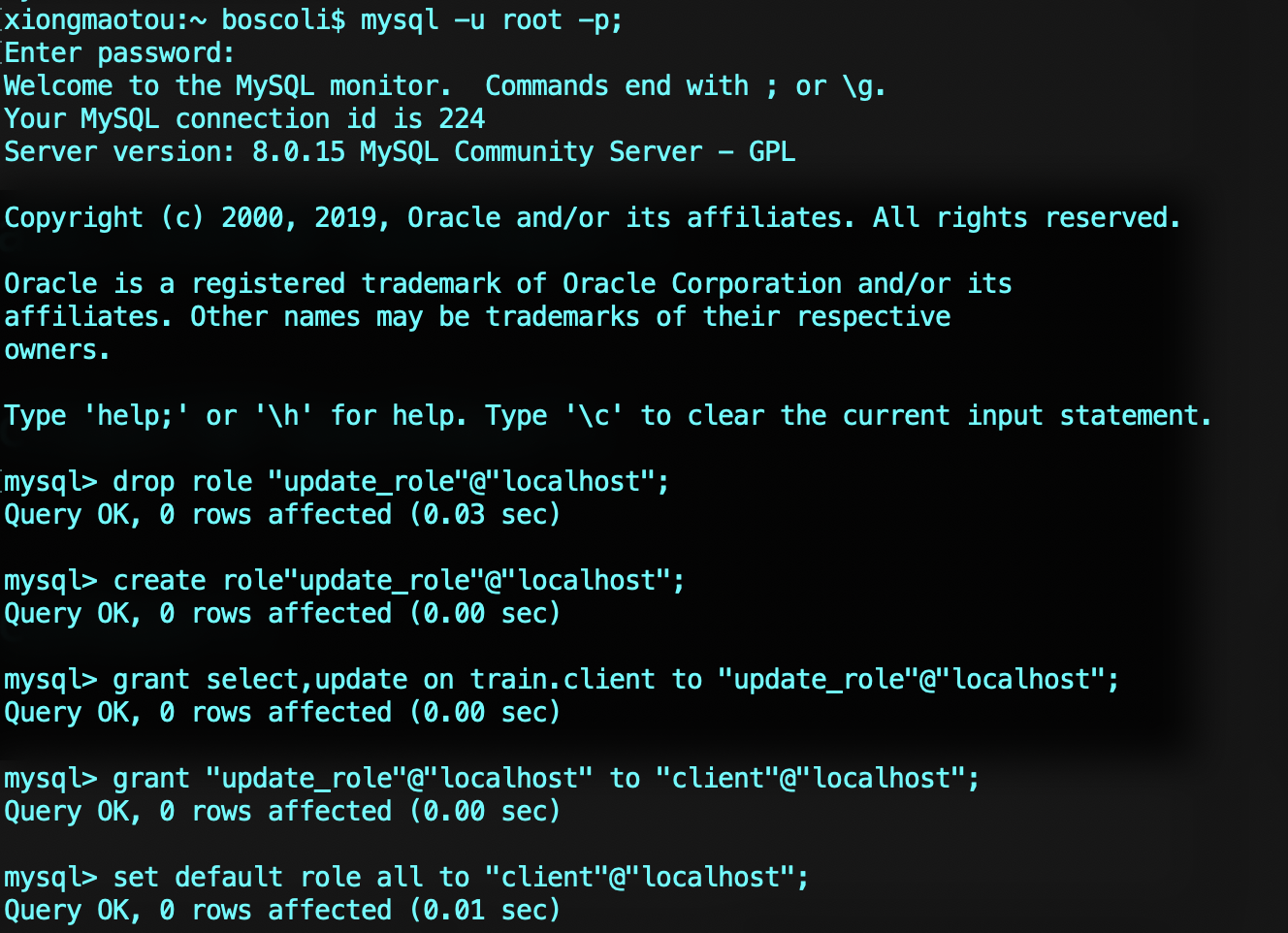


（2）检查普通用户是否创建成功并登陆，结果显示登陆成功

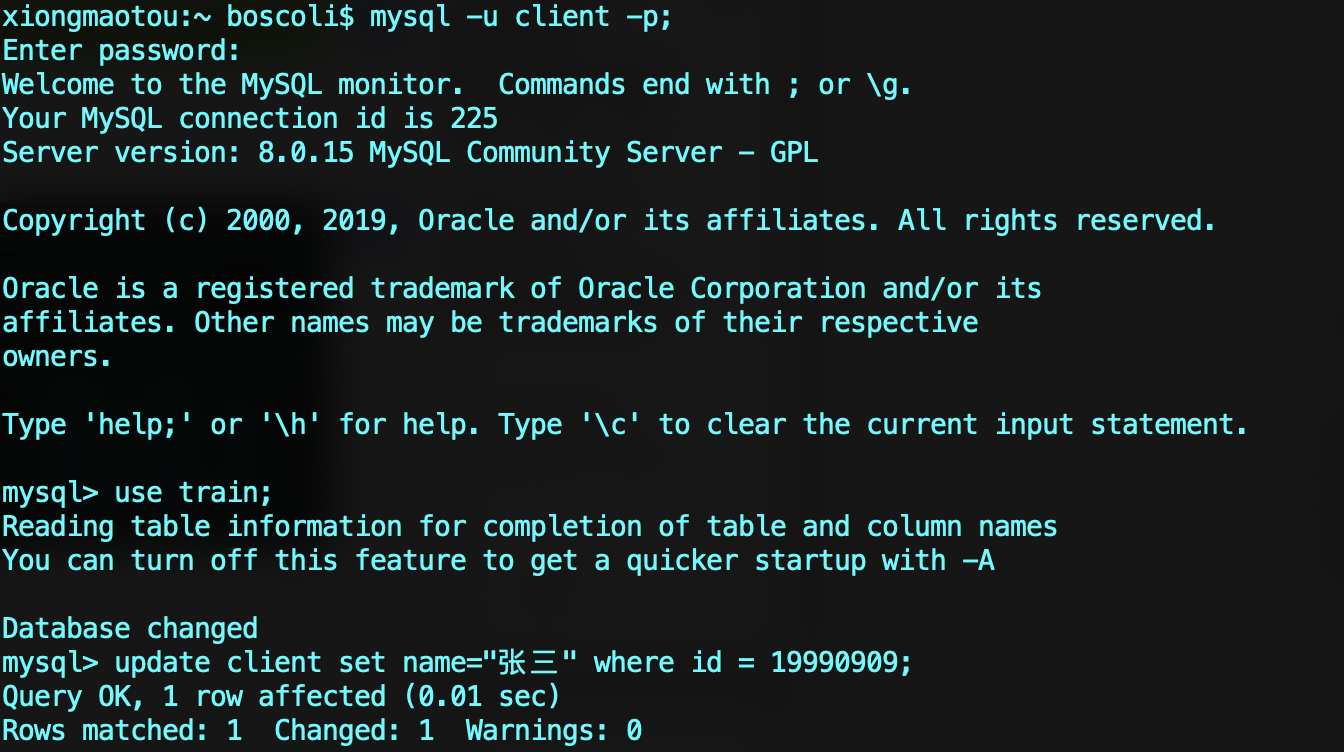
（3）检查其是否有查看表的权限，结果表面其具有此权限

（4）检查其是否有更新表的权限，结果表面其不具有此权限

（5）在root数据库创建角色并赋予在client用户上修改client表信息的权限



（6）检查client用户是否有修改client信息的权限,结果表明其具有此权限

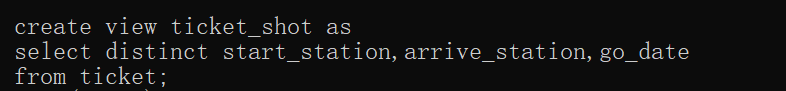


合理性论证：创建一个client用户，只分配用于select查询train数据库中表的权限，防止顾客篡改数据库中的数据

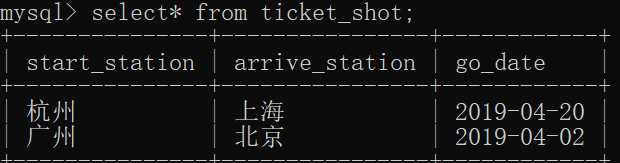
1. 设计视图, 并论证设计的合理性

此环节设计在数据库应用生命周期的需求分析和收集阶段标识的用户视图。通常，视图使用SQL或类似QBE的工具创建。例如，查询车票的站点信息

（1）创建视图



（2）检验视图的

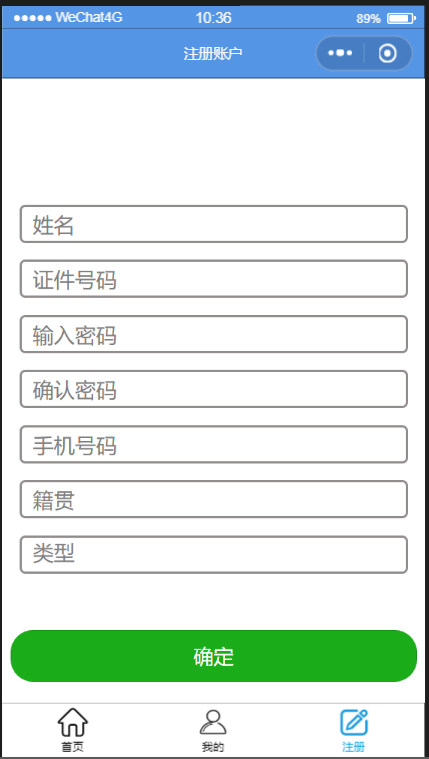


合理性论证：创建一个ticket\_shot视图，使得ticket表中的始发站和到达站的数据更加清晰而且便于查询，提高了数据的重用性和封装性。

**概要设计**

1、注册功能

通过输入姓名、证件号码、密码、手机号码、籍贯、类型来注册用户。并且确保确认密码和注册密码相同才可成功注册。



2、登录功能

通过输入证件号码和密码登录用户。并且检验证件号码是否在数据库中，密码是否和证件号码匹配。



3、查询用户信息功能

使用client表的主键查询获得用户的详细信息，并且可以通过点击退出账户按钮退出当前用户。需在用户登录之后才可以使用此功能。



4、修改个人信息功能

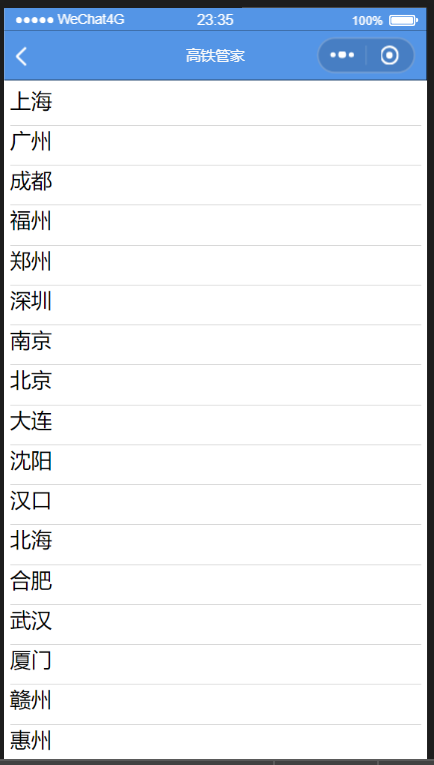
可以修改用户的任意一项信息，并点击确认按钮确认修改。需在用户登录之后才可以使用此功能。



5、查询车票功能

通过输入出发站、目的站、出发日期、是否为学生票查询车票信息。点击站点之间的交换图标可以直接交换出发站和目的站。







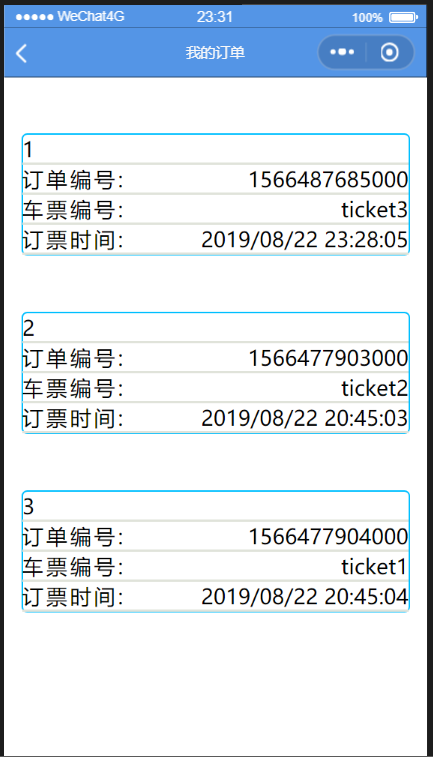
6、订票功能

通过单击车票订票，需要确保用户登录之后才可以订票，确保成人不能订学生票，确保余票为0的车票不能再被购买。



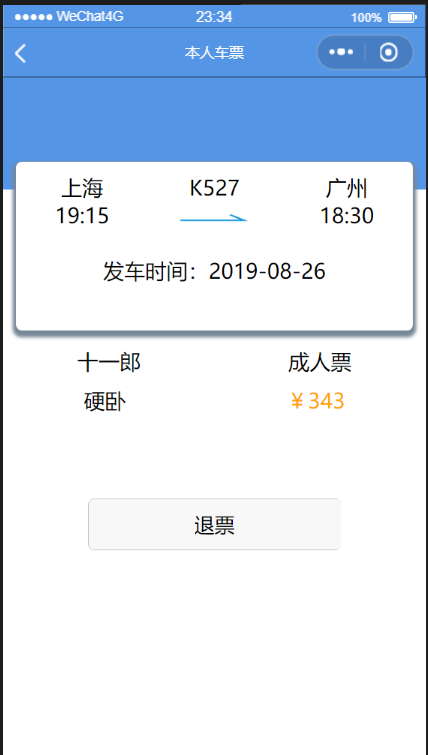
7、订单查询功能

通过orders表的client\_id属性获得当前用户的所有订单信息。需在用户登录之后才能使用此功能。



8、退票功能

通过单击退票按钮进行退票。退票后所退的票的余量自动加1。



**详细设计**

1.引言

1.1编写目的

本文档旨在为火车票销售系统提供设计说明，并且向数据库实训的评委们详细介绍团队的项目和工作。

1.2背景

a.   开发系统名称：火车票销售系统

b.  任务提出者：曾兵老师

c. 用户：华南理工大学教职工和在校学生

1.3使用技术

后端框架：SpringBoot+Mybatis

前端框架：微信小程序

数据库：MySQL

2. 系统的结构

3．前端模块设计说明

3.1模块描述

该模块显示在用户端的模块。主要负责给用户浏览，与用户进行交互。

3.2功能

具有与用户交互的窗口，可以使用户进行注册，登录，查询个人信息，修改个人信息，查询车票，订票、查看订票信息、退票等操作。

3.3性能

能够流畅运行在一般服务器上

3.4输入项

1、出发站、到达站、出发日期

2、用户id和密码

3.5输出项

1、火车票信息

2、用户信息

3.6设计方法（算法）

使用微信小程序的wx.request调用后端的接口

3.8接口

1、<http://localhost:8080/ticket/findArriveStation>

2、http://localhost:8080/client/findById

3、http://localhost:8080/client/update

4、http://localhost:8080/client/findAll

5、http://localhost:8080/orders/findByClient\_id

6、http://localhost:8080/ticket/findBy4

7、http://localhost:8080/orders/save

8、http://localhost:8080/client/save

9、http://localhost:8080/ticket/findStartStation

10、http://localhost:8080/ticket/findById

11、http://localhost:8080/orders/delete

4．数据库模块设计说明

4.1模块描述

该模块是火车票售票系统的数据库部分，负责存储用户数据

4.2功能

存储并保存用户数据

4.3性能

可以运行在一般的服务器上

4.4设计方法

使用MySQL构建数据库

5．后端模块设计说明

5.1模块描述

该模块负责与数据库和微信小程序之间的交互

5.2功能

负责前后端的数据交互，可以完成注册、登录、查票、买票等操作。

5.3性能

能够流畅运行在服务器上

5.4输入项

出发站、到达站、日期

5.5输出项

火车票

5.6设计方法

使用SpringBoot+Spring Data JPA完成后端的设计

5.7接口

localhost:8080/client/findAll 客户查询

localhost:8080/client/findById?id= 根据id查询用户

localhost:8080/client/save?id=3333&name=一 插入用户数据

localhost:8080/orders/findAll 查询订单信息

等等

**软件测试**

该系统可以完美运行在IOS系统以及Android系统之上，还可以兼容手机各个版本。在运行过程中，如果数据错误还会给予用户以详细的提示，很好的实现了人机交互。

除此之外，还对此火车票售票管理系统进行了低强度的压力测试。测试结果较为理想

**软件演示**

截图见第三部分概要设计，视频参见录制视屏见

**软件使用教程**

在系统先创建数据库train，后端代码导入IntelliJ IDEA并运行，然后前端代码导入微信开发者工具并运行。

**十、个人总结**

在这次数据库实训中，我使用的是SpringBoot+MyBatis的框架，springboot框架主要分为实体层、接口层、服务层、控制层4个层，运用了控制反转和面向切面编程等方法，使得代码和开发难度大大简化。一个好的框架足以事半功倍。在微信小程序的开发中，我使用wx.request调用后端实现的接口，在前端实现了ticket表的查询功能，实现了client表的插入数据功能、修改数据功能、查询功能，实现了orders表的插入数据功能、删除数据功能、查询功能。

小程序中有很多封装好的函数可以直接调用，降低了开发的复杂性。但是小程序中的一些数据类型，特别是时间类的数据类型与数据库中的不同，使得要在前端进行数据截取和类型转换之类的工作，比较繁琐。小程序中各个页面的样式设计以简洁为主，但使用流畅，符合我们的理念。

通过这次数据库实训，我加深了数据库的设计和使用方面的掌握程度，前端开发技能得到提升，收获很大。