

宾馆客房管理系统设计报告

数据库原理课程设计

# （ 2 0 2 3/ 2 0 2 4 学年 第二学期 ）

|  |  |
| --- | --- |
| **指 导 老 师 ：** | 王艳 |
| **姓 名 ：** | 王璨，王晨好，徐晨瑜 |
| **学 号 ：** | 222241807528，222241807529，222241807531 |
| **专 业 ：** | 计算机科学与技术 |

宾馆客房管理系统

——数据库原理课程设计

组长：王璨222241807528

组员：王晨好222241807529

徐晨瑜222241807531

分工：

王 璨： GUI后端+博客撰写

王晨好：数据库实施+实验报告撰写

徐晨瑜：GUI前端+PPT制作

目录

[（ 2 0 2 3/ 2 0 2 4 学年 第二学期 ） 1](#_Toc169276343)

[1.宾馆客房管理系统规划 1](#_Toc169276344)

[1.1研究项目现状及研究意义 1](#_Toc169276345)

[1.2研究项目可行性分析 1](#_Toc169276346)

[1.2.1技术可行性 1](#_Toc169276347)

[1.2.2经济可行性 1](#_Toc169276348)

[1.2.3操作可行性 1](#_Toc169276349)

[2.宾馆客房管理系统需求分析 2](#_Toc169276350)

[2.1需求描述 2](#_Toc169276351)

[2.2功能描述 2](#_Toc169276352)

[3.宾馆客房管理系统概念结构设计 4](#_Toc169276353)

[3.1宾馆客房管理系统的实体 4](#_Toc169276354)

[3.2局部 ER 图（子系统） 4](#_Toc169276355)

[3.3全局ER图 6](#_Toc169276356)

[4.宾馆客房管理系统逻辑结构设计 7](#_Toc169276357)

[4.1关系模型的设计 7](#_Toc169276358)

[4.2视图设计 7](#_Toc169276359)

[5.宾馆客房管理系统物理结构设计 8](#_Toc169276360)

[5.1数据库表结构的设计 8](#_Toc169276361)

[5.2数据库表之间的关系图 10](#_Toc169276362)

[5.3索引的建立 11](#_Toc169276363)

[5.4评价物理结构 12](#_Toc169276364)

[6.宾馆客房管理系统数据库系统实现 15](#_Toc169276365)

[6.1数据库的建立 15](#_Toc169276366)

[6.1.1创建数据库 15](#_Toc169276367)

[6.1.2创建表 15](#_Toc169276368)

[6.1.3视图的创建 17](#_Toc169276369)

[6.1.4索引建立 18](#_Toc169276370)

[6.2数据装载 18](#_Toc169276371)

[6.3安全性设计 20](#_Toc169276372)

[6.4完整性设计 20](#_Toc169276373)

[7.宾馆客房管理系统运行和维护 22](#_Toc169276374)

[7.1主要功能测试 22](#_Toc169276375)

[8.遇到的问题及调试整改 35](#_Toc169276376)

[9.总结与展望 35](#_Toc169276377)

# 1.宾馆客房管理系统规划

## 1.1研究项目现状及研究意义

随着信息技术的迅猛发展和互联网的普及，传统的宾馆管理方式已逐渐不能满足现代化宾馆的需求。现代宾馆行业竞争激烈，客户对服务的要求越来越高。为了提升宾馆的管理效率、服务质量以及客户的满意度，开发一套完善的宾馆客房管理系统显得尤为必要。传统的手工管理方式效率低下，数据容易丢失或损坏，查询不便，无法适应高效、快捷的服务需求。通过开发宾馆客房管理系统，可以实现对宾馆客房、客户、订单等信息的集中管理，减少人工操作，提高工作效率，提升服务响应速度，保障数据的安全性和完整性，同时降低运营成本，提高宾馆的整体运营效率。通过信息化手段，宾馆管理系统能够帮助宾馆在市场竞争中占据优势，推动宾馆行业的现代化发展。

## 1.2研究项目可行性分析

### 1.2.1技术可行性

系统设计采用了成熟的数据库管理技术，包括MySQL数据库的创建、表的建立、视图和索引的设置，以及触发器的使用，确保了系统的稳定性和高效性。团队具备丰富的数据库设计和开发经验，能够有效应对项目开发过程中可能遇到的技术挑战。

系统采用模块化设计，各功能模块相对独立且相互协作，便于后期的维护和扩展。

### 1.2.2经济可行性

项目开发成本主要包括人力成本和软硬件成本。通过详细的项目规划和进度控制，可以有效控制开发成本。系统的投入使用可以显著提升宾馆的管理效率，减少人力成本，增加客户满意度，从而提高宾馆的综合竞争力，具备良好的投资回报前景。

### 1.2.3操作可行性

系统设计界面友好，操作简单，提供详细的用户手册和培训，确保员工能够快速上手使用系统。系统设计了完善的数据备份和恢复功能，提供灵活的报表导出和分析功能，便于日常的系统维护和数据管理。

# 2.宾馆客房管理系统需求分析

## 2.1需求描述

该宾馆管理系统旨在为宾馆提供包括员工管理、客房管理、客户管理、订单管理、报表维护等方面的一体化解决方案。同时，还提供用户登录和密码管理等功能。其业务需求涵盖了从客户预订、入住到退房的整个流程管理，并提供详细的报表支持，以便于宾馆的日常运营和决策。

员工管理：管理员可以添加新员工的信息，如姓名、职位、联系方式等；删除现有员工的信息，确保信息的准确性和及时性；更新员工的详细信息，维护员工档案的完整性；支持通过姓名、职位等多种方式查询员工信息，方便管理。

客房管理：管理客户的预约信息。录入客户的预订信息，生成预订单；处理客户的预约取消请求，更新预订状态。管理宾馆的房间信息，包括添加新房间和修改现有房间信息，录入新房间的详细信息，如房间号、房型、价格等，更新现有房间的信息，保持房间资料的准确性。

客户管理：提供多种方式查询客户信息，方便服务和管理；录入新客户的信息，建立客户档案。

订单管理：查询客户的预订信息，跟踪预订状态；查询客户的订单信息

报表和维护：定期备份系统数据库，确保数据安全；导出数据库数据，进行本地存储或迁移；导出特定数据表，便于分析和审查。生成和展示各种管理报表，提供数据支持和决策依据；导出报表数据表格，便于记录和分享。

登录功能：用户通过用户名和密码登录系统，确保系统安全；提供找回密码功能，帮助用户重新获取访问权限；用户可以修改个人密码，确保账号的安全性。

## 2.2功能描述

**员工管理功能**

添加员工：录入员工的详细信息，如姓名、职位、联系方式，建立员工档案。

删除员工：删除员工信息，确保信息的准确性和及时性。

修改员工：更新员工的详细信息，维护员工档案的完整性。

查询员工信息：通过姓名、职位等多种方式查询员工信息，方便管理和调度。

**客房管理功能**

预约功能

预约：录入并管理客户的预约信息，生成预订单。

取消预约：处理预约取消请求，更新预订状态。

入住功能

个人入住：记录客户入住信息，分配房间。

团体入住：处理团体客户入住，分配多个房间。

房间管理

添加房间：录入新房间信息，如房间号、房型等，更新系统房间数据。

修改房间：更新现有房间信息，保持数据准确性。

**客户管理功能**

查询客户：通过多种方式查询客户信息，方便服务和管理。

添加客户：录入新客户信息，建立客户档案。

订单管理功能

查询预订：查询并跟踪客户预订信息。

查询订单：查询客户的订单信息，管理订单状态。

**报表和维护功能**

数据库备份：定期备份数据库，确保数据安全。

导出数据库：导出数据库数据，便于存储和迁移。

导出数据表：导出特定数据表，方便分析和审查。

报表展示

生成报表：生成各类管理报表，支持决策。

导出报表：导出报表数据，便于记录和分享。

**登录功能**

登录：用户通过用户名和密码登录系统。

找回密码：提供找回密码功能，确保用户可以重新获取登录权限。

修改密码：用户可以修改个人密码，确保账号安全。

# 3.宾馆客房管理系统概念结构设计

## 3.1宾馆客房管理系统的实体

员工staff：{员工编号sid，员工姓名sname，员工性别ssex，登录账号susername，登录密码spassword，权限等级srole }

个人客户client：{客户姓名cname，客户编号cid，客户电话cphone，客户年龄cage，客户性别csex，登记员工编号register\_sid,登记时间register\_time}

客房room：{客房编号rid，客房类型rtype，客房人数rsize，客房价格rprice，客房状态rstatus }

订单hotelorder：{订单编号id，订单类型ordertype，开始时间start\_time，结束时间end\_time，客房编号rid，订单金额money，订单时间order\_time,登记员工编号register\_sid}

团队客户：{团队名称tname，团队编号tid，团体电话tphone, 登记员工编号check\_in\_sid,登记时间register\_time }

## 3.2局部 ER 图（子系统）

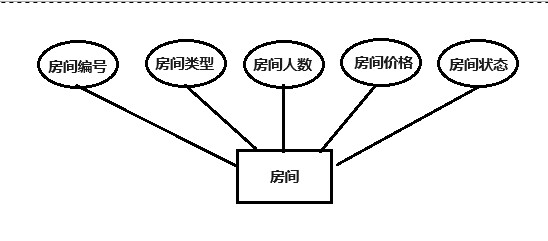
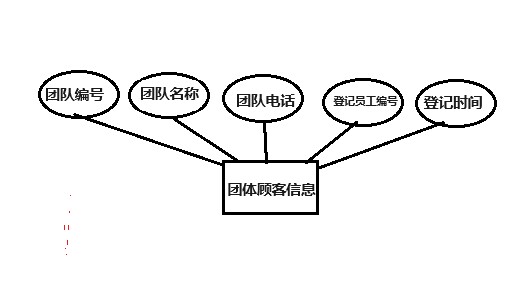


图 1 局部 ER 图 1



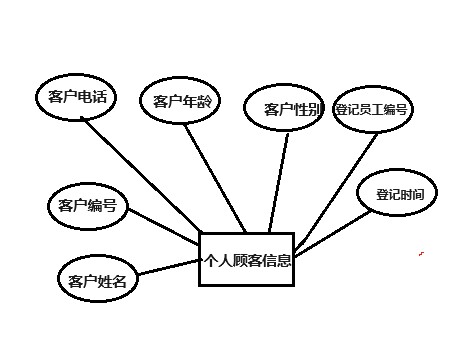
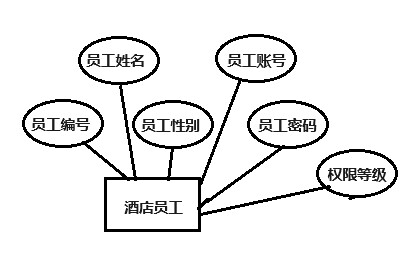
图 2 局部 ER 图 2

图 3 局部 ER 图 3

图 4 局部 ER 图 4

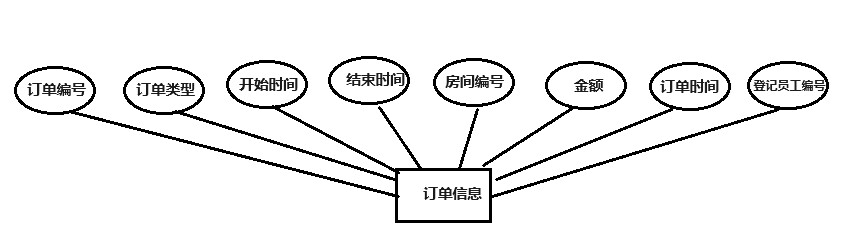


图 5 局部 ER 图 5

## 3.3全局ER图

图 6 全局 ER 图 1

# 4.宾馆客房管理系统逻辑结构设计

## 4.1关系模型的设计

团体客户和客房的入驻联系是1：n，可以将它们之间的联系与n端实体团体客户合并；客户与订单的预订关系也是1：n，同样可以将其中的联系与n端实体合并，而员工与客房之间的管理和员工与订单的登记联系方式是n：m，将其联系转化称独立的关系模式。具体的基本E-R图向关系模式的转化如下所述：

客房（客房编号，客房类型，客房容量，客房价格，客房状态）

员工（员工编号，员工姓名，员工性别，登录账号，登录密码，权限等级）

个人客户（客户编号，客户姓名，客户电话，客户年龄，客户性别，登记时间，登记员工编号）

团队客户（团队编号，团队名称，团体电话，登记时间，登记员工编号）

预订（预约编号，类型,客户编号,客房编号,入住时间，退房时间，预订时间, 登记员工编号）

前台登记（登记编号,类型,客户编号,客房编号,入住时间，退房时间，总价,登记员工编号）

订单（订单编号，订单类型，开始时间，结束时间，房间编号,订单金额，订单时间,登记员工编号）

## 4.2视图设计

外模式（用户模式）的设计，为了向客户提供友好的用户界面，需要设计一些视图。

客户基本信息视图Customers视图

房间基本信息视图Rooms视图

入住订单信息视图Living视图

管理员信息视图Administrators视图

# 5.宾馆客房管理系统物理结构设计

## 5.1数据库表结构的设计

物理结构设计的目的是为一个给定的逻辑数据模型选取一个最适合应用环境的物理结构。宾馆客房预订管理系统数据库中各个表结构设计结果见表。每个表表示在数据库中的一张表。

表 5.1 房间信息表room

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| rid | varchar(255) | Not null (主键) | 房间编号 |
| rtype | varchar(255) | Not null | 房间类型 |
| rsize | SMALLINT | Not null | 房间人数 |
| rprice | varchar(255) | Not null | 房间价格 |
| rstatus | varchar(255) | Not null | 房间状态 |

表 5.2 宾馆员工的个人信息表staff

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| sid | varchar(255) | Not null (主键) | 员工编号 |
| sname | varchar(255) | Not null | 员工姓名 |
| ssex | varchar(255) | 可为空 | 员工性别 |
| susername | varchar(255) | Not null | 员工账号 |
| spassword | varchar(255) | Not null | 员工密码 |
| srole | varchar(255) | Not null | 权限等级 |

表 5.3 个人顾客信息表client

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| cname | varchar(255) | Not null | 客户姓名 |
| cid | varchar(255) | Not null (主键) | 客户编号 |
| cphone | varchar(255) | 可为空 | 客户电话 |
| cage | varchar(255) | Not null | 客户年龄 |
| csex | varchar(255) | 可为空 | 客户性别 |
| register\_sid | varchar(255) | 可为空 | 登记员工编号 |
| register\_time | timestamp | 可为空 | 登记时间 |

表 5.4 团体顾客信息表team

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| tname | varchar(255) | Not null | 团队名称 |
| tid | varchar(255) | Not null (主键) | 团队编号 |
| tphone | varchar(255) | 可为空 | 团队电话 |
| check\_in\_sid | varchar(255) | 可为空 | 登记员工编号 |
| register\_time | timestamp | 可为空 | 登记时间 |

表 5.5 预约记录表booking表

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| booking\_id | int | Not null(主键) | 预约编号 |
| type | enum('individual','team') | Not null | 预约类型 |
| client\_or\_team\_id | varchar(255) | Not null | 客户编号 |
| room\_id | varchar(255) | Not null | 房间编号 |
| start\_time | date | 可为空 | 入住时间 |
| end\_time | date | 可为空 | 退房时间 |
| booking\_time | timestamp | 可为空 | 预订时间 |
| register\_sid | varchar(255) | 可为空 | 登记员工编号 |

表 5.6 前台登记记录表checkin

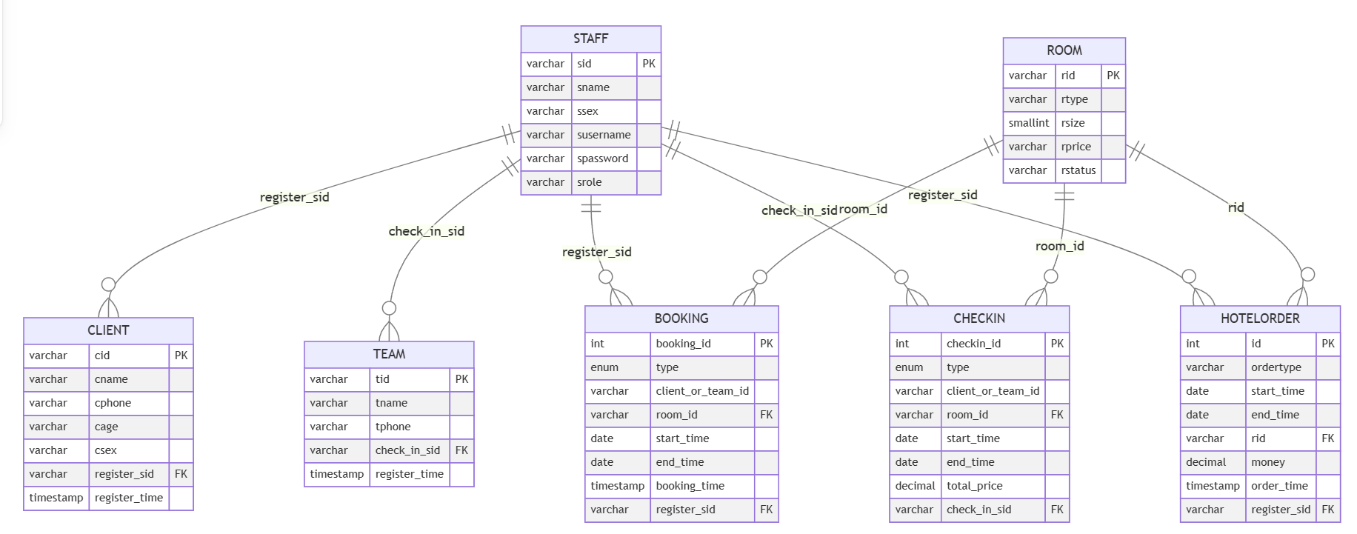
| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| checkin\_id | int | Not null(主键) | 登记编号 |
| type | enum('individual','team') | Not null | 登记类型 |
| client\_or\_team\_id | varchar(255) | Not null | 客户编号 |
| room\_id | varchar(255) | Not null | 房间编号 |
| start\_time | date | 可为空 | 入住时间 |
| end\_time | date | 可为空 | 退房时间 |
| total\_price | decimal(10,2) | 可为空 | 总价 |
| register\_sid | varchar(255) | 可为空 | 登记员工编号 |

表 5.7宾馆所有订单信息和收入情况表hotelorder

| **表中列名** | **数据类型** | **可否为空** | **说明** |
| --- | --- | --- | --- |
| id |  | Not null (主键) | 订单编号 |
| ordertype | varchar(255) | Not null | 订单类型 |
| start\_time | date | Not null | 订单开始时间 |
| end\_time | date | Not null | 订单结束时间 |
| rid | varchar(255) | Not null | 房间编号 |
| money | decimal(10,2) | 可为空 | 金额 |
| order\_time | timestamp | 可为空 | 订单时间 |
| register\_sid | varchar(255) | 可为空 | 登记员工编号 |

## 5.2数据库表之间的关系图

以上数据库表与表之间的关系如图7所示（下图仅为示例图）

图 7 表间关系图1

## 5.3索引的建立

物理结构设计的任务之一是根据数据库管理系统支持的存取方式确定选取，常见的存取方式为索引。

索引设计

--客户电话号码索引：加快通过电话号码查询客户信息的速度

CREATE INDEX idx\_cphone ON client (cphone);

--房间价格索索引：加快通过价格查询房间的速度

CREATE INDEX idx\_rprice ON room (rprice);

--为状态字段添加索引可能会有助于快速筛选

CREATE INDEX idx\_rstatus ON room (rstatus);

--根据入住日期、预订日期或注册时间来查询数据

CREATE INDEX idx\_booking\_start\_time ON booking (start\_time);

CREATE INDEX idx\_booking\_end\_time ON booking (end\_time);

CREATE INDEX idx\_client\_register\_time ON client (register\_time);

CREATE INDEX idx\_checkin\_start\_time ON checkin (start\_time);

CREATE INDEX idx\_checkin\_end\_time ON checkin (end\_time);

## 5.4评价物理结构

数据库的物理设计，现用相关截图表示如下：

图 8 数据库物理结构设计图1



图 9 数据库物理结构设计图2



图10 数据库物理结构设计图3

图 11 数据库物理结构设计图4

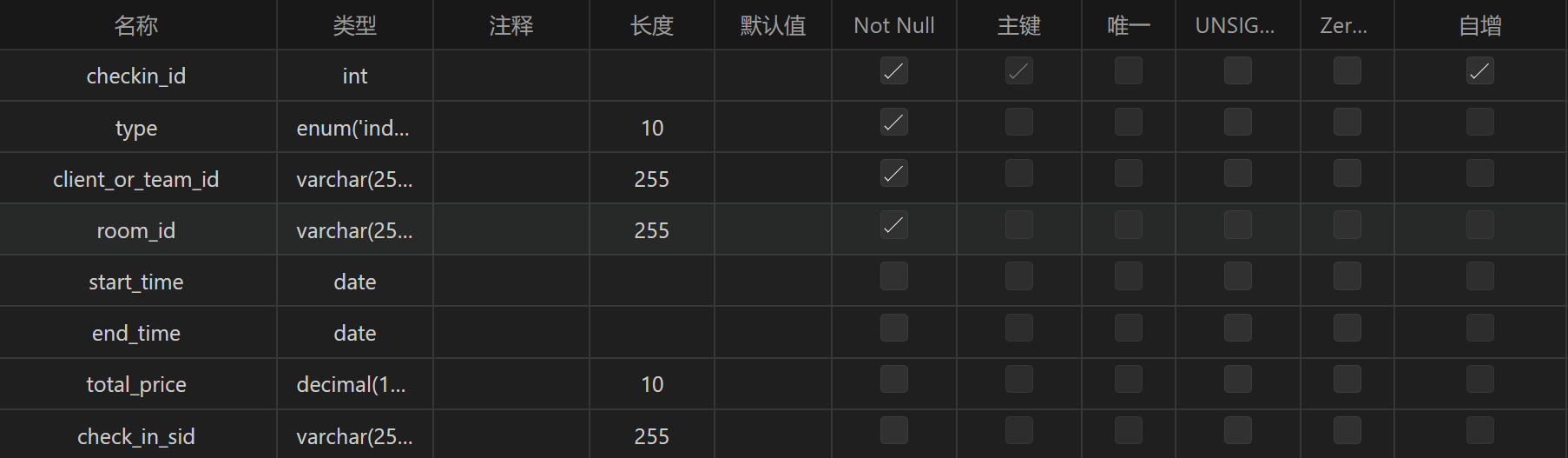
图 12 数据库物理结构设计图5

图 13 数据库物理结构设计图6

**物理结构设计评价**

**表结构设计**：

各表的字段设计合理，涵盖了业务需求的所有关键属性，如员工、客户、房间、订单等信息。字段类型的选择也符合数据存储需求，如varchar、smallint、timestamp等。

所有表都设计了主键，确保了记录的唯一性。同时，使用外键来建立表与表之间的关系，保证了数据的完整性和一致性。

**索引设计**：

在常用查询字段上建立索引，如客户电话、房间价格、状态等，能够显著提高查询效率，减少查询时间。

对于涉及多个条件查询的情况，设计了组合索引，如房间类型和价格的组合索引，能够进一步优化复杂查询的性能。

**视图设计**：

通过视图来简化常用的查询操作，例如客户信息视图、房间信息视图、入住信息视图等，减少了直接查询表的复杂性，提升了开发和运维的效率。

视图设计还可以在一定程度上提高数据安全性，通过视图可以限制用户对底层数据表的直接访问。

**安全性设计**：

设计了不同级别的员工角色，并为不同角色分配了相应的权限，确保系统的安全性。例如，1级员工只能访问部分数据，2级员工则可以访问所有数据。

提供了密码修改和找回功能，确保用户账户的安全性，防止未经授权的访问。

**触发器设计**：

通过触发器实现了一些自动化操作，例如在客户入住时自动更新房间状态，在订单插入时自动记录订单信息。这些触发器能够有效减少人工干预，提高系统的自动化水平。

触发器还确保了数据的一致性，例如在客户退房时自动更新房间状态为“空闲”，防止数据出现错误。

总结

总体来看，宾馆客房管理系统的物理结构设计较为完善，考虑了数据存储的合理性、查询性能的优化以及数据安全性的保障。通过合理的表设计、索引优化、视图简化、权限管理和触发器自动化等手段，该系统能够高效地支持宾馆的日常运营和管理需

# 6.宾馆客房管理系统数据库系统实现

## 6.1数据库的建立

### 6.1.1创建数据库

|  |
| --- |
| 建库 |
| CREATE DATABASE `HotelSystem` DEFAULT CHARACTER SET = 'utf8mb4';  SHOW DATABASES;  USE HotelSystemplus; |

### 6.1.2创建表

|  |
| --- |
| 建表 |
| CREATE TABLE `room` (  `rid` VARCHAR(255) NOT NULL,  `rtype` VARCHAR(255) NOT NULL,  `rsize` SMALLINT NOT NULL,  `rprice` VARCHAR(255) NOT NULL,  `rstatus` VARCHAR(255) NOT NULL,  PRIMARY KEY (`rid`),  KEY `rprice` (`rprice`),  KEY `rtype` (`rtype`, `rprice`)  );  CREATE TABLE `staff` (  `sid` varchar(255) NOT NULL,  `sname` varchar(255) NOT NULL,  `ssex` varchar(255) DEFAULT NULL,  `susername` varchar(255) NOT NULL,  `spassword` varchar(255) NOT NULL,  `srole` varchar(255) NOT NULL,  PRIMARY KEY (`sid`),  UNIQUE KEY `susername` (`susername`)  );  CREATE TABLE `client` (  `cname` varchar(255) NOT NULL,  `cid` varchar(255) NOT NULL,  `cphone` varchar(255) DEFAULT NULL,  `cage` varchar(255) NOT NULL,  `csex` varchar(255) DEFAULT NULL,  `register\_sid` varchar(255) DEFAULT NULL,  `register\_time` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  PRIMARY KEY (`cid`),  KEY `sid` (`register\_sid`),  KEY `cid` (`cid`, `register\_sid`),  CONSTRAINT `sid` FOREIGN KEY (`register\_sid`) REFERENCES `staff` (`sid`)  );  CREATE TABLE `team` (  `tname` varchar(255) NOT NULL,  `tid` varchar(255) NOT NULL,  `tphone` varchar(255) DEFAULT NULL,  `check\_in\_sid` varchar(255) DEFAULT NULL,  `register\_time` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  PRIMARY KEY (`tid`),  KEY `team\_sid` (`check\_in\_sid`),  CONSTRAINT `team\_sid` FOREIGN KEY (`check\_in\_sid`) REFERENCES `staff` (`sid`)  );  CREATE TABLE `booking` (  `booking\_id` INT AUTO\_INCREMENT PRIMARY KEY,  `type` ENUM('individual', 'team') NOT NULL,  `client\_or\_team\_id` VARCHAR(255) NOT NULL,  `room\_id` VARCHAR(255) NOT NULL,  `start\_time` DATE DEFAULT NULL,  `end\_time` DATE DEFAULT NULL,  `booking\_time` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  `register\_sid` VARCHAR(255) DEFAULT NULL,  FOREIGN KEY (`room\_id`) REFERENCES `room` (`rid`),  FOREIGN KEY (`register\_sid`) REFERENCES `staff` (`sid`)  );  CREATE TABLE `checkin` (  `checkin\_id` INT AUTO\_INCREMENT PRIMARY KEY,  `type` ENUM('individual', 'team') NOT NULL,  `client\_or\_team\_id` VARCHAR(255) NOT NULL,  `room\_id` VARCHAR(255) NOT NULL,  `start\_time` DATE DEFAULT NULL,  `end\_time` DATE DEFAULT NULL,  `total\_price` DECIMAL(10, 2) DEFAULT NULL,  `check\_in\_sid` VARCHAR(255) DEFAULT NULL,  FOREIGN KEY (`room\_id`) REFERENCES `room` (`rid`),  FOREIGN KEY (`check\_in\_sid`) REFERENCES `staff` (`sid`)  );  CREATE TABLE IF NOT EXISTS `hotelorder` (  `id` INT AUTO\_INCREMENT PRIMARY KEY,  `ordertype` VARCHAR(255) NOT NULL,  `start\_time` DATE NOT NULL,  `end\_time` DATE NOT NULL,  `rid` VARCHAR(255) NOT NULL,  `money` DECIMAL(10, 2) DEFAULT NULL,  `order\_time` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  `register\_sid` VARCHAR(255) DEFAULT NULL,  FOREIGN KEY (`rid`) REFERENCES `room` (`rid`),  FOREIGN KEY (`register\_sid`) REFERENCES `staff` (`sid`)  ); |

### 6.1.3视图的创建

|  |  |  |
| --- | --- | --- |
|  | 视图 |  |
| -- 创建Customers视图：包含客户的ID、姓名、性别和电话信息  CREATE VIEW Customers AS SELECT Cid, Cname, Csex, Cphone FROM Client;  -- 创建Rooms视图：包含房间的ID、类型、大小、价格和状态  CREATE VIEW Rooms AS  SELECT Rid, Rtype, Rsize, Rprice, rstatus  FROM Room;  -- 创建Living视图：包含入住的房间ID、客户ID、开始时间、结束时间和总价格  CREATE VIEW Living AS  SELECT  checkin\_id,  client\_or\_team\_id,  start\_time,  end\_time,  total\_price  FROM checkin;  -- 创建Administrators视图：包含管理员的ID、姓名和用户名，角色大于1的员工  CREATE VIEW Administrators AS  SELECT Sid, Sname, Susername  FROM Staff  WHERE{  Srole > 1  } | | |

### 6.1.4索引建立

|  |
| --- |
| 主键索引 |
| --客户电话号码索引：加快通过电话号码查询客户信息的速度  CREATE INDEX idx\_cphone ON client (cphone);  --房间价格索索引：加快通过价格查询房间的速度  CREATE INDEX idx\_rprice ON room (rprice);  --为状态字段添加索引可能会有助于快速筛选  CREATE INDEX idx\_rstatus ON room (rstatus);  --根据入住日期、预订日期或注册时间来查询数据  CREATE INDEX idx\_booking\_start\_time ON booking (start\_time);  CREATE INDEX idx\_booking\_end\_time ON booking (end\_time);  CREATE INDEX idx\_client\_register\_time ON client (register\_time);  CREATE INDEX idx\_checkin\_start\_time ON checkin (start\_time);  CREATE INDEX idx\_checkin\_end\_time ON checkin (end\_time); |

## 6.2数据装载

|  |
| --- |
| 数据装载 |
| Room表装载  INSERT INTO  `room` (  `rid`,  `rtype`,  `rsize`,  `rprice`,  `rstatus`  )  VALUES ('201',  '标准间（单人）',  1,  '208',  '空闲'  )…….  Staff表装载  INSERT INTO  `staff` (  `sid`,  `sname`,  `ssex`,  `susername`,  `spassword`,  `srole`  )  VALUES (  '1',  '冉冉',  '男',  'ranxi',  '123456',  '2'  )…….  Client表装载  INSERT INTO  `client` (  `cname`,  `cid`,  `cphone`,  `cage`,  `csex`,  `register\_sid`,  `register\_time`  )  VALUES (  '吴超梦',  '130898199212233434',  '13898322223',  '28',  '女',  '1',  '2024-05-20 10:48:42'  )………  Team表装载  INSERT INTO  `team` (  `tname`,  `tid`,  `tphone`,  `check\_in\_sid`,  `register\_time`  )  VALUES (  '团队1',  '1',  '13896534534',  '1',  '2020-01-06 00:50:46'  )……  Booking表装载  INSERT INTO  `booking` (  `type`,  `client\_or\_team\_id`,  `room\_id`,  `start\_time`,  `end\_time`,  `booking\_time`,  `register\_sid`  )  VALUES (  'individual',  '130898199212233434',  '201',  '2024-06-01',  '2024-06-05',  '2024-05-29 10:15:00',  '1'  )……  Checkin表装载  INSERT INTO  `checkin` (  `type`,  `client\_or\_team\_id`,  `room\_id`,  `start\_time`,  `end\_time`,  `total\_price`,  `check\_in\_sid`  )  VALUES  (  'individual',  '130898199212233434',  '201',  '2024-06-06',  '2024-06-10',  832,  '1'  )…… |

## 6.3安全性设计

-- 创建1级权限员工角色

CREATE ROLE Level1\_Staff;

-- 创建2级权限员工角色

CREATE ROLE Level2\_Staff;

--授予员工权限

GRANT Level1\_Staff TO 'zs123' @'%';

GRANT Level2\_Staff TO 'ranxi' @'%';

1级员工授予部分权限

2级员工授予全部权限

切记激活角色权限,不然授予用户后无法使用权限

SET global activate\_all\_roles\_on\_login = ON;

## 6.4完整性设计

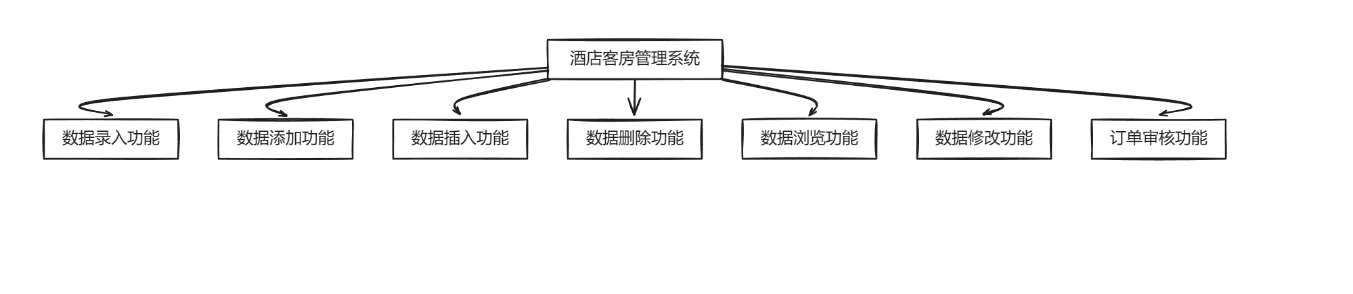
**触发器建立**

|  |
| --- |
| 触发器建立 |
| -- 创建入住以及退房时更新房间状态的触发器  DELIMITER $$  CREATE TRIGGER UpdateRoomStatusOnCheckIn  AFTER INSERT ON checkin  FOR EACH ROW  BEGIN  UPDATE room  SET rstatus = '占用'  WHERE rid = NEW.room\_id;  END$$  CREATE TRIGGER UpdateRoomStatusOnCheckOut  AFTER UPDATE ON checkin  FOR EACH ROW  BEGIN  IF NEW.end\_time < CURRENT\_DATE THEN  UPDATE room  SET rstatus = '空闲'  WHERE rid = NEW.room\_id;  END IF;  END$$  DELIMITER;  --预约订单和登记订单自动插入到全部订单表的触发器  DELIMITER $$  CREATE TRIGGER after\_insert\_checkin  AFTER INSERT ON checkin  FOR EACH ROW  BEGIN  -- 更新房间状态为占用  UPDATE room  SET rstatus = '占用'  WHERE rid = NEW.room\_id;  -- 插入订单到 hotelorder 表  INSERT INTO hotelorder (ordertype, start\_time, end\_time, rid, money, order\_time, register\_sid)  VALUES ('前台登记', NEW.start\_time, NEW.end\_time, NEW.room\_id, NEW.total\_price, NOW(), NEW.check\_in\_sid);  END$$  CREATE TRIGGER after\_update\_checkin  AFTER UPDATE ON checkin  FOR EACH ROW  BEGIN  IF NEW.end\_time < CURRENT\_DATE THEN  UPDATE room  SET rstatus = '空闲'  WHERE rid = NEW.room\_id;  END IF;  END$$  DELIMITER; |

# 7.宾馆客房管理系统运行和维护

## 7.1主要功能测试

需要实现出需求分析中的所有功能模块，主要功能如图所示。

图14 功能模块图

该系统运用 mysql 和qt共同设计而成，该系统效率较高，同时 mysql 在创建索引后一定程度上也提高了我们系统的效率，mysql 和 qt实现了以下主要功能。

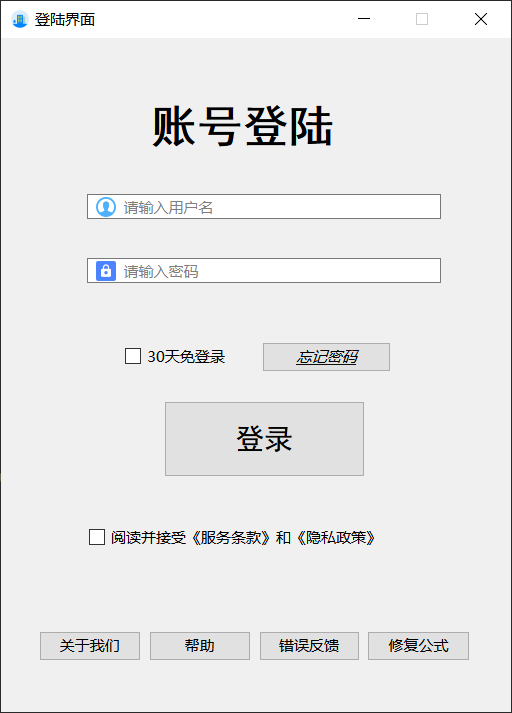
**登录界面：**

图15 测试截图 1（管理员登录界面）



图16 测试截图 2（主页）

**客房管理功能界面：**



图 17 测试截图 3（客房管理界面）

图 18 测试截图 4（客房查询功能界面）

图 19 测试截图 5（房间号查询功能界面）

图 20 测试截图 6（客房信息删除功能界面）

图21 测试截图 7（添加客房功能界面）

**客户管理功能界面：**

图 22 测试截图 8（客户管理功能界面）

图 23 测试截图 9（客户查询功能界面）

图 24 测试截图 10（添加客户功能界面）

图 25 测试截图 11（添加客户成功界面）

图 26 测试截图 12（增加团队功能界面）

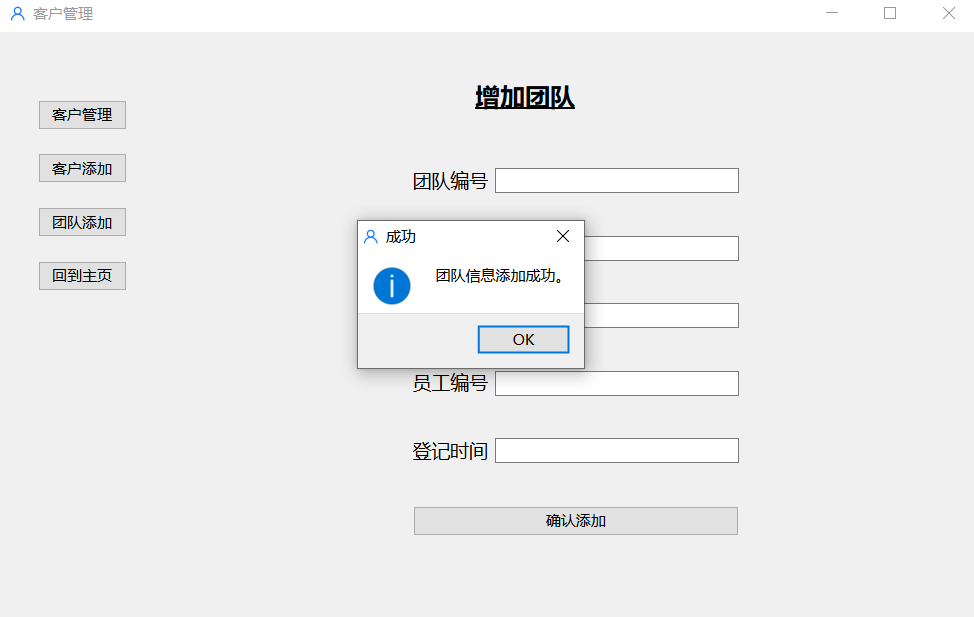


图 27 测试截图 13（增加团队功能界面）

**员工管理功能界面：**

图 28 测试截图 14（查询员工功能界面）

图 29 测试截图 15（增加员工功能界面）

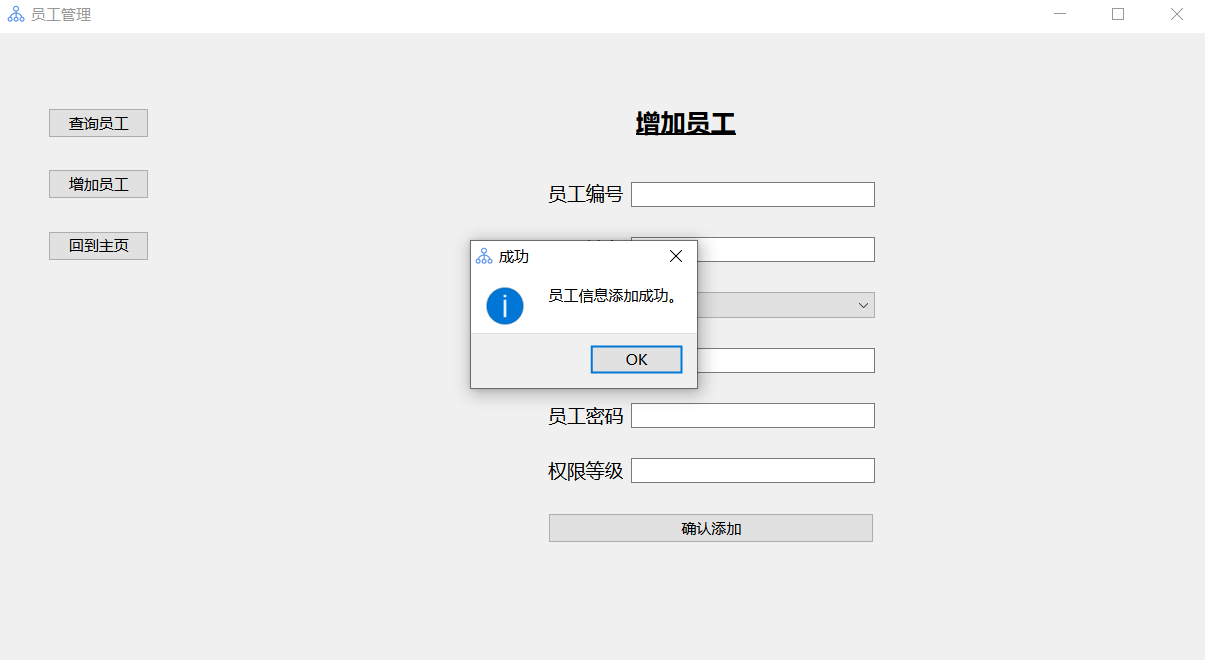
图 30 测试截图 16（增加员工功能界面）

图 31 测试截图 17（删除员工功能界面）

**订单管理功能界面：**

图 32 测试截图 18（删除员工功能界面）

图 33 测试截图 19（订单管理功能界面）

图 34 测试截图 20（订单查询功能界面）

**预约/登记功能界面：**

图 35 测试截图 21（预约管理功能界面）

图 36 测试截图 22（登记管理功能界面）

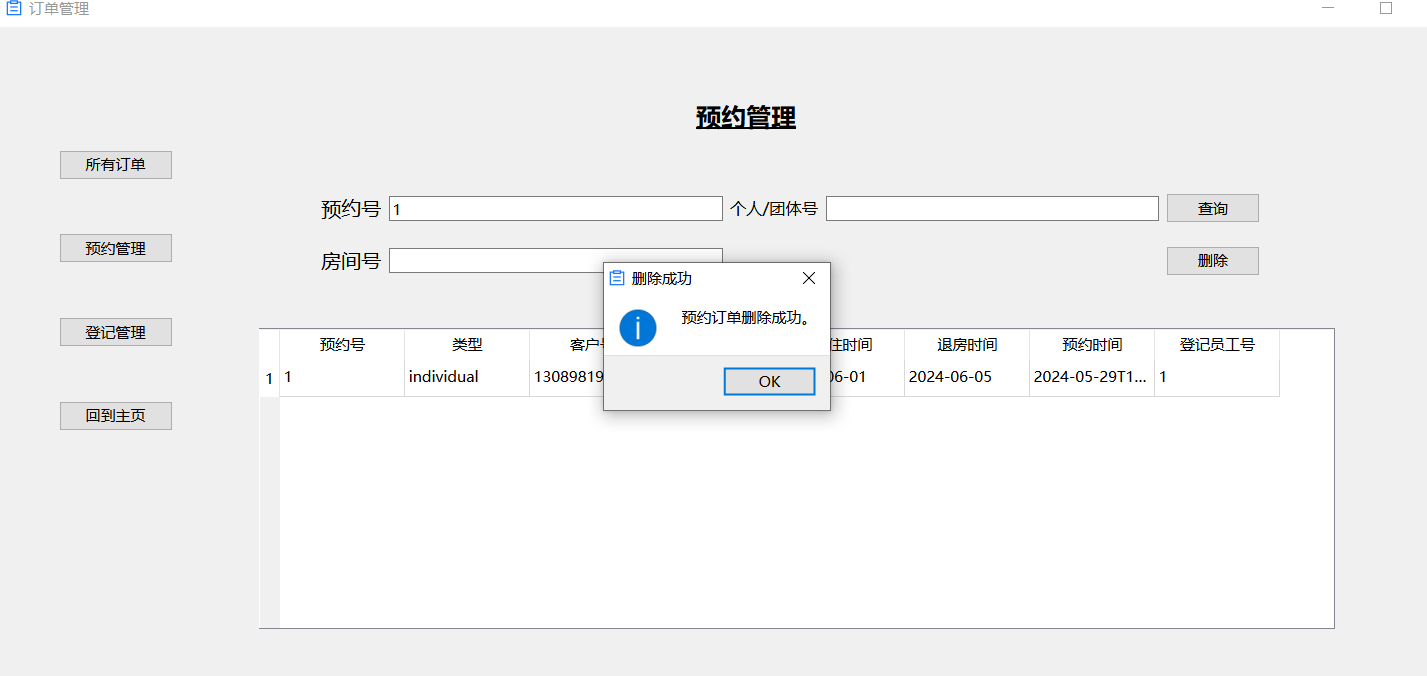
图 37 测试截图 23（预约删除功能界面）

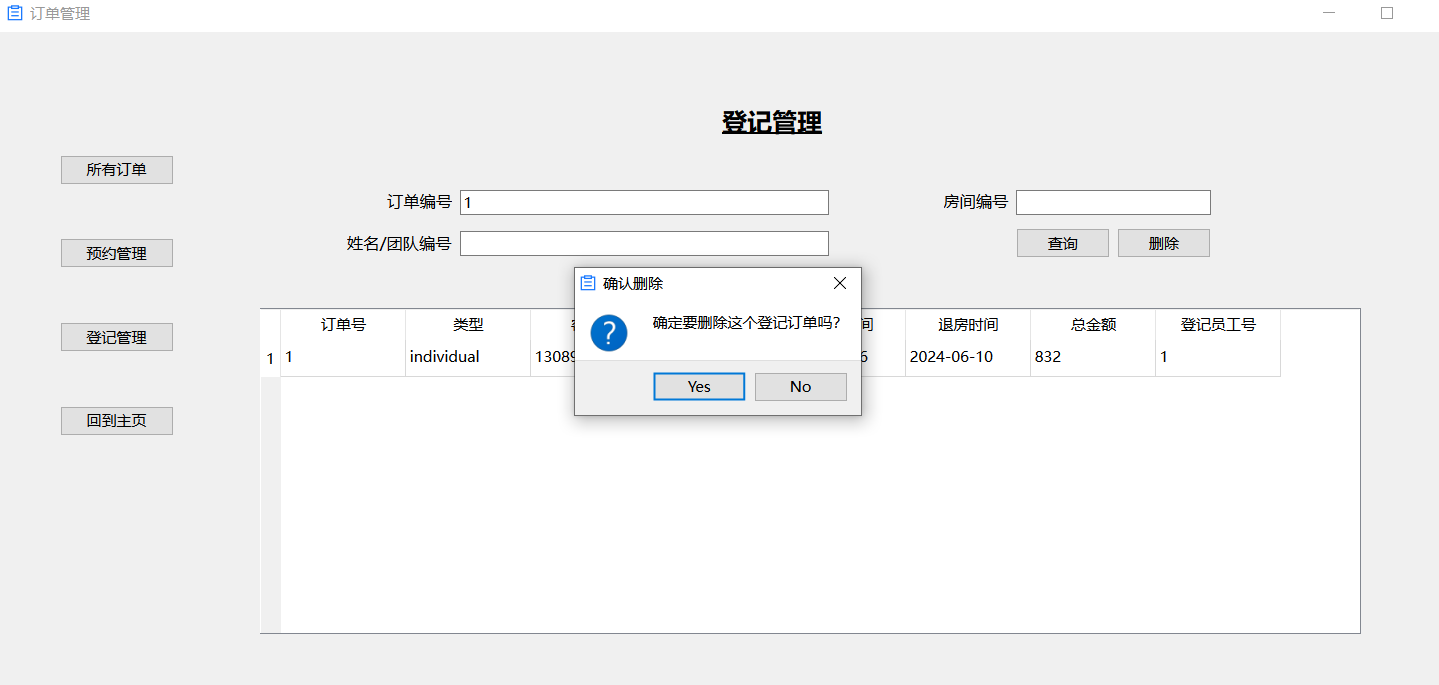
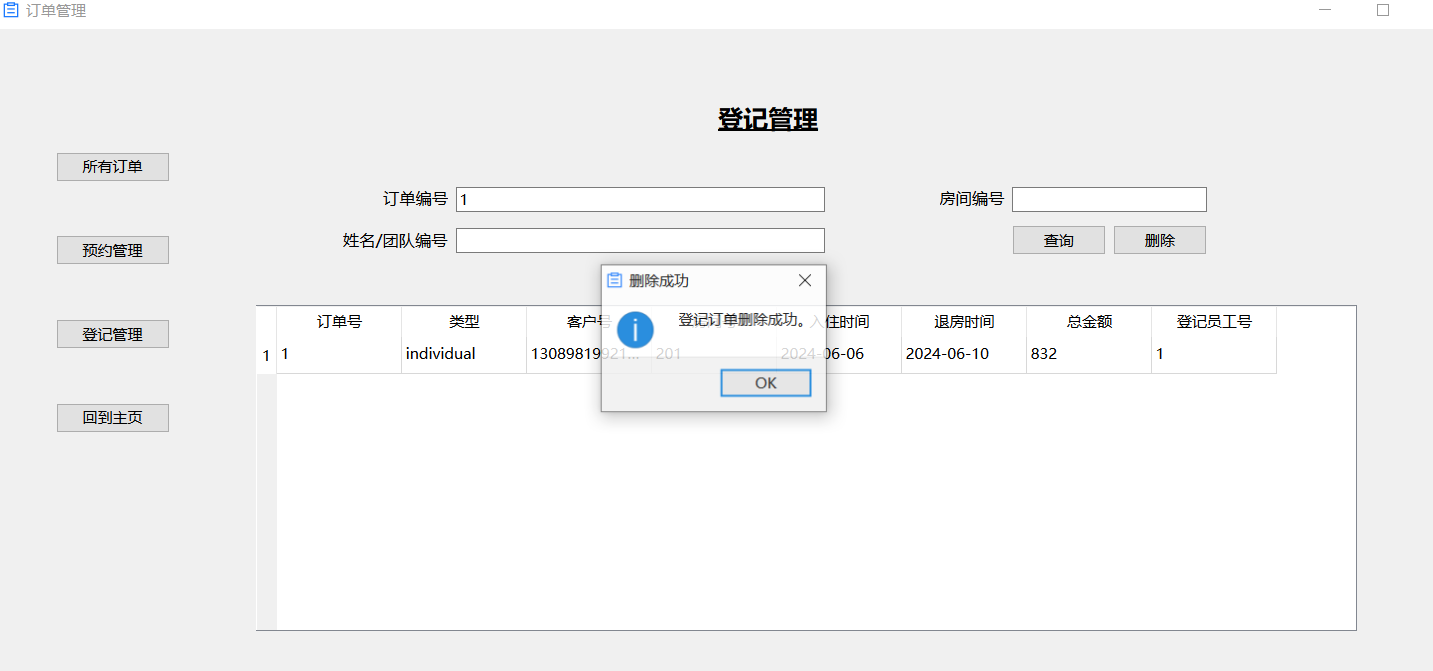
图 38 测试截图 24（登记删除功能界面）

图 39 测试截图 25（登记删除功能界面）

# 8.遇到的问题及调试整改

1.创建完角色并将权限赋予给用户后,用户无法使用?

因为角色权限默认关闭,需要激活

--激活所有角色权限

SET global activate\_all\_roles\_on\_login = ON;

# 9.总结与展望

以下是宾馆客房管理系统设计报告的总结与展望部分，以及该系统的不足之处：

总结：

通过本次宾馆客房管理系统的设计与实现，我们开发了一套能够提升宾馆管理效率和服务质量的管理系统。系统集成了员工管理、客房管理、客户管理、订单管理、报表生成等多种功能模块，极大地提高了宾馆的整体运营效率。

在技术实现上，我们采用了MySQL数据库管理系统，使用了视图、索引、触发器等多种数据库优化手段，保证了系统的高效性和稳定性。通过模块化设计，各功能模块相对独立，便于后期维护和扩展。系统在操作上界面友好，使用简单，能够快速被宾馆员工掌握和使用。

展望：

未来，我们计划进一步优化系统性能，增加更多实用功能，如在线预订系统、移动端应用等，提升系统的便捷性和用户体验。此外，我们还将继续完善数据分析和报表功能，帮助宾馆管理层做出更准确的经营决策。

在技术上，我们将探索更多先进的数据库管理和优化技术，提升系统的处理能力和响应速度。同时，我们也将加强系统的安全性设计，保护客户隐私和数据安全。

不足之处：

1. 功能覆盖面不够全面：目前系统功能相对基础，未能完全覆盖所有可能的宾馆管理需求。例如，缺乏在线预订功能和移动端支持。

2. 用户体验需提升：界面设计和用户交互细节方面还有提升空间，可以通过用户反馈进一步优化，使系统更符合用户习惯和需求。

3. 安全性有待完善：虽然设计了不同级别的员工权限，但在实际使用中可能需要更细化的权限管理和更严格的安全措施，防止数据泄露和非法访问。

通过总结与反思，我们将不断改进系统功能和性能，为宾馆提供更加高效和安全的管理工具。