

厦门大学计算机科学系本科生课程

《数据库系统原理》

数据库设计报告

实验名：酒店管理系统

组长：杜俊鹏

成员：朱卉祥



厦门大学数据库实验室

二零二四年六月

版 本 历 史

版本/状态	作者	参与者	起止日期	备注
V1.0 /初稿	杜俊鹏	朱卉祥	2024-06-24	
V1.1/终稿	杜俊鹏	朱卉祥	2024-06-29	

目 录

0. 文档介绍	4
0.1 文档目的	4
0.2 文档范围	4
0.3 术语与缩写解释	4
1. 数据库环境说明	5
2. 数据库的命名规则	5
3. 需求分析	5
4. 概念结构设计	7
5.逻辑设计	8
5.1 数据库逻辑图	8
5.2 数据库关系模型（红色为主键）	8
6. 物理设计	8
6.0 表汇总	8
7. 安全性设计	9
7.1 防止用户直接操作数据库的方法	9
8.数据库管理与维护说明	12
8.1 数据库转储（备份）	12
8.2 数据库恢复	12
8.3 注意事项	12

0. 文档介绍

此数据库设计文档是对酒店管理系统项目所需要用到数据库及其关系的具体设计以及对数据库安全性的设计与说明。

0.1 文档目的

为了开发酒店管理系统这个项目，我们需要详细设计出所有功能块所依赖的数据库结构，以支持酒店日常运营管理的各个方面，并模拟出简化的酒店业务流程。

本文档的编写是对酒店管理系统需求的抽象和细化的设计，旨在为开发团队提供明确的开发基础，以及在开发过程中的团队合作的有力保证。同时，本文档也将作为项目评审和验收的重要依据之一。

0.2 文档范围

本文档介绍了酒店管理系统数据库的设计，涵盖了：数据库逻辑结构设计（包括 E-R 模型转关系模型的转换）、物理结构设计（包括表的设计）、安全性设计（包括不同访问模式的设计）以及数据库管理及维护说明。

0.3 术语与缩写解释

使用到的专业术语	解释
UI 层	调用 BLL 层和实体类实现所需要的功能
BLL 层	业务逻辑层，用于连接 UI 层和 DAL 层
DAL 层	层存放访问数据库类的 DLL 工程
实体类	从数据库中的表，创建相应的类

1. 数据库环境说明

操作系统：Windows10

数据库系统：Microsoft SqlServer 2019

设计工具：Power Designer

2. 数据库的命名规则

系统名为 HotelMange，则数据库命名为 HotelManage。

数据库表命名规范，表名长度不能超过 30 个字符，表名中含有单词全部采用复数形式，单词首字母要大写，多个单词首个单词用缩写。

数据库字段命名规范，表别名+单词，每个单词首字母都大写。

3. 需求分析

3.1、引言

酒店管理系统数据库是酒店运营的核心组成部分，它负责存储、管理和分析酒店运营过程中产生的各类数据。一个优秀的酒店管理系统数据库不仅能提升酒店的运营效率，还能优化客户体验，增强酒店的竞争力。

3.2、需求分析

● 数据存储与管理

- 客户信息：包括客户姓名、联系方式、身份证号码、入住记录等。
- 房间信息：包括房间类型、房间号、房间状态（空闲、预订、入住等）、房间价格等。
- 预订记录：包括预订人信息、预订房间信息、预订时间、入住时间和离店时间等。
- 财务数据：包括客户账单、收支明细、财务报表等。

数据库应支持对这些数据的集中存储、管理和查询，确保数据的准确性和完整性。

● 数据分析与挖掘

- 客户需求分析：通过分析客户的历史入住记录、消费习惯等，了解客户的

需求和偏好，为酒店的营销策略提供支持。

- 市场趋势分析：通过对酒店运营数据的分析，了解市场趋势和竞争态势，为酒店的发展战略提供参考。
- 业务瓶颈分析：通过分析酒店运营数据，发现业务瓶颈和问题，为酒店的改进和优化提供依据。

数据库应提供强大的数据分析工具，支持对历史数据的深入分析和挖掘。

- 客户关系管理

- 客户信息档案：建立客户信息档案，记录客户的个人信息、入住记录、消费习惯等，方便酒店对客户进行精准营销和服务。
- 会员管理：建立会员制度，根据客户的消费金额和频率给予不同的会员等级和优惠，增强客户黏性。
- 积分管理：根据客户的消费金额给予积分奖励，积分可用于抵扣房费或兑换礼品等。

数据库应支持客户信息的集中存储和管理，方便酒店进行客户关系管理。

- 预订与房态管理

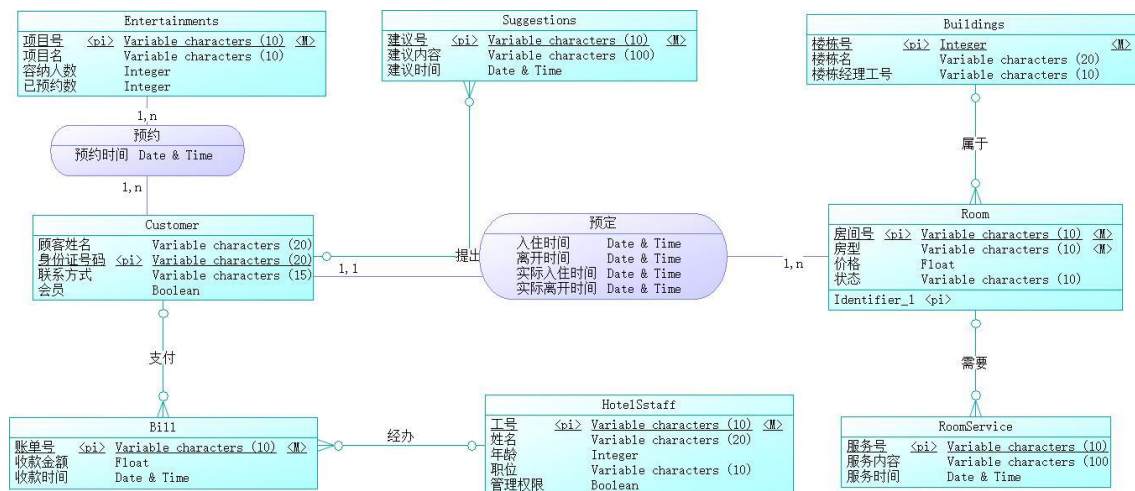
- 预订管理：支持在线预订、电话预订等多种预订方式，并能实时更新预订信息。
- 房态管理：实时更新房间的预订情况和状态，确保酒店能够准确掌握房态信息，提高房间利用率。

数据库应支持预订信息和房态信息的实时更新和查询。

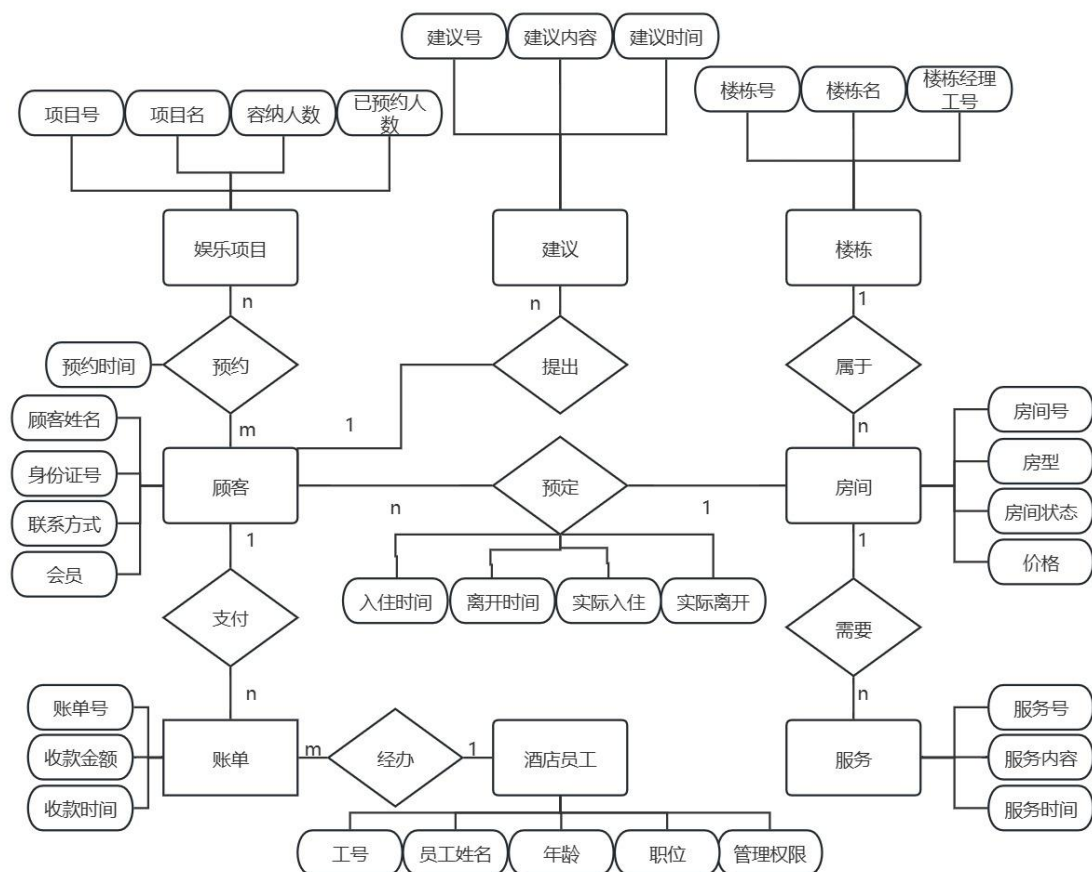
- 性能需求

- 稳定性：数据库应具有高可用性、高并发处理能力和容错机制，确保酒店管理系统在高峰期能够稳定运行，避免因数据库故障导致的业务中断。
- 数据安全性：数据库应提供数据加密、访问控制等安全功能，确保酒店数据在存储和传输过程中的安全性。同时，数据库应具备备份和恢复机制，以应对数据丢失或意外情况。
- 易用性和可扩展性：数据库产品应具有友好的用户界面和易于操作的管理工具，降低酒店员工的学习成本。此外，数据库应具备良好的可扩展性，以满足酒店业务不断增长的需求。

4. 概念结构设计



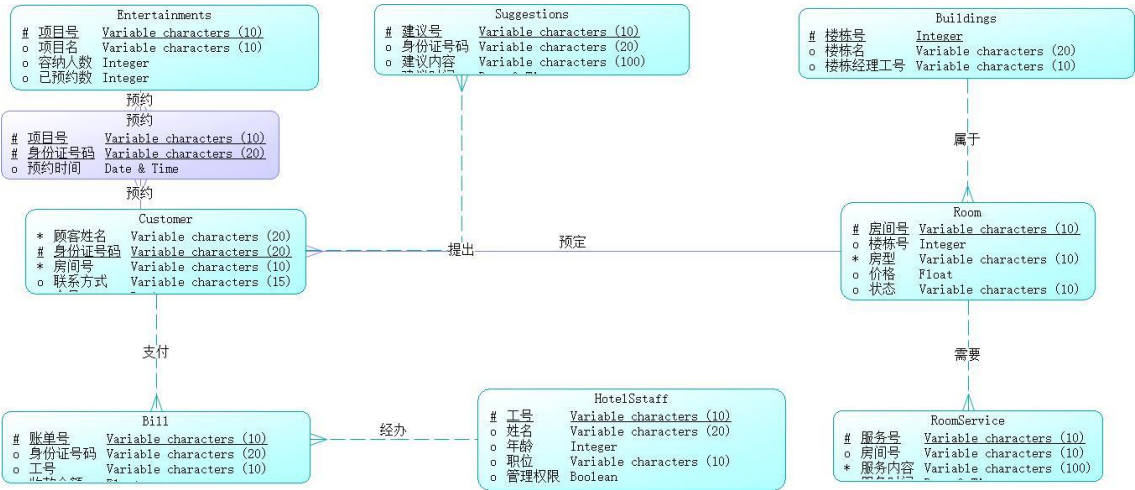
概念数据模型



E-R 模型

5.逻辑设计

5.1 数据库逻辑图



逻辑数据模型

5.2 数据库关系模型（红色为主键）

顾客（顾客姓名，**身份证号码**，联系方式，会员，入住时间，离开时间，实际入住时间，房间号，密码）外码：房间号

房间（**房间号**，楼栋号，房型，价格，房间状态）外码：楼栋号

楼栋（**楼栋号**，楼栋名，楼栋经理工号）

客房服务（**服务号**，房间号，服务内容，服务发布时间）外码：房间号

娱乐项目（**项目号**，项目名，容纳人数，已预约人数）

预约信息（**预约号**，项目号，身份证号码，预约时间）外码：项目号，身份证号码

账单信息（**账单号**，支付人身份证号码，收款金额，收款时间，经办人工号）外码：支付人身份证号码，经办人工号

酒店员工（**工号**，姓名，年龄，职位，管理权限，密码）

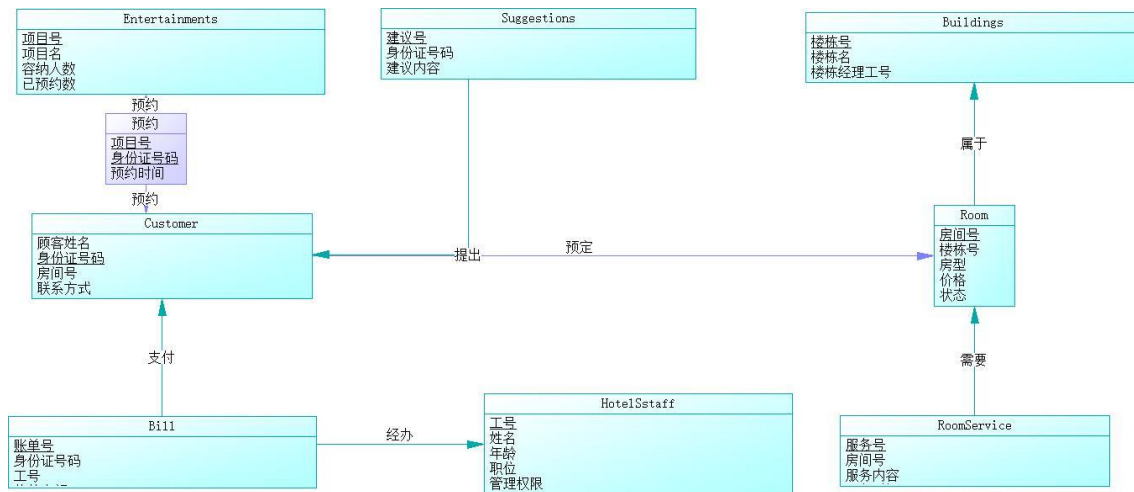
反馈表（**建议号**，身份证号码，建议内容，建议时间）外码：身份证号码

6. 物理设计

6.0 表汇总

表 6.1 数据库：ExamManage 有表信息

表 名	功 能 说 明
Customer	存放顾客信息
Room	存放房间信息
Entertainments	存放娱乐项目信息
Reservation	存放预约娱乐项目信息
Suggestions	存放建议与反馈信息
HotelStaff	存放酒店员工信息
Bill	存放账单信息
Buildings	存放楼栋信息
RoomService	存放客房服务信息



7. 安全性设计

7.1 防止用户直接操作数据库的方法

防止用户直接操作数据库的方法是设置三种访问模式，顾客模式、经理模式、员工模式，通过登录界面可登录不同模式。三种模式各自实现一部分功能，功能如下。

● 顾客模式：

个人信息
客房服务
娱乐项目预约
账单支付（到前台）
建议与反馈
修改密码
退出

● 经理模式：

顾客查询
房间查询
办理入住
结账
办理离开
员工管理
客房服务记录
预约记录
反馈记录
退出

- 普通员工模式：

顾客查询
房间查询
办理入住
结账
办理离开
客房服务记录
预约记录
反馈记录
退出

其中经理和普通员工模式的区别在于：

顾客查询功能（只能实现单个查询，不能直接查看所有顾客信息，所以只保留一个选项卡）

房间查询功能（只能给出所有空闲房间信息，所以只保留一个选项卡）

办理入住功能（相同）

结账功能（相同）

办理离开功能（相同）

没有员工管理功能

查看客房服务记录功能（相同）

查看反馈记录功能（相同）

退出功能（相同）

这样不仅保持了普通员工进行日常工作，而且防止了员工进行越权操作，使数据库更加安全。

8.数据库管理与维护说明

为确保数据库在遭遇故障时能够迅速恢复，减少数据损失和业务中断的风险，实施有效的数据库转储和恢复策略至关重要。以下将详细说明如何使用 Microsoft SQL Server 2019 进行数据库的转储（备份）和恢复操作。

8.1 数据库转储（备份）

转储，即备份，是将数据库中的数据和结构信息以特定格式保存至外部存储介质的过程。在 SQL Server 2019 中，可以使用其内置的备份工具或 T-SQL 语句来执行此操作。以下是备份的基本步骤：

1. **选择备份类型：**根据需求，选择完全备份、差异备份、事务日志备份或文件/文件组备份。
2. **设置备份目标：**指定备份文件（如 DAT 文件）的存储位置和文件名。
3. **执行备份：**使用 SQL Server Management Studio (SSMS) 或 T-SQL BACKUP DATABASE 语句来执行备份操作。

8.2 数据库恢复

当数据库发生故障或数据丢失时，可以使用之前创建的备份文件来恢复数据库。以下是恢复的基本步骤：

1. **选择恢复模式：**根据备份的类型和数据库的当前状态，选择适当的恢复模式，如完整恢复、时间点恢复等。
2. **指定恢复源：**选择之前创建的备份文件作为恢复源。
3. **执行恢复：**使用 SSMS 或 T-SQL RESTORE DATABASE 语句来执行恢复操作。
在恢复过程中，您可能需要指定恢复选项，如是否覆盖现有数据库、是否保留现有事务日志等。

8.3 注意事项

1. **定期备份：**为了最大程度地减少数据损失，建议定期（如每天、每周或每月）对数据库进行备份。
2. **验证备份：**定期验证备份文件的完整性和可恢复性，确保在需要时能够成功恢复数据库。
3. **备份存储：**将备份文件存储在安全、可靠的位置，并考虑使用冗余存储或远程备份来防止数据丢失。

4. **备份策略：**根据业务需求和数据重要性制定合适的备份策略，包括备份类型、频率、保留期限等。
5. **恢复演练：**定期进行恢复演练，测试备份文件的可恢复性和恢复过程的有效性。这有助于确保在真正发生故障时能够迅速、准确地恢复数据库。