|  |
| --- |
|  |

**数据库原理与实践期末项目**

**学号：**

**姓名：**



数据库原理与实践期末项目

20 —20 学年 学期

|  |  |
| --- | --- |
| 课程名称： |  |
| 专业学院： |  |
| 专业班级： |  |
| 学 号： |  |
| 姓 名： |  |
| 指导教师： |  |

目 录

[1、 项目背景及需求分析 1](#_Toc153742448)

[1.1项目背景 1](#_Toc153742449)

[1.2需求分析 1](#_Toc153742450)

[2、 概念结构设计 2](#_Toc153742451)

[2.1抽象出系统实体 2](#_Toc153742452)

[2.2局部E-R图 2](#_Toc153742453)

[2.3全局E-R图 2](#_Toc153742454)

[3、 逻辑结构设计 3](#_Toc153742455)

[3.1简单说明 3](#_Toc153742456)

[3.2关系模式 3](#_Toc153742457)

[3.3表结构 3](#_Toc153742458)

[4、 创建数据库及SQL相关操作 4](#_Toc153742459)

[4.1 创建数据库 4](#_Toc153742460)

[4.1.1 创建表 4](#_Toc153742461)

[4.1.2 添加记录 4](#_Toc153742462)

[4.1.3 修改记录 5](#_Toc153742463)

[4.2 删除记录 5](#_Toc153742464)

[4.3查询问题及查询结果 5](#_Toc153742465)

[4.3.1 选择列 5](#_Toc153742466)

[4.3.2函数使用 5](#_Toc153742467)

[4.3.3 LIKE子句 6](#_Toc153742468)

[4.3.4 条件查询 6](#_Toc153742469)

[4.4.5多表查询 6](#_Toc153742470)

[4.4.6子查询 6](#_Toc153742471)

[4.5建立视图 7](#_Toc153742472)

[4.5.1 定义视图 7](#_Toc153742473)

[4.5.2利用视图插入数据 7](#_Toc153742474)

[5、建立存储过程及函数 8](#_Toc153742475)

[5.1创建存储过程 8](#_Toc153742476)

[5.2存储过程应用 8](#_Toc153742477)

[6、建立触发器 9](#_Toc153742478)

[7、项目设计总结 10](#_Toc153742479)

# 1、 项目背景及需求分析

## 1.1项目背景

实验运用数据库技术构建高校宿舍管理系统，实现对学生宿舍信息、学生信息、宿舍分配、报修管理等功能的有效管理，旨在帮助学生深入理解数据库管理系统的基本概念、原理和操作，熟练掌握数据库设计、创建、数据操作以及数据查询等关键技能，培养学生运用数据库技术解决实际问题的能力，并提高学生的动手实践和独立思考能力。

## 1.2需求分析

（一）学生信息管理

• 学生基本信息录入：包括姓名、性别、学号、专业、班级、联系方式等，支持批量导入功能。

• 信息查询与修改：可根据学号、姓名等关键词快速查询学生信息，并对信息进行修改、更新，如联系方式变更、专业调整等。

• 学生入住与毕业退房办理：记录学生入住时间、宿舍分配信息，在毕业退房时进行相关费用的查询和结算、信息登记。

（二）宿舍信息管理

• 宿舍基本信息维护：如宿舍编号、楼层、房间类型（单人间、多人间等）、可容纳人数、家具及设施配备情况等。

• 宿舍分配与调整：根据学生性别、专业、年级等因素进行宿舍分配，可手动调整宿舍安排，并记录分配历史。

• 空宿舍查询：方便管理人员查看当前空宿舍情况，以便安排新生入住或进行宿舍调整。

（三）设施设备管理

• 设施设备登记：对宿舍内的家具、电器、门窗等设施设备进行详细登记，包括品牌、型号、购置时间、使用状态等。

• 设施设备维修管理：学生可在线申报设施设备损坏情况，管理人员接收申报后安排维修人员进行维修，并记录维修进度、维修费用、维修人员等信息，维修完成后进行验收。

• 设施设备报废处理：对于无法修复或达到使用年限的设施设备，进行报废登记，记录报废原因、处理方式等。

（四）费用管理

• 费用项目设定：包括住宿费、水电费、物业费等，可设置不同费用的收费标准、计费周期。

• 费用收缴记录：记录学生缴费时间、缴费金额、缴费方式等信息，支持多种缴费方式（如线上支付、线下缴费）的对接与数据录入。

• 欠费提醒与催缴：系统自动生成欠费学生名单，并通过短信、系统消息等方式提醒学生缴费，可打印催缴通知单。

（五）访客管理

• 访客登记：访客进入公寓时，需登记姓名、身份证号、联系方式、访问对象、访问时间等信息，可通过扫描身份证或手动录入。

• 访客权限设置：根据宿舍学生的授权，设置访客的访问权限，如可访问的楼层、宿舍范围、访问时长等。

• 访客记录查询与统计：管理人员可查询历史访客记录，统计访客流量、访问高峰时段等信息，以便加强安全管理。

（六）系统管理

• 用户管理：创建不同角色的用户账号，如管理员、宿管人员、学生等，并分配相应的操作权限，如管理员拥有系统的全面管理权限，宿管人员负责所在楼栋的日常管理操作，学生可进行个人信息查询、费用查询、维修申报等操作。

• 数据备份与恢复：定期对系统数据进行备份，确保数据的安全性和完整性，在数据丢失或损坏时可进行数据恢复操作。

• 日志管理：记录系统操作日志，包括用户登录时间、操作内容、操作结果等信息，以便进行系统操作审计与问题追溯。

# 2、 概念结构设计

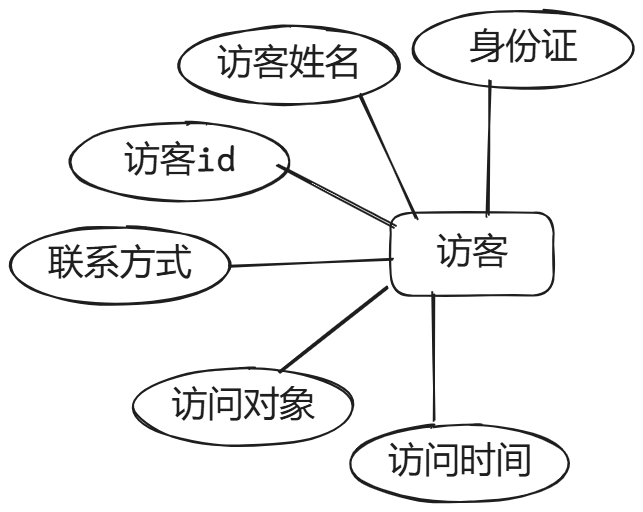
## 2.1抽象出系统实体

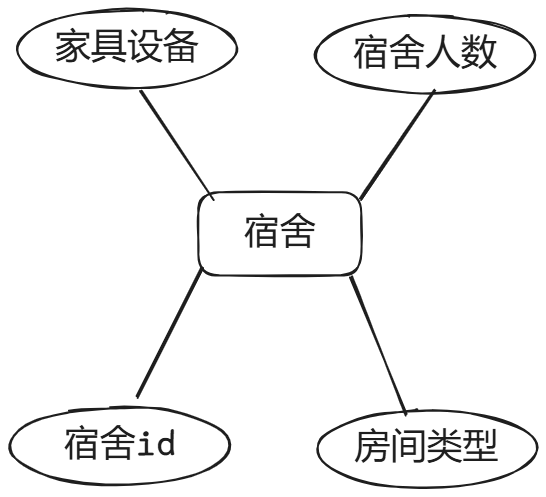
根据需求分析，抽象出你们的系统所用到的实体，要求至少要有4个或4个以上实体，包含实体本身属性设置。

简单描述

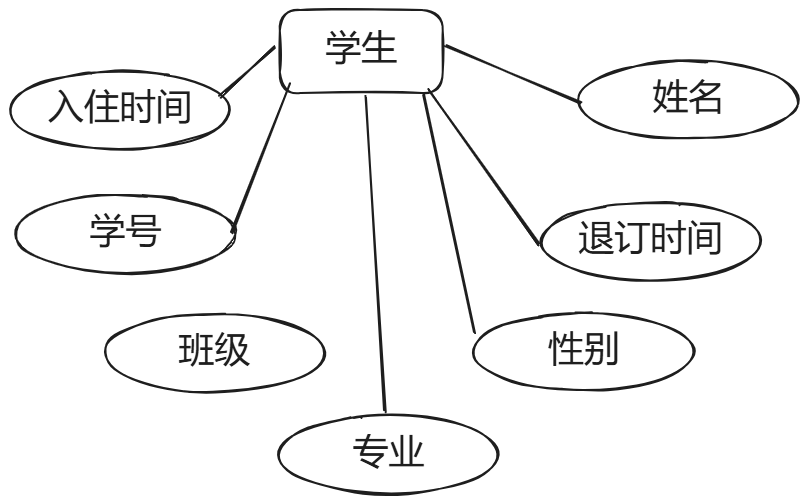
## 2.2局部E-R图

1、访客：

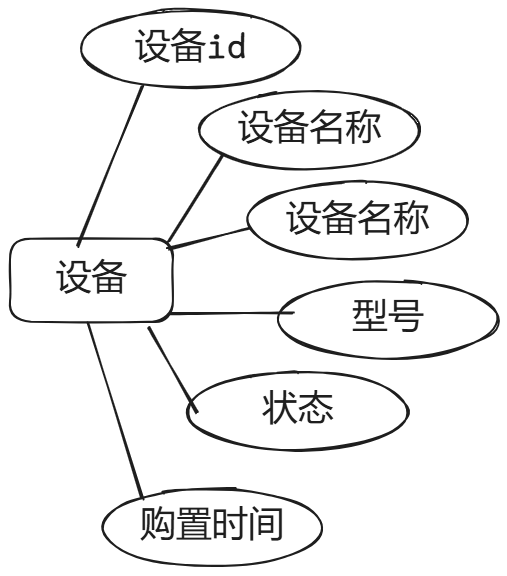


2、宿舍

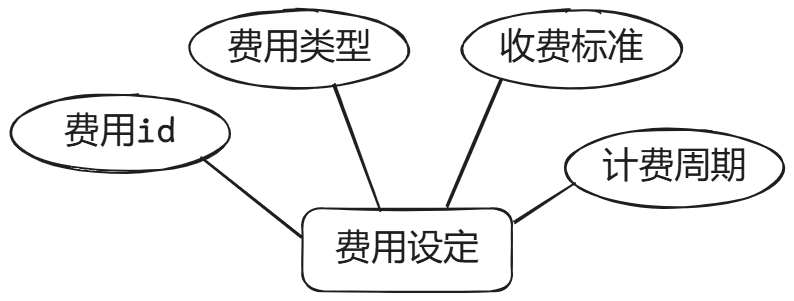
3、学生



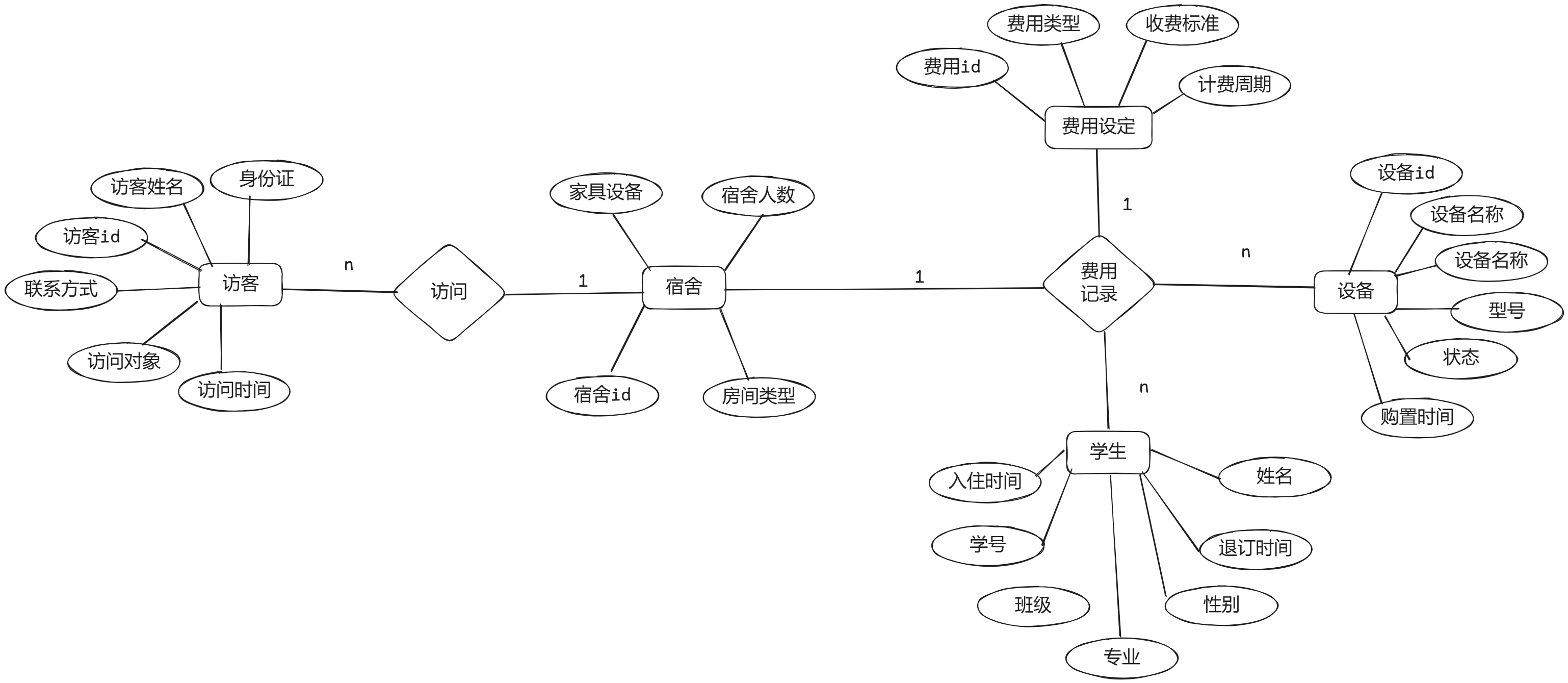
4、设备



5、费用设定：



## 2.3全局E-R图



# 3、 逻辑结构设计

## 3.1简单说明

宿舍管理系统设计的逻辑结构涵盖了宿舍管理、学生管理、设施管理、费用管理、访客管理、用户管理及系统日志记录。通过关系数据库的方式组织和存储数据，各表之间通过主键和外键建立联系，以满足多样化的管理需求。

## 3.2关系模式

**1. 宿舍表 (dormitories)**

**关系模式**:  
宿舍 (dorm\_id, floor, room\_type, capacity, current\_occupants, furniture\_info, allocation\_history)

* **主键**: dorm\_id
* **外键**: 无

**2. 学生表 (students)**

**关系模式**:  
学生 (student\_id, name, gender, major, class, contact, dorm\_id, check\_in\_date, check\_out\_date, encrypted\_contact)

* **主键**: student\_id
* **外键**: dorm\_id → 宿舍表 (dormitories.dorm\_id)

**3. 设施表 (facilities)**

**关系模式**:  
设施 (facility\_id, dorm\_id, name, brand, model, purchase\_date, status)

* **主键**: facility\_id
* **外键**: dorm\_id → 宿舍表 (dormitories.dorm\_id)

**4. 维修申请表 (maintenance\_requests)**

**关系模式**:  
维修申请 (request\_id, facility\_id, report\_date, description, status, repair\_date, repair\_cost, repair\_person)

* **主键**: request\_id
* **外键**: facility\_id → 设施表 (facilities.facility\_id)

**5. 费用设置表 (fee\_settings)**

**关系模式**:  
费用设置 (fee\_id, fee\_type, amount, billing\_cycle)

* **主键**: fee\_id
* **外键**: 无

**6. 费用记录表 (fee\_records)**

**关系模式**:  
费用记录 (record\_id, student\_id, fee\_id, payment\_date, amount, payment\_method)

* **主键**: record\_id
* **外键**:
  + student\_id → 学生表 (students.student\_id)
  + fee\_id → 费用设置表 (fee\_settings.fee\_id)

**7. 访客表 (visitors)**

**关系模式**:  
访客 (visitor\_id, name, id\_card, contact, visit\_target, visit\_time, access\_level, duration)

* **主键**: visitor\_id
* **外键**: visit\_target → 学生表 (students.student\_id)

**8. 用户表 (users)**

**关系模式**:  
用户 (user\_id, username, password\_hash, role, permissions)

* **主键**: user\_id
* **外键**: 无

**9. 日志表 (logs)**

**关系模式**:  
日志 (log\_id, user\_id, action, action\_time, result)

* **主键**: log\_id
* **外键**: user\_id → 用户表 (users.user\_id)

## 3.3表结构

**1. 宿舍表 (dormitories)**

| **字段名** | **数据类型** | **说明** |
| --- | --- | --- |
| dorm\_id | INT | 宿舍编号 (主键) |
| floor | INT | 宿舍楼层 |
| room\_type | ENUM | 房间类型 |
| capacity | INT | 可容纳人数 |
| current\_occupants | INT | 当前入住人数 |
| furniture\_info | TEXT | 家具设施信息 |
| allocation\_history | TEXT | 分配历史记录 |

**2. 学生表 (students)**

| **字段名** | **数据类型** | **说明** |
| --- | --- | --- |
| student\_id | INT | 学生编号 (主键) |
| name | VARCHAR(50) | 姓名 |
| gender | ENUM | 性别 |
| major | VARCHAR(100) | 专业 |
| class | VARCHAR(50) | 班级 |
| contact | VARCHAR(15) | 联系方式 |
| dorm\_id | INT | 宿舍编号 |
| check\_in\_date | DATE | 入住时间 |
| check\_out\_date | DATE | 退房时间 |
| encrypted\_contact | TEXT | 加密联系方式 |

**3. 设施表 (facilities)**

| **字段名** | **数据类型** | **说明** |
| --- | --- | --- |
| facility\_id | INT | 设备编号 (主键) |
| dorm\_id | INT | 宿舍编号 |
| name | VARCHAR(100) | 设备名称 |
| brand | VARCHAR(50) | 品牌 |
| model | VARCHAR(50) | 型号 |
| purchase\_date | DATE | 购置时间 |
| status | ENUM | 状态 |

**4. 维修申请表 (maintenance\_requests)**

| **字段名** | **数据类型** | **说明** |
| --- | --- | --- |
| request\_id | INT | 申请编号 (主键) |
| facility\_id | INT | 设备编号 |
| report\_date | DATE | 报修日期 |
| description | TEXT | 问题描述 |
| status | ENUM | 维修状态 |
| repair\_date | DATE | 完成日期 |
| repair\_cost | DECIMAL(10, 2) | 维修费用 |
| repair\_person | VARCHAR(50) | 维修人员 |

**5. 费用设置表 (fee\_settings)**

| **字段名** | **数据类型** | **说明** |
| --- | --- | --- |
| fee\_id | INT | 费用编号 (主键) |
| fee\_type | ENUM | 费用类型 |
| amount | DECIMAL(10, 2) | 收费标准 |
| billing\_cycle | ENUM | 计费周期 |

**6. 费用记录表 (fee\_records)**

| **字段名** | **数据类型** | **说明** |
| --- | --- | --- |
| record\_id | INT | 记录编号 (主键) |
| student\_id | INT | 学号 |
| fee\_id | INT | 费用编号 |
| payment\_date | DATE | 缴费时间 |
| amount | DECIMAL(10, 2) | 缴费金额 |
| payment\_method | ENUM | 缴费方式 |

**7. 访客表 (visitors)**

| **字段名** | **数据类型** | **说明** |
| --- | --- | --- |
| visitor\_id | INT | 访客编号 (主键) |
| name | VARCHAR(50) | 姓名 |
| id\_card | VARCHAR(18) | 身份证号 |
| contact | VARCHAR(15) | 联系方式 |
| visit\_target | INT | 访问对象 (学号) |
| visit\_time | DATETIME | 访问时间 |
| access\_level | TEXT | 访问权限 |
| duration | INT | 访问时长 (分钟) |

**8. 用户表 (users)**

| **字段名** | **数据类型** | **说明** |
| --- | --- | --- |
| user\_id | INT | 用户编号 (主键) |
| username | VARCHAR(50) | 用户名 |
| password\_hash | TEXT | 哈希密码 |
| role | ENUM | 用户角色 |
| permissions | TEXT | 权限列表 |

**9. 日志表 (logs)**

| **字段名** | **数据类型** | **说明** |
| --- | --- | --- |
| log\_id | INT | 日志编号 (主键) |
| user\_id | INT | 操作用户编号 |
| action | TEXT | 操作内容 |
| action\_time | DATETIME | 操作时间 |
| result | ENUM | 操作结果 |

# 4、 创建数据库及SQL相关操作

## 4.1 创建数据库

建立数据库，数据库名称自取，设置字符集为UTF-8,排序规则为utf8\_general\_ci。

代码：

-- 创建数据库，名称为dorm\_management

CREATE DATABASE dorm\_management CHARACTER SET utf8 COLLATE utf8\_general\_ci;

-- 使用该数据库

USE dorm\_management;

截图结果：



### 4.1.1 创建表

根据需要，在你们创建的数据库中创建你们的每个数据表。

CREATE TABLE dormitories (

dorm\_id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '宿舍编号',

floor INT NOT NULL COMMENT '楼层',

room\_type ENUM('单人间', '双人间', '多人间') NOT NULL COMMENT '房间类型',

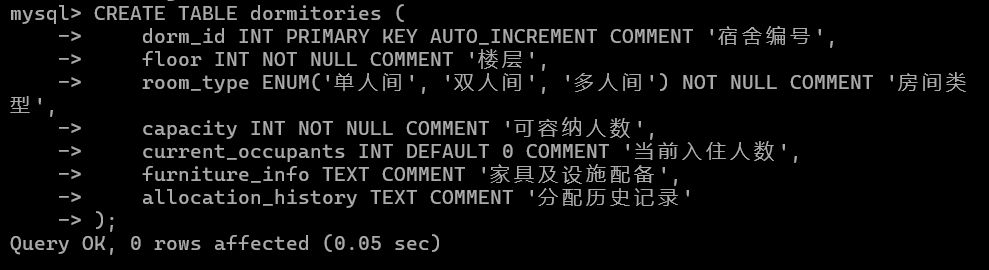
capacity INT NOT NULL COMMENT '可容纳人数',

current\_occupants INT DEFAULT 0 COMMENT '当前入住人数',

furniture\_info TEXT COMMENT '家具及设施配备',

allocation\_history TEXT COMMENT '分配历史记录'

);



CREATE TABLE students (

student\_id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '学号',

name VARCHAR(50) NOT NULL COMMENT '姓名',

gender ENUM('男', '女') NOT NULL COMMENT '性别',

major VARCHAR(100) COMMENT '专业',

class VARCHAR(50) COMMENT '班级',

contact VARCHAR(15) NOT NULL COMMENT '联系方式',

dorm\_id INT DEFAULT NULL COMMENT '宿舍编号',

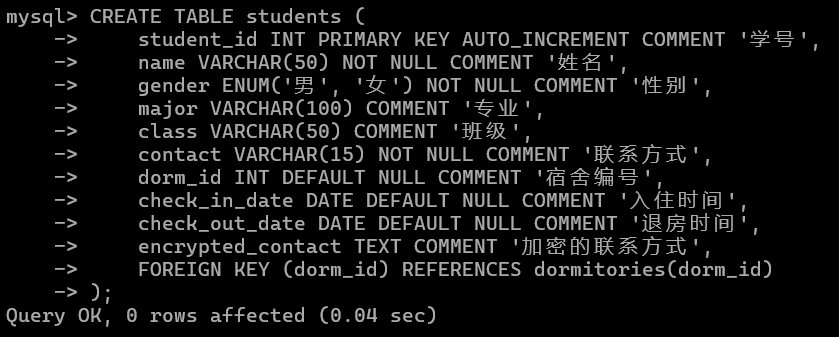
check\_in\_date DATE DEFAULT NULL COMMENT '入住时间',

check\_out\_date DATE DEFAULT NULL COMMENT '退房时间',

encrypted\_contact TEXT COMMENT '加密的联系方式',

FOREIGN KEY (dorm\_id) REFERENCES dormitories(dorm\_id)

);



CREATE TABLE facilities (

facility\_id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '设备编号',

dorm\_id INT NOT NULL COMMENT '宿舍编号',

name VARCHAR(100) NOT NULL COMMENT '设备名称',

brand VARCHAR(50) COMMENT '品牌',

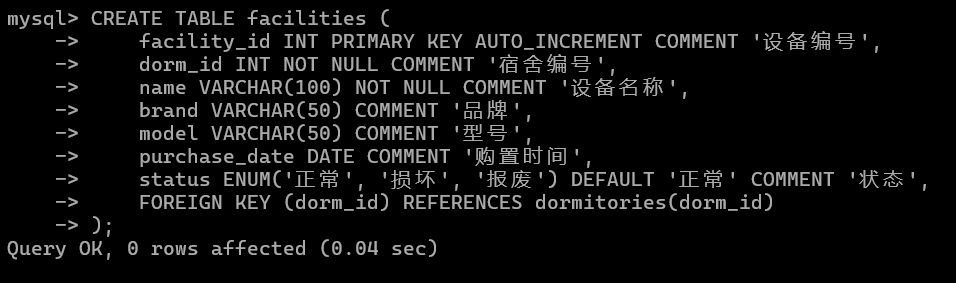
model VARCHAR(50) COMMENT '型号',

purchase\_date DATE COMMENT '购置时间',

status ENUM('正常', '损坏', '报废') DEFAULT '正常' COMMENT '状态',

FOREIGN KEY (dorm\_id) REFERENCES dormitories(dorm\_id)

);



CREATE TABLE maintenance\_requests (

request\_id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '申请编号',

facility\_id INT NOT NULL COMMENT '设备编号',

report\_date DATE NOT NULL COMMENT '报修日期',

description TEXT COMMENT '问题描述',

status ENUM('待处理', '维修中', '已完成') DEFAULT '待处理' COMMENT '状态',

repair\_date DATE DEFAULT NULL COMMENT '完成日期',

repair\_cost DECIMAL(10, 2) DEFAULT NULL COMMENT '维修费用',

repair\_person VARCHAR(50) DEFAULT NULL COMMENT '维修人员',

FOREIGN KEY (facility\_id) REFERENCES facilities(facility\_id)

);



CREATE TABLE fee\_settings (

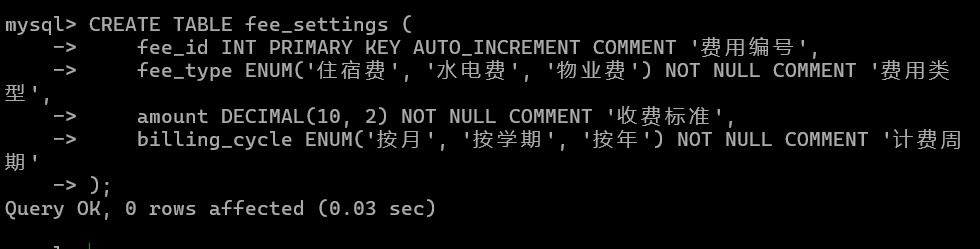
fee\_id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '费用编号',

fee\_type ENUM('住宿费', '水电费', '物业费') NOT NULL COMMENT '费用类型',

amount DECIMAL(10, 2) NOT NULL COMMENT '收费标准',

billing\_cycle ENUM('按月', '按学期', '按年') NOT NULL COMMENT '计费周期'

);



CREATE TABLE fee\_records (

record\_id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '记录编号',

student\_id INT NOT NULL COMMENT '学号',

fee\_id INT NOT NULL COMMENT '费用编号',

payment\_date DATE DEFAULT NULL COMMENT '缴费时间',

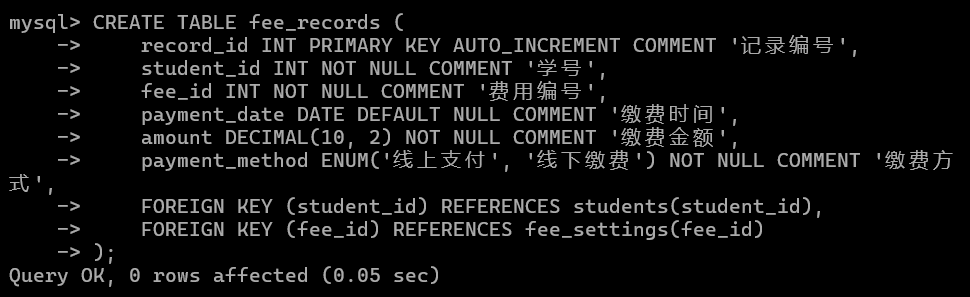
amount DECIMAL(10, 2) NOT NULL COMMENT '缴费金额',

payment\_method ENUM('线上支付', '线下缴费') NOT NULL COMMENT '缴费方式',

FOREIGN KEY (student\_id) REFERENCES students(student\_id),

FOREIGN KEY (fee\_id) REFERENCES fee\_settings(fee\_id)

);



CREATE TABLE visitors (

visitor\_id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '访客编号',

name VARCHAR(50) NOT NULL COMMENT '姓名',

id\_card VARCHAR(18) NOT NULL COMMENT '身份证号',

contact VARCHAR(15) NOT NULL COMMENT '联系方式',

visit\_target INT NOT NULL COMMENT '访问对象（学生编号）',

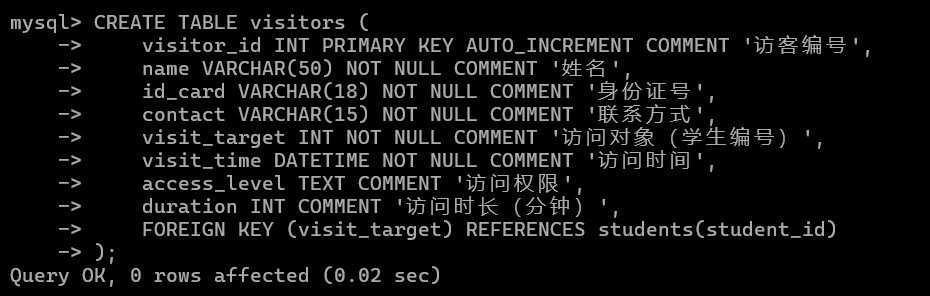
visit\_time DATETIME NOT NULL COMMENT '访问时间',

access\_level TEXT COMMENT '访问权限',

duration INT COMMENT '访问时长（分钟）',

FOREIGN KEY (visit\_target) REFERENCES students(student\_id)

);



CREATE TABLE users (

user\_id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '用户编号',

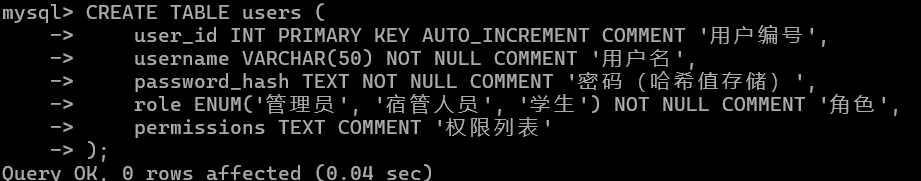
username VARCHAR(50) NOT NULL COMMENT '用户名',

password\_hash TEXT NOT NULL COMMENT '密码（哈希值存储）',

role ENUM('管理员', '宿管人员', '学生') NOT NULL COMMENT '角色',

permissions TEXT COMMENT '权限列表'

);



CREATE TABLE logs (

log\_id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '日志编号',

user\_id INT NOT NULL COMMENT '操作用户编号',

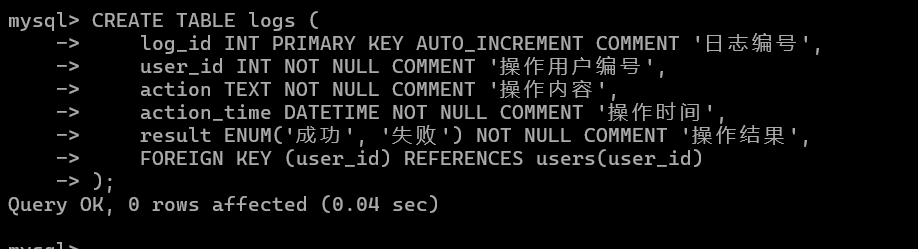
action TEXT NOT NULL COMMENT '操作内容',

action\_time DATETIME NOT NULL COMMENT '操作时间',

result ENUM('成功', '失败') NOT NULL COMMENT '操作结果',

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);



### 4.1.2 添加记录

-- 插入宿舍表记录

INSERT INTO dormitories (floor, room\_type, capacity, current\_occupants, furniture\_info, allocation\_history)

VALUES

(1, '单人间', 1, 0, '床, 衣柜, 桌椅', ''),

(1, '双人间', 2, 2, '床, 衣柜, 桌椅', '历史记录1'),

(2, '多人间', 4, 4, '床, 衣柜, 桌椅, 空调', '历史记录2'),

(2, '单人间', 1, 1, '床, 衣柜, 桌椅, 热水器', ''),

(3, '双人间', 2, 1, '床, 衣柜, 桌椅', '历史记录3'),

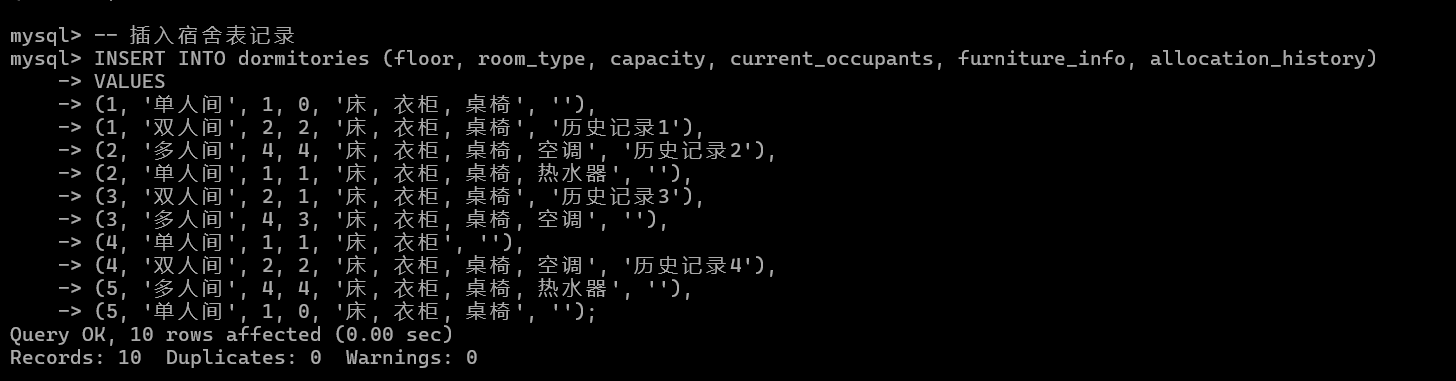
(3, '多人间', 4, 3, '床, 衣柜, 桌椅, 空调', ''),

(4, '单人间', 1, 1, '床, 衣柜', ''),

(4, '双人间', 2, 2, '床, 衣柜, 桌椅, 空调', '历史记录4'),

(5, '多人间', 4, 4, '床, 衣柜, 桌椅, 热水器', ''),

(5, '单人间', 1, 0, '床, 衣柜, 桌椅', '');



-- 插入学生表记录

INSERT INTO students (name, gender, major, class, contact, dorm\_id, check\_in\_date, check\_out\_date, encrypted\_contact)

VALUES

('张三', '男', '计算机科学', '计科1班', '13800138000', 1, '2024-09-01', NULL, 'encrypted1'),

('李四', '女', '电子工程', '电工2班', '13800138001', 2, '2024-09-01', NULL, 'encrypted2'),

('王五', '男', '机械设计', '机械3班', '13800138002', 3, '2024-09-01', NULL, 'encrypted3'),

('赵六', '女', '数学', '数学1班', '13800138003', 4, '2024-09-01', NULL, 'encrypted4'),

('钱七', '男', '物理', '物理1班', '13800138004', 5, '2024-09-01', NULL, 'encrypted5'),

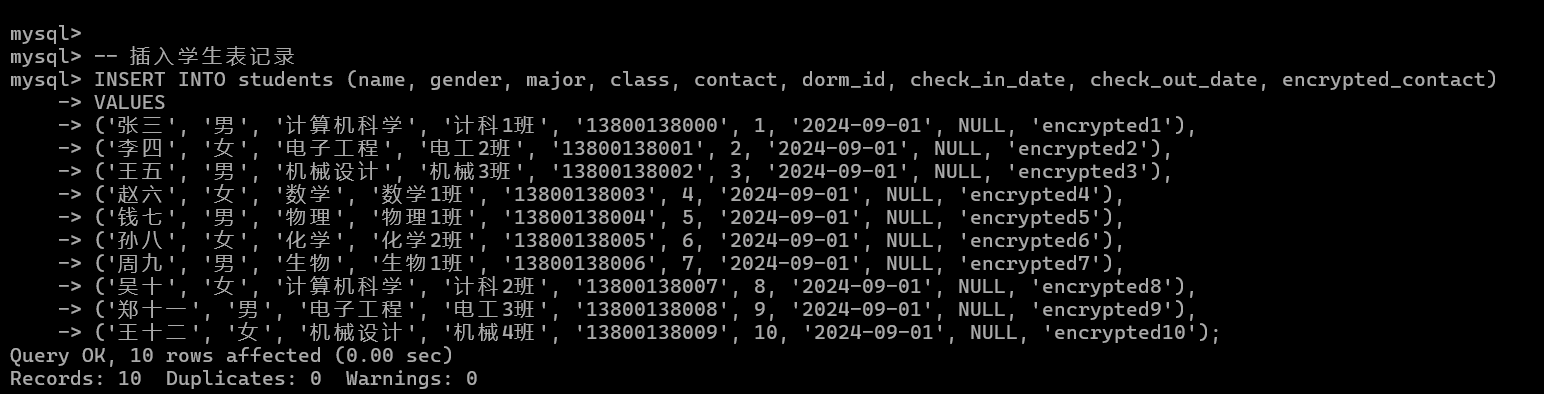
('孙八', '女', '化学', '化学2班', '13800138005', 6, '2024-09-01', NULL, 'encrypted6'),

('周九', '男', '生物', '生物1班', '13800138006', 7, '2024-09-01', NULL, 'encrypted7'),

('吴十', '女', '计算机科学', '计科2班', '13800138007', 8, '2024-09-01', NULL, 'encrypted8'),

('郑十一', '男', '电子工程', '电工3班', '13800138008', 9, '2024-09-01', NULL, 'encrypted9'),

('王十二', '女', '机械设计', '机械4班', '13800138009', 10, '2024-09-01', NULL, 'encrypted10');



-- 插入设施表记录

INSERT INTO facilities (dorm\_id, name, brand, model, purchase\_date, status)

VALUES

(1, '空调', '格力', 'KFR-35GW', '2023-06-01', '正常'),

(2, '热水器', '美的', 'JSQ24', '2023-06-01', '正常'),

(3, '冰箱', '海尔', 'BCD-206', '2023-06-01', '正常'),

(4, '电视', '创维', 'S82', '2023-06-01', '正常'),

(5, '微波炉', '松下', 'NN-DS592', '2023-06-01', '正常'),

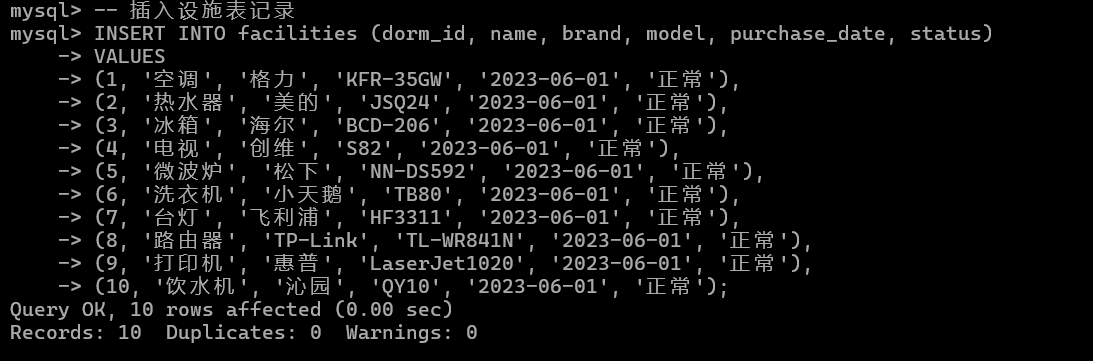
(6, '洗衣机', '小天鹅', 'TB80', '2023-06-01', '正常'),

(7, '台灯', '飞利浦', 'HF3311', '2023-06-01', '正常'),

(8, '路由器', 'TP-Link', 'TL-WR841N', '2023-06-01', '正常'),

(9, '打印机', '惠普', 'LaserJet1020', '2023-06-01', '正常'),

(10, '饮水机', '沁园', 'QY10', '2023-06-01', '正常');



-- 插入维修申请表记录

INSERT INTO maintenance\_requests (facility\_id, report\_date, description, status, repair\_date, repair\_cost, repair\_person)

VALUES

(1, '2024-10-01', '空调不制冷', '维修中', NULL, NULL, NULL),

(2, '2024-10-02', '热水器漏水', '待处理', NULL, NULL, NULL),

(3, '2024-10-03', '冰箱不制冷', '待处理', NULL, NULL, NULL),

(4, '2024-10-04', '电视无信号', '维修中', NULL, NULL, NULL),

(5, '2024-10-05', '微波炉不加热', '已完成', '2024-10-06', 50.00, '维修员1'),

(6, '2024-10-06', '洗衣机异响', '维修中', NULL, NULL, NULL),

(7, '2024-10-07', '台灯不亮', '待处理', NULL, NULL, NULL),

(8, '2024-10-08', '路由器无法连接', '维修中', NULL, NULL, NULL),

(9, '2024-10-09', '打印机卡纸', '待处理', NULL, NULL, NULL),

(10, '2024-10-10', '饮水机漏水', '维修中', NULL, NULL, NULL);



-- 插入费用设置表记录

INSERT INTO fee\_settings (fee\_type, amount, billing\_cycle)

VALUES

('住宿费', 1200.00, '按学期'),

('水电费', 50.00, '按月'),

('物业费', 100.00, '按学期'),

('住宿费', 2400.00, '按年'),

('水电费', 100.00, '按月'),

('物业费', 200.00, '按年'),

('住宿费', 600.00, '按学期'),

('水电费', 30.00, '按月'),

('物业费', 150.00, '按年'),

('住宿费', 1500.00, '按学期');



-- 插入费用记录表记录

INSERT INTO fee\_records (student\_id, fee\_id, payment\_date, amount, payment\_method)

VALUES

(1, 1, '2024-10-01', 1200.00, '线上支付'),

(2, 2, '2024-10-02', 50.00, '线下缴费'),

(3, 3, '2024-10-03', 100.00, '线上支付'),

(4, 4, '2024-10-04', 2400.00, '线下缴费'),

(5, 5, '2024-10-05', 100.00, '线上支付'),

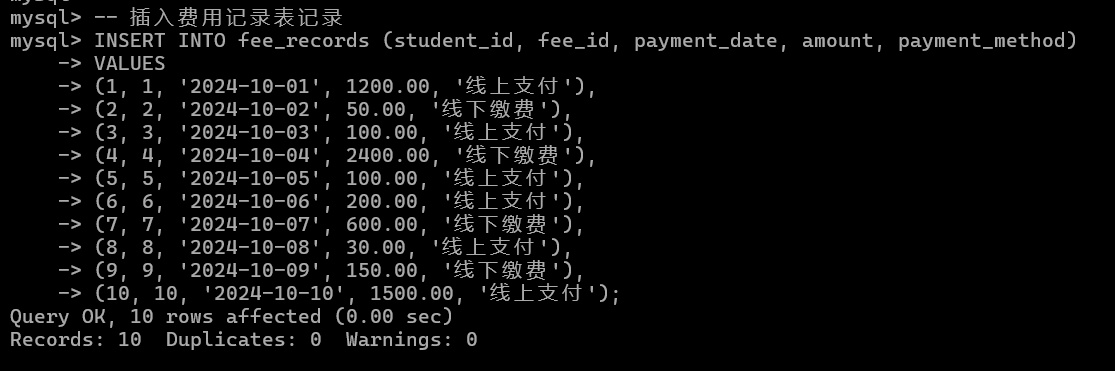
(6, 6, '2024-10-06', 200.00, '线上支付'),

(7, 7, '2024-10-07', 600.00, '线下缴费'),

(8, 8, '2024-10-08', 30.00, '线上支付'),

(9, 9, '2024-10-09', 150.00, '线下缴费'),

(10, 10, '2024-10-10', 1500.00, '线上支付');



-- 插入访客表记录

INSERT INTO visitors (name, id\_card, contact, visit\_target, visit\_time, access\_level, duration)

VALUES

('李访客1', '110101199001011234', '13800138010', 1, '2024-10-01 10:00:00', '普通访客', 60),

('张访客2', '110101199002021234', '13800138011', 2, '2024-10-02 10:00:00', '普通访客', 30),

('王访客3', '110101199003031234', '13800138012', 3, '2024-10-03 10:00:00', '普通访客', 45),

('赵访客4', '110101199004041234', '13800138013', 4, '2024-10-04 10:00:00', '普通访客', 50),

('钱访客5', '110101199005051234', '13800138014', 5, '2024-10-05 10:00:00', '普通访客', 60),

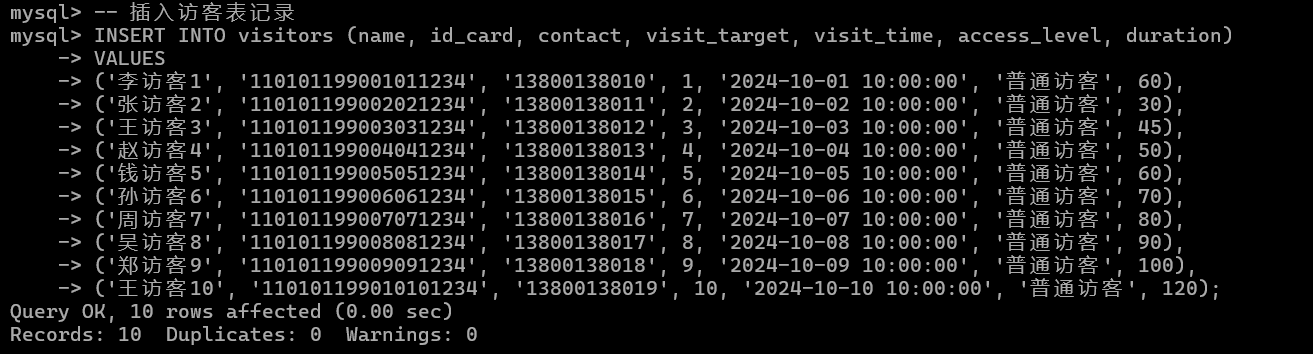
('孙访客6', '110101199006061234', '13800138015', 6, '2024-10-06 10:00:00', '普通访客', 70),

('周访客7', '110101199007071234', '13800138016', 7, '2024-10-07 10:00:00', '普通访客', 80),

('吴访客8', '110101199008081234', '13800138017', 8, '2024-10-08 10:00:00', '普通访客', 90),

('郑访客9', '110101199009091234', '13800138018', 9, '2024-10-09 10:00:00', '普通访客', 100),

('王访客10', '110101199010101234', '13800138019', 10, '2024-10-10 10:00:00', '普通访客', 120);



-- 插入用户表记录

INSERT INTO users (username, password\_hash, role, permissions)

VALUES

('admin', 'hash1', '管理员', '全部权限'),

('staff1', 'hash2', '宿管人员', '分配宿舍'),

('staff2', 'hash3', '宿管人员', '管理费用'),

('student1', 'hash4', '学生', '查看信息'),

('student2', 'hash5', '学生', '查看信息'),

('student3', 'hash6', '学生', '查看信息'),

('student4', 'hash7', '学生', '查看信息'),

('student5', 'hash8', '学生', '查看信息'),

('student6', 'hash9', '学生', '查看信息'),

('student7', 'hash10', '学生', '查看信息');



-- 插入日志表记录

INSERT INTO logs (user\_id, action, action\_time, result)

VALUES

(1, '添加宿舍', '2024-10-01 10:00:00', '成功'),

(2, '分配宿舍', '2024-10-02 10:00:00', '成功'),

(3, '修改费用', '2024-10-03 10:00:00', '失败'),

(4, '查看信息', '2024-10-04 10:00:00', '成功'),

(5, '缴纳费用', '2024-10-05 10:00:00', '成功'),

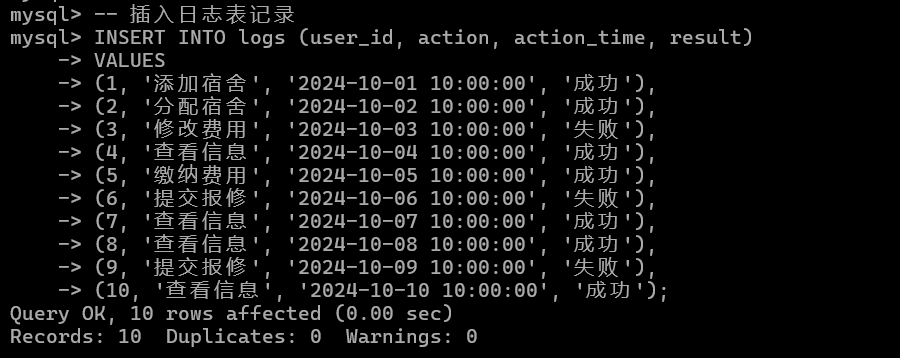
(6, '提交报修', '2024-10-06 10:00:00', '失败'),

(7, '查看信息', '2024-10-07 10:00:00', '成功'),

(8, '查看信息', '2024-10-08 10:00:00', '成功'),

(9, '提交报修', '2024-10-09 10:00:00', '失败'),

(10, '查看信息', '2024-10-10 10:00:00', '成功');



### 4.1.3 修改记录

表一

代码：

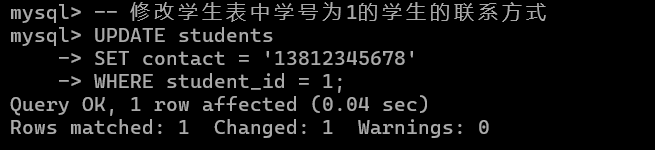
-- 修改学生表中学号为1的学生的联系方式

UPDATE students

SET contact = '13812345678'

WHERE student\_id = 1;

结果截图：



## 4.2 删除记录

删除一个表中的一条记录

代码：

DELETE FROM logs WHERE user\_id = 1;

结果截图：



## 4.3查询问题及查询结果

### 4.3.1 选择列

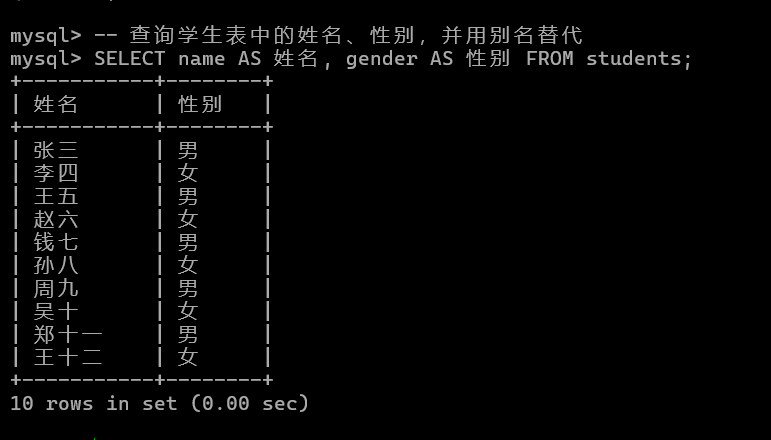
自行设计查询单表中的几列，并用别名代替。

代码：

-- 查询学生表中的姓名、性别，并用别名替代

SELECT name AS 姓名, gender AS 性别 FROM students;

结果截图：



### 4.3.2函数使用

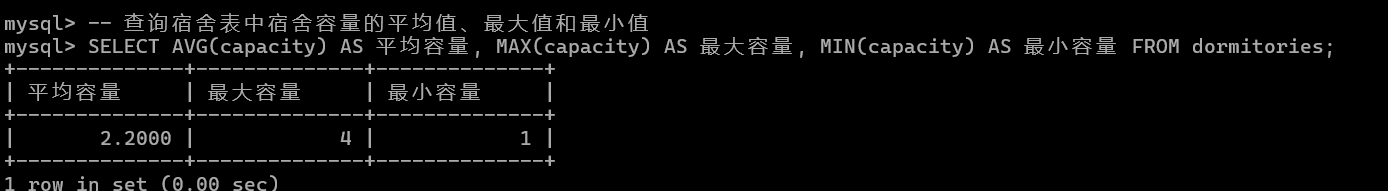
计算满足条件的平均值，最大值，最小值。

代码：

-- 查询宿舍表中宿舍容量的平均值、最大值和最小值

SELECT AVG(capacity) AS 平均容量, MAX(capacity) AS 最大容量, MIN(capacity) AS 最小容量 FROM dormitories;

结果截图：



### 4.3.3 LIKE子句

用LIKE子句实现模糊查询

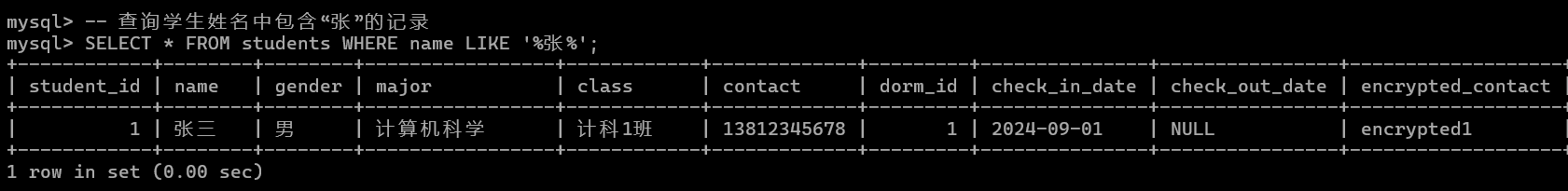
说明实现的功能：

代码：

-- 查询学生姓名中包含“张”的记录

SELECT \* FROM students WHERE name LIKE '%张%';

结果截图：



### 4.3.4 条件查询

用给定条件，查询结果

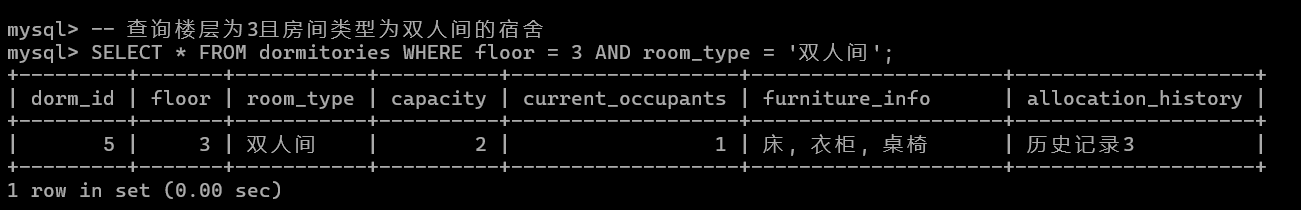
说明实现的功能：

代码：

-- 查询楼层为3且房间类型为双人间的宿舍

SELECT \* FROM dormitories WHERE floor = 3 AND room\_type = '双人间';

结果截图：



### 4.4.5多表查询

至少牵扯两个表，根据给定条件，查询结果

说明实现的功能：

代码：

-- 连接宿舍表和学生表，查询学生的姓名及其宿舍信息

SELECT students.name AS 姓名, dormitories.floor AS 楼层, dormitories.room\_type AS 房间类型

FROM students

JOIN dormitories ON students.dorm\_id = dormitories.dorm\_id;

结果截图：



### 4.4.6子查询

单行子查询举例

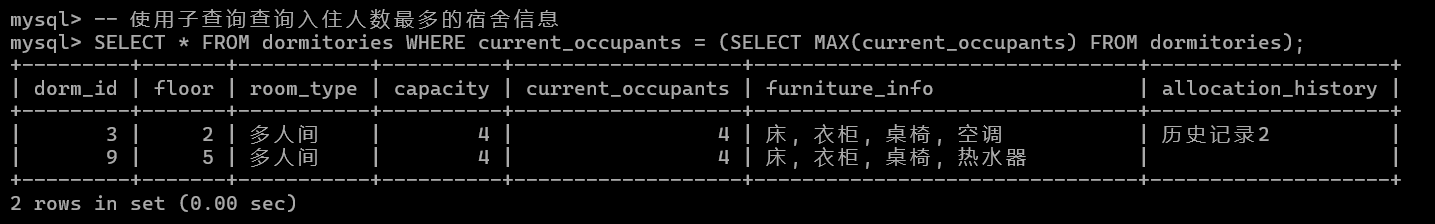
说明实现的功能：

代码：

-- 使用子查询查询入住人数最多的宿舍信息

SELECT \* FROM dormitories WHERE current\_occupants = (SELECT MAX(current\_occupants) FROM dormitories);

结果截图：



多行子查询举例（用in，any,all任意一个均可）

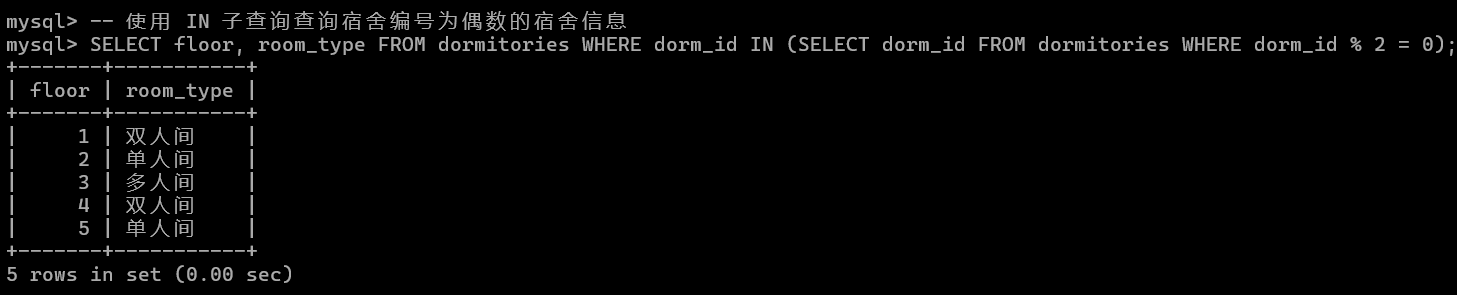
说明实现的功能：

代码：

-- 使用 IN 子查询查询宿舍编号为偶数的宿舍信息

SELECT floor, room\_type FROM dormitories WHERE dorm\_id IN (SELECT dorm\_id FROM dormitories WHERE dorm\_id % 2 = 0);

结果截图：



## 4.5建立视图

建立一个用户user01，给user01账户授予建立视图的权限。

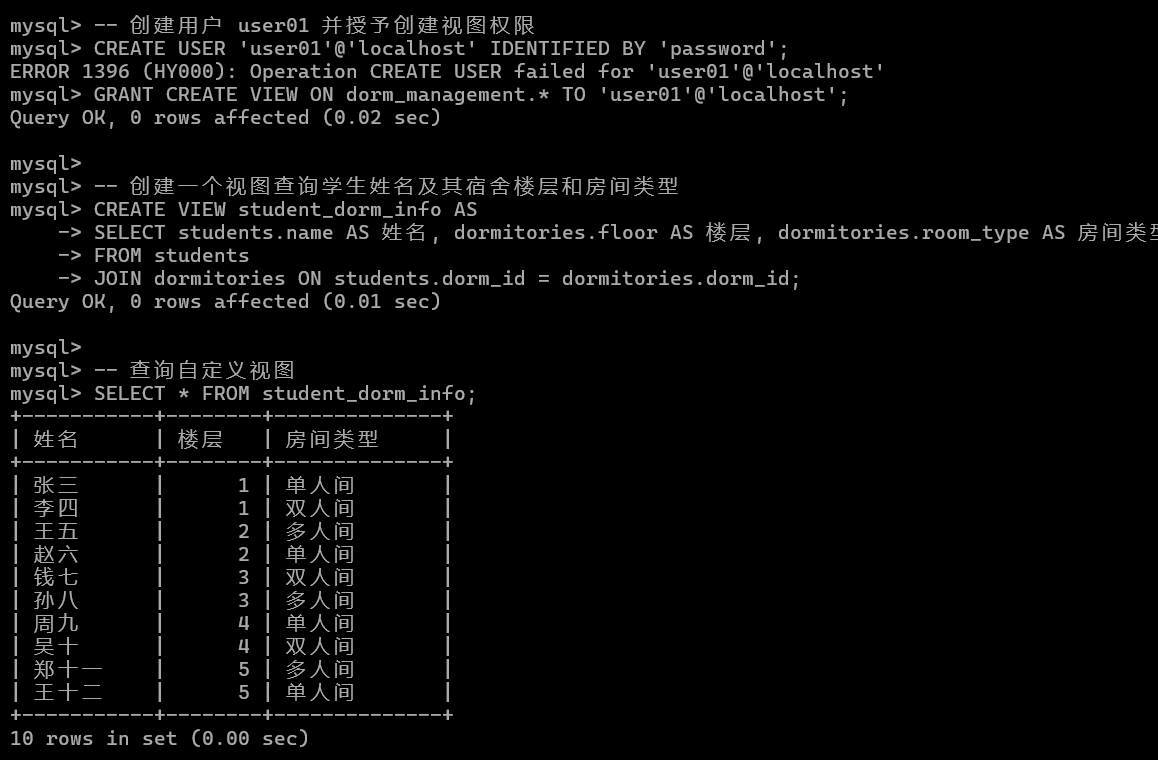
代码：

-- 创建用户 user01 并授予创建视图权限

CREATE USER 'user01'@'localhost' IDENTIFIED BY 'password';

GRANT CREATE VIEW ON dorm\_management.\* TO 'user01'@'localhost';

结果截图：



### 4.5.1 定义视图

自定义一个视图，要求实现多表查询。

-- 创建一个视图查询学生姓名及其宿舍楼层和房间类型

CREATE VIEW student\_dorm\_info AS

SELECT students.name AS 姓名, dormitories.floor AS 楼层, dormitories.room\_type AS 房间类型

FROM students

JOIN dormitories ON students.dorm\_id = dormitories.dorm\_id;

-- 查询自定义视图

SELECT \* FROM student\_dorm\_info;

结果截图：



### 4.5.2利用视图插入数据

创建简单视图代码：

-- 创建一个简单视图插入学生和宿舍编号信息

CREATE VIEW student\_basic\_info AS

SELECT student\_id, name, dorm\_id FROM students;

-- 利用视图插入一条学生记录

INSERT INTO students (name, gender, dorm\_id) VALUES ('刘十三', '男', 5);

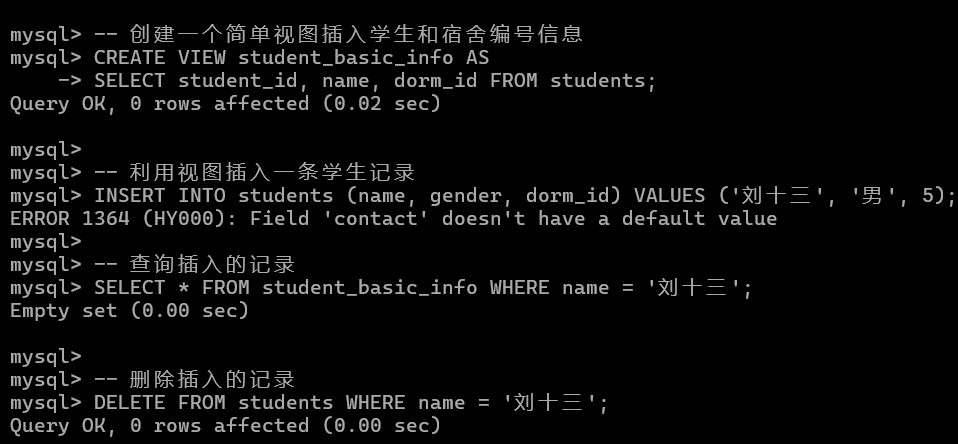
-- 查询插入的记录

SELECT \* FROM student\_basic\_info WHERE name = '刘十三';

-- 删除插入的记录

DELETE FROM students WHERE name = '刘十三';

删除结果截图：



# 5、建立存储过程及函数

## 5.1创建存储过程

创建一个存储过程，该存储过程实现输入某一个参数可以返回对应行上另外一个参数的值（如给定学号，返回选修课程信息等）。

代码：

-- 创建存储过程，根据学号查询学生姓名

DELIMITER $$

CREATE PROCEDURE GetStudentName(IN stu\_id INT, OUT stu\_name VARCHAR(50))

BEGIN

SELECT name INTO stu\_name FROM students WHERE student\_id = stu\_id;

END $$

DELIMITER ;

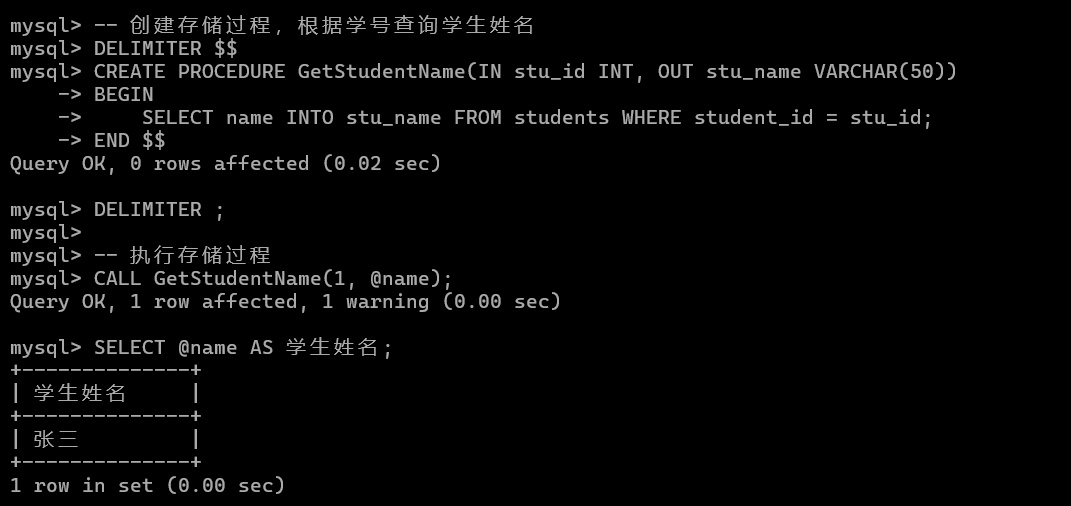
-- 执行存储过程

CALL GetStudentName(1, @name);

SELECT @name AS 学生姓名;

执行存储过程代码：

结果截图：



## 5.2存储过程应用

创建一个函数过程，实现统计功能（统计个数或平均值等）。

代码：

-- 创建函数，统计宿舍表中当前入住人数的平均值

DELIMITER $$

CREATE FUNCTION AvgOccupants() RETURNS DECIMAL(10,2)

BEGIN

DECLARE avg\_value DECIMAL(10,2);

SELECT AVG(current\_occupants) INTO avg\_value FROM dormitories;

RETURN avg\_value;

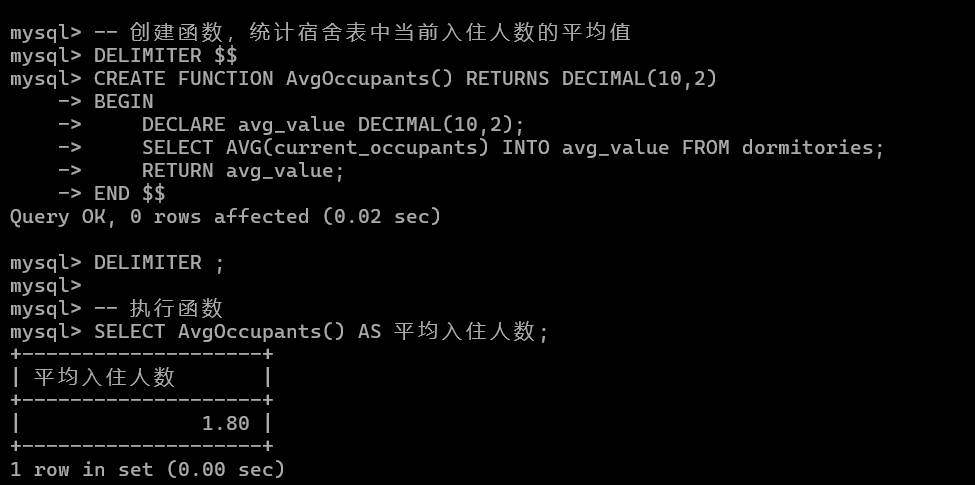
END $$

DELIMITER ;

-- 执行函数

SELECT AvgOccupants() AS 平均入住人数;

结果截图：



# 6、建立触发器

在数据库的XXXX表中，定义一个触发器，当一个记录被删除时，把XXXX的主键添加到delXXXX（此表可单独创建）表中。

-- 创建一个空表，用于记录被删除的宿舍编号

CREATE TABLE delDormitories (

deleted\_dorm\_id INT COMMENT '被删除的宿舍编号'

);

-- 创建触发器，当宿舍表有记录被删除时，将主键插入到 delDormitories 表中

DELIMITER $$

CREATE TRIGGER after\_dorm\_delete

AFTER DELETE ON dormitories

FOR EACH ROW

BEGIN

INSERT INTO delDormitories (deleted\_dorm\_id) VALUES (OLD.dorm\_id);

END $$

DELIMITER ;

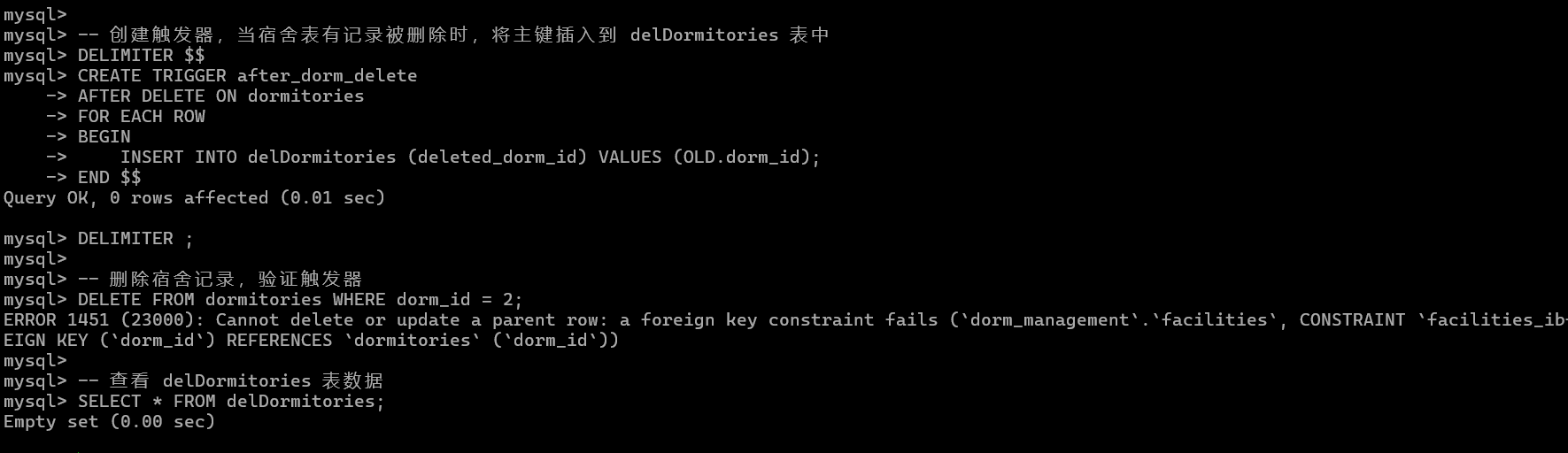
-- 删除宿舍记录，验证触发器

DELETE FROM dormitories WHERE dorm\_id = 2;

-- 查看 delDormitories 表数据

SELECT \* FROM delDormitories;

结果截图：



# 7、项目设计总结

**1. 经验教训**

在本次宿舍管理系统的设计和开发过程中，遇到了以下问题和挑战：

1. **数据库表设计问题**
   * **问题**：初期在设计表结构时，字段的定义和外键关系不够完善，导致后续数据插入和查询时出现数据完整性问题。
   * **解决方法**：通过反复思考系统的业务逻辑，使用ER图分析各实体的关系，重新定义主键和外键关系，确保表之间的联系清晰且符合逻辑。
2. **视图和触发器的实际应用**
   * **问题**：初次接触视图和触发器时，未能理解其作用，设计不合理，导致数据冗余或不一致。
   * **解决方法**：通过参考文档和实践，明确视图用于简化复杂查询，触发器用于自动执行业务逻辑，最终实现合理的设计。
3. **存储过程和函数的调试**
   * **问题**：编写存储过程和函数时，常出现语法错误或未预料到的结果，尤其是在处理输入参数和返回值时。
   * **解决方法**：逐步拆解存储过程的逻辑，单独测试每一部分的功能，同时借助调试工具和日志功能来排查问题。

**2. 收获**

1. **SQL语言的全面掌握**
   * 在本次项目中，深度学习了SQL的高级功能，如视图、存储过程、触发器以及多表查询的复杂用法。这些技术的应用不仅提升了项目的功能完整性，也扩展了个人的数据库知识。
2. **数据库设计能力的提升**
   * 通过分析需求、设计表结构、定义主键和外键，学会了如何以业务逻辑为中心进行数据库设计，并确保数据完整性和一致性。这对实际项目的数据库规划具有重要意义。
3. **问题解决能力的提升**
   * 在开发过程中遇到的问题，如查询逻辑不清晰、触发器误触发等，锻炼了分析问题、查找资料、解决问题的能力。特别是解决复杂问题的能力得到了显著提高。