

中山大學



报告名称：数据库大作业

课程名字：数据库原理

授课老师：张亮

姓 名：宋佳，石镇岩，孙以恒

学 号：24347084, 24347082, 24347086

学 院：网络空间安全

专 业：网络空间安全

日 期：2026. 1. 14

一、实验目的和要求

1. 数据库大作业-医院门诊管理数据库系统

为某个具体部门或单位（如：图书馆、医院、电商平台、教务处等）开发一个完整的数据库应用系统。要求和参考数据库具有相似的功能要求。

二、实验内容与方法

2. 通过 mysql 代码实现数据库的插入，更新，查询，统计。

三、实验步骤与过程

3.1 测试用例：

```
INSERT INTO Department (department_name, department_desc) VALUES
('内科', '常见内科疾病诊疗'),
('外科', '外科手术及创伤处理'),
('儿科', '儿童疾病诊疗'),
('妇产科', '妇科及产科相关诊疗'),
('骨科', '骨骼与关节疾病'),
('皮肤科', '皮肤相关疾病'),
('眼科', '眼部疾病诊疗'),
('耳鼻喉科', '耳鼻喉相关疾病'),
('口腔科', '牙齿与口腔疾病'),
('中医科', '中医诊疗服务');
```

```
INSERT INTO Doctor (name, gender, title, department_id, contact) VALUES
('张建国', '男', '主任医师', 1, '13800000001'),
('李晓梅', '女', '副主任医师', 2, '13800000002'),
('王海', '男', '主治医师', 3, '13800000003'),
('赵敏', '女', '主治医师', 4, '13800000004'),
('陈强', '男', '副主任医师', 5, '13800000005'),
('刘芳', '女', '主任医师', 6, '13800000006'),
('孙伟', '男', '主治医师', 7, '13800000007'),
('周婷', '女', '主治医师', 8, '13800000008'),
('吴磊', '男', '副主任医师', 9, '13800000009'),
('何静', '女', '主任医师', 10, '13800000010');
```

```
UPDATE Department SET director_id = 1 WHERE department_id = 1;
UPDATE Department SET director_id = 2 WHERE department_id = 2;
UPDATE Department SET director_id = 3 WHERE department_id = 3;
UPDATE Department SET director_id = 4 WHERE department_id = 4;
UPDATE Department SET director_id = 5 WHERE department_id = 5;
```

```

UPDATE Department SET director_id = 6 WHERE department_id = 6;
UPDATE Department SET director_id = 7 WHERE department_id = 7;
UPDATE Department SET director_id = 8 WHERE department_id = 8;
UPDATE Department SET director_id = 9 WHERE department_id = 9;
UPDATE Department SET director_id = 10 WHERE department_id = 10;

INSERT INTO ConsultingRoom (department_id, room_number, max_capacity) VALUES
(1, '101', 30),
(2, '201', 25),
(3, '301', 20),
(4, '401', 20),
(5, '501', 25),
(6, '601', 20),
(7, '701', 15),
(8, '801', 15),
(9, '901', 20),
(10, '1001', 20);

INSERT INTO Patient (name, gender, id_card, phone, insurance_type) VALUES
('王小明', '男', '440101199001010001', '13900000001', '医保'),
('李红', '女', '440101199002020002', '13900000002', '自费'),
('张强', '男', '440101199003030003', '13900000003', '医保'),
('刘洋', '男', '440101199004040004', '13900000004', '自费'),
('陈丽', '女', '440101199005050005', '13900000005', '医保'),
('赵磊', '男', '440101199006060006', '13900000006', '医保'),
('孙娜', '女', '440101199007070007', '13900000007', '自费'),
('周军', '男', '440101199008080008', '13900000008', '医保'),
('吴霞', '女', '440101199009090009', '13900000009', '自费'),
('何俊', '男', '440101199010100010', '13900000010', '医保');

INSERT INTO Cashier (name, department_id, contact, work_status) VALUES
('收费员 A', 1, '13700000001', '在岗'),
('收费员 B', 2, '13700000002', '在岗'),
('收费员 C', 3, '13700000003', '在岗'),
('收费员 D', 4, '13700000004', '在岗'),
('收费员 E', 5, '13700000005', '在岗'),
('收费员 F', 6, '13700000006', '休假'),
('收费员 G', 7, '13700000007', '在岗'),
('收费员 H', 8, '13700000008', '在岗'),
('收费员 I', 9, '13700000009', '在岗'),
('收费员 J', 10, '13700000010', '在岗');

INSERT INTO Reservation
(patient_id, department_id, reservation_date, expected_time) VALUES

```

```

(1, 1, '2026-01-16', '09:00'),
(2, 2, '2026-01-16', '09:30'),
(3, 3, '2026-01-16', '10:00'),
(4, 4, '2026-01-16', '10:30'),
(5, 5, '2026-01-16', '11:00'),
(6, 6, '2026-01-16', '11:30'),
(7, 7, '2026-01-16', '14:00'),
(8, 8, '2026-01-16', '14:30'),
(9, 9, '2026-01-16', '15:00'),
(10, 10, '2026-01-16', '15:30');

```

```

INSERT INTO MedicalRecord
(patient_id, doctor_id, room_id, visit_date, visit_status, symptom, diagnosis) VALUES
(1, 1, 1, NOW(), '已完成', '发热、咳嗽', '上呼吸道感染'),
(2, 2, 2, NOW(), '已完成', '腹痛', '急性阑尾炎'),
(3, 3, 3, NOW(), '就诊中', '咳嗽', '支气管炎'),
(4, 4, 4, NOW(), '已离院', '头晕', '贫血'),
(5, 5, 5, NOW(), '已完成', '关节疼痛', '关节炎'),
(6, 6, 6, NOW(), '就诊中', '皮疹', '过敏性皮炎'),
(7, 7, 7, NOW(), '已完成', '视力模糊', '近视'),
(8, 8, 8, NOW(), '已完成', '耳鸣', '中耳炎'),
(9, 9, 9, NOW(), '已完成', '牙痛', '龋齿'),
(10, 10, 10, NOW(), '已完成', '失眠', '神经衰弱');

```

```

INSERT INTO Payment
(record_id, total_amount, insurance_amount, self_amount, payment_method, payment_date, cashier_id) VALUES
(1, 100.00, 60.00, 40.00, '医保', NOW(), 1),
(2, 300.00, 0.00, 300.00, '现金', NOW(), 2),
(3, 120.00, 80.00, 40.00, '医保', NOW(), 3),
(4, 150.00, 90.00, 60.00, '医保', NOW(), 4),
(5, 200.00, 0.00, 200.00, '微信', NOW(), 5),
(6, 180.00, 100.00, 80.00, '医保', NOW(), 6),
(7, 90.00, 0.00, 90.00, '支付宝', NOW(), 7),
(8, 110.00, 60.00, 50.00, '医保', NOW(), 8),
(9, 70.00, 0.00, 70.00, '现金', NOW(), 9),
(10, 130.00, 80.00, 50.00, '医保', NOW(), 10);

```

```

INSERT INTO Schedule
(doctor_id, room_id, schedule_date, start_time, end_time, max_reservations, reserved_count) VALUES
(1, 1, '2026-01-16', '08:00', '12:00', 20, 5),
(2, 2, '2026-01-16', '08:00', '12:00', 20, 8),
(3, 3, '2026-01-16', '08:00', '12:00', 15, 6),
(4, 4, '2026-01-16', '08:00', '12:00', 15, 4),
(5, 5, '2026-01-16', '14:00', '18:00', 20, 10),

```

```
(6, 6, '2026-01-16', '14:00', '18:00', 15, 7),
(7, 7, '2026-01-16', '14:00', '18:00', 10, 3),
(8, 8, '2026-01-16', '14:00', '18:00', 10, 4),
(9, 9, '2026-01-16', '08:00', '12:00', 15, 9),
(10, 10, '2026-01-16', '08:00', '12:00', 15, 6);
```

3.2 基本功能：

-- 数据库与表结构的展示

```
SHOW DATABASES;
USE My_Hospital;
SHOW TABLES;
DESCRIBE Patient;
DESCRIBE Doctor;
DESCRIBE MedicalRecord;
```

-- 基础数据展示

```
SELECT * FROM Department;
SELECT * FROM Doctor;
SELECT * FROM ConsultingRoom;
```

-- 患者预约

```
INSERT INTO Reservation
(patient_id, department_id, reservation_date, expected_time)
VALUES
(1, 1, '2026-01-18', '09:30');

SELECT
r.reservation_id,
p.name AS 患者,
d.department_name AS 科室,
r.reservation_date,
r.expected_time,
r.status
FROM Reservation r
JOIN Patient p ON r.patient_id = p.patient_id
JOIN Department d ON r.department_id = d.department_id
ORDER BY r.reservation_id DESC;
```

-- 到院登记

```
INSERT INTO MedicalRecord
(patient_id, doctor_id, room_id, visit_date, visit_status, symptom)
VALUES
(1, 1, 1, NOW(), '就诊中', '发热、头痛');
```

```

UPDATE Reservation
SET status = '已就诊'
ORDER BY reservation_id DESC
LIMIT 1;

SELECT
m.record_id,
p.name AS 患者,
doc.name AS 医生,
cr.room_number AS 诊室,
m.visit_status
FROM MedicalRecord m
JOIN Patient p ON m.patient_id = p.patient_id
JOIN Doctor doc ON m.doctor_id = doc.doctor_id
JOIN ConsultingRoom cr ON m.room_id = cr.room_id
ORDER BY m.record_id DESC;

```

-- 缴费结算

```

INSERT INTO Payment
(record_id, total_amount, insurance_amount, self_amount, payment_method, payment_date, cashier_id)
VALUES
(
    (SELECT record_id FROM MedicalRecord ORDER BY record_id DESC LIMIT 1),
    150.00,
    100.00,
    50.00,
    '医保',
    NOW(),
    1
);
UPDATE MedicalRecord
SET visit_status = '已出院'
ORDER BY record_id DESC
LIMIT 1;

SELECT
p.name AS 患者,
pay.total_amount,
pay.insurance_amount,
pay.self_amount,
pay.payment_method
FROM Payment pay
JOIN MedicalRecord m ON pay.record_id = m.record_id
JOIN Patient p ON m.patient_id = p.patient_id
ORDER BY pay.payment_id DESC;

```

-- 管理统计功能

```
SELECT
    d.department_name AS 科室,
    SUM(p.total_amount) AS 总收入
FROM Payment p
JOIN MedicalRecord m ON p.record_id = m.record_id
JOIN Doctor doc ON m.doctor_id = doc.doctor_id
JOIN Department d ON doc.department_id = d.department_id
GROUP BY d.department_name;
SELECT COUNT(*) AS 已完成就诊人次
FROM MedicalRecord
WHERE visit_status = '已离院';
```

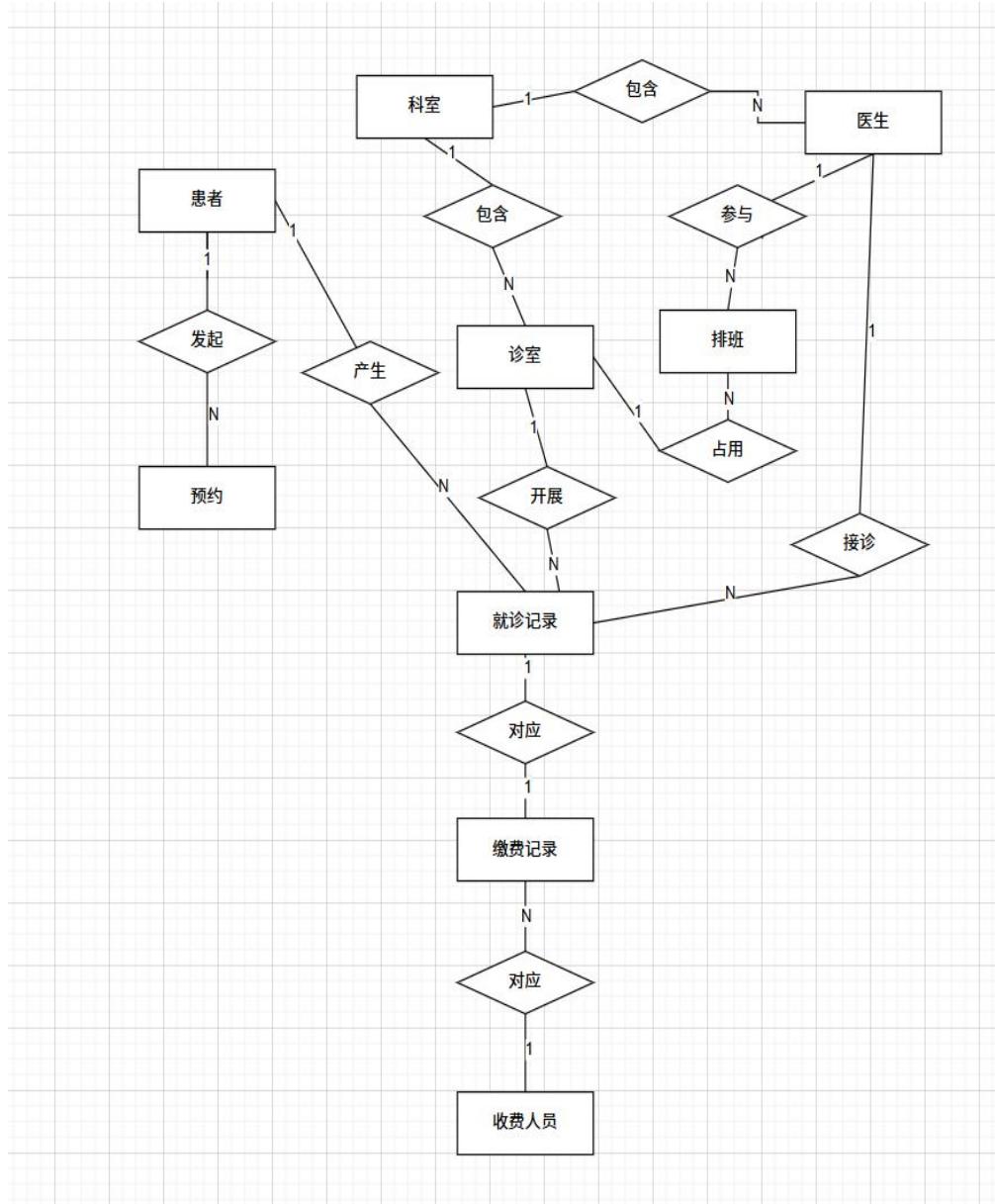
-- 切换用户 hospital_admin

```
SHOW TABLES;
SELECT
    d.department_name,
    SUM(p.total_amount) AS total_income
FROM Payment p
JOIN MedicalRecord m ON p.record_id = m.record_id
JOIN Doctor doc ON m.doctor_id = doc.doctor_id
JOIN Department d ON doc.department_id = d.department_id
GROUP BY d.department_name;
```

-- 切换用户 hospital_cashier

```
SELECT * FROM Payment;
INSERT INTO Payment
(record_id, total_amount, insurance_amount, self_amount, payment_method, payment_date, cashier_id)
VALUES
(1, 200.00, 120.00, 80.00, '医保', NOW(), 1);
SELECT * FROM Patient;
```

3. 3, E-R 图:



设计：业务流程闭环：从“患者预约→到院就诊→缴费离院”的全流程，通过实体关系完整覆盖。

实验结果：

1. 预约流程与到院就诊：

```
mysql my_hospital 运行 停止 专家解释
1 --插入数据
2 INSERT INTO MedicalRecord
3 (patient_id, doctor_id, room_id, visit_date, visit_status, symptom)
4 VALUES (1, 1, 1, NOW(), '就诊中', '发热、头痛');
5
6 UPDATE Reservation
7 SET status = '已预约'
8 WHERE reservation_id = Reservation_Id DESC
9
10 LIMIT 1;
11
12
13 SELECT
14 * FROM record;
15 p.name AS 病人名,
16 d.name AS 医生名,
17 r.name AS 诊室,
18 cr.room_number AS 房号,
19 m.visit_status
20
21 JOIN Patient p ON record.patient_id = p.patient_id
22 JOIN Doctor d ON record.doctor_id = d.doc_doctor_id
23 JOIN ConsultingRoom cr ON m.room_id = cr.room_id
24 ORDER BY m.record_id DESC;
```

缴费结算:

```
mysql -u my_user -p my_hospital > 运行 < 停止 < 解释
1 -- 插入数据
2 INSERT INTO Payment
3 (record_id, total_amount, insurance_amount, self_amount, payment_method, payment_date, cashier_id)
4 VALUES
5 (1, 130.00, 80.00, 50.00, '医保', '2023-10-01', 1),
6 (2, 70.00, 0.00, 70.00, '现金', '2023-10-01', 2),
7 (3, 110.00, 60.00, 50.00, '医保', '2023-10-01', 1),
8 (4, 90.00, 0.00, 90.00, '支付宝', '2023-10-01', 3),
9 (5, 180.00, 100.00, 80.00, '医保', '2023-10-01', 1),
10 (6, 200.00, 0.00, 200.00, '微信', '2023-10-01', 4),
11 (7, 150.00, 90.00, 60.00, '医保', '2023-10-01', 1),
12 (8, 120.00, 80.00, 40.00, '医保', '2023-10-01', 1),
13 (9, 300.00, 0.00, 300.00, '现金', '2023-10-01', 5),
14 );
15
16 UPDATE MedicalRecord
17 SET visit_status='已离院'
18 ORDER BY record_id DESC
19 LIMIT 1;
20
21 SELECT
22     patient AS '患者',
23     pay.total_amount,
24     pay.insurance_amount,
25     pay.self_amount,
26     pay.payment_method
27     FROM Payment pay
28     JOIN MedicalRecord m ON pay.record_id = m.record_id
29     JOIN Patient p ON p.patient_id = m.patient_id
30     ORDER BY pay.payment_id DESC;
31 
```

管理统计:

```
mysql my_hospital 运行 停止 解释
1 -- 管理统计功能
2 SELECT
3 d.department_name AS 科室,
4 SUM(p.total_amount) AS 总收入
5 FROM Payment p
6 JOIN MedicalRecord m ON p.record_id = m.record_id
7 JOIN Doctor doc ON m.doctor_id = doc.doctor_id
8 JOIN Department d ON doc.department_id = d.department_id
9 GROUP BY d.department_name;
10
11 SELECT COUNT(*) AS 已完成就诊人次
12 FROM MedicalRecord
13 WHERE visit_status = '已离院';
消息 摘要 结果 1 结果 2
数据 信息
科室 总收入
内科 1800.00
外科 300.00
儿科 120.00
妇产科 150.00
骨科 200.00
皮肤科 180.00
眼科 90.00
耳鼻喉科 110.00
口腔科 70.00
中医科 130.00
单元格编辑器 数据分析 导出
- 管理统计功能 SELECT d.department_name AS 科室, SUM(p.total_amount) AS 总收入 FROM Payment p JOIN Medic... 只读 第 10 条记录 (共 10 条) In 9, Col 28
```

```

mysql -u my_hospital -p
1 SELECT
2   d.department_name AS 科室,
3   SUM(p.total_amount) AS 总收入
4 FROM Payment p
5 JOIN MedicalRecord m ON p.record_id = m.record_id
6 JOIN Doctor doc ON m.doctor_id = doc.doctor_id
7 JOIN Department d ON doc.department_id = d.department_id
8 GROUP BY d.department_name;
9
10 SELECT COUNT(*) AS 已完成就诊人次
11 FROM MedicalRecord
12 WHERE visit_status = '已出院';
13

```

消息 摘要 结果 1 结果 2
数据 信息
已完成就诊人次 7

+ - ✎ ×
SELECT COUNT(*) AS 已完成就诊人次 FROM MedicalRecord WHERE visit_status = '已出院'
只读 第1条记录 (共1条) Ln 11, Col 19 运行

管理员权限：

```

hospital_admin -> my_hospital
1 -- 切换用户hospital_admin
2 SHOW TABLES;
3
4 d.department_name,
5 SUM(p.total_amount) AS total_income
6 FROM Payment p
7 JOIN MedicalRecord m ON p.record_id = m.record_id
8 JOIN Doctor doc ON m.doctor_id = doc.doctor_id
9 JOIN Department d ON doc.department_id = d.department_id
10 GROUP BY d.department_name;

Tables_in_my_hospital
administrator
cashier
consultingroom
department
doctor
medicalrecord
patient
payment
reservation
schedule

+ - ✎ ×  
- 切换用户hospital_admin SHOW TABLES  
只读 第3条记录 (共10条) Ln 10, Col 28 运行

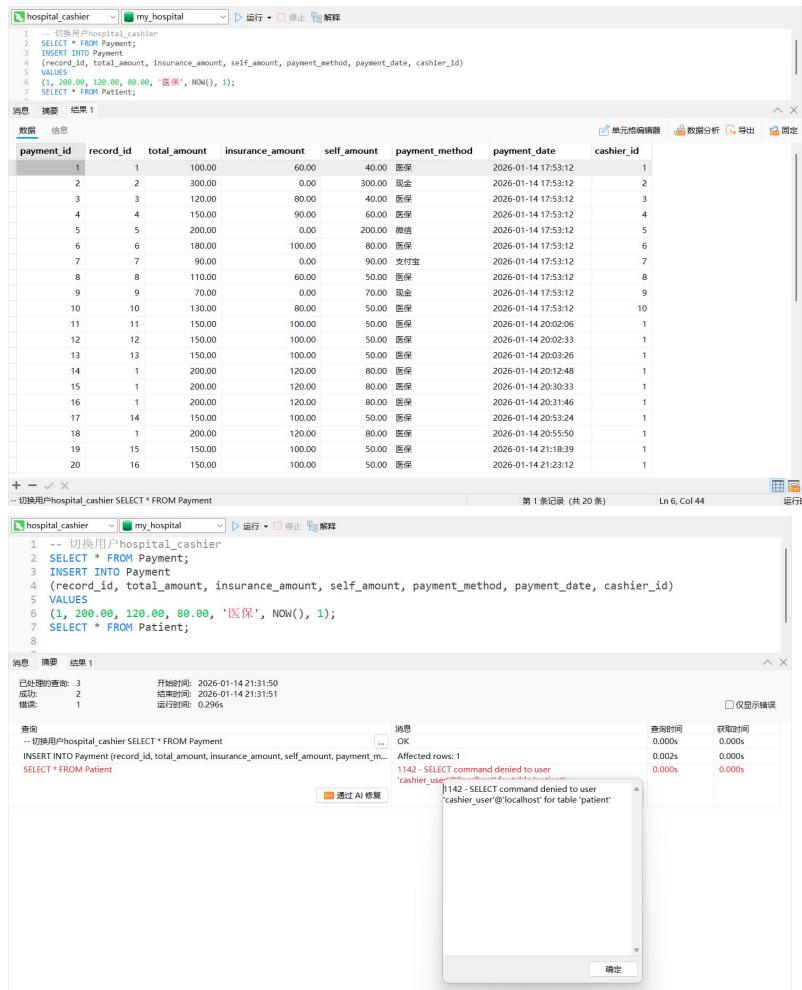
mysql -u hospital_admin -p
1 -- 切换用户hospital_admin
2 SHOW TABLES;
3
4 d.department_name,
5 SUM(p.total_amount) AS total_income
6 FROM Payment p
7 JOIN MedicalRecord m ON p.record_id = m.record_id
8 JOIN Doctor doc ON m.doctor_id = doc.doctor_id
9 JOIN Department d ON doc.department_id = d.department_id
10 GROUP BY d.department_name;

消息 摘要 结果 1 结果 2  
数据 信息  
department_name total_income  
内科 1800.00  
外科 300.00  
儿科 120.00  
妇产科 150.00  
骨科 200.00  
皮肤科 180.00  
眼科 90.00  
耳鼻喉科 110.00  
口腔科 70.00  
中医科 130.00

+ - ✎ ×  
SELECT d.department_name, SUM(p.total_amount) AS total_income FROM Payment p JOIN MedicalRecord m ON | 只读 第1条记录 (共10条) Ln 10, Col 28 运行

```

收费人员权限：



四、实验结论与体会

4. 在系统完成之后，我们围绕医院门诊的实际业务流程，对数据库进行了整体测试。测试主要集中在基础数据是否正确、业务流程是否连贯，以及不同角色访问数据是否符合预期。

首先，在基础信息方面，通过录入患者、医生、科室和诊室等数据，检查各数据表是否能够正常存储和查询，测试结果表明表结构设计合理，数据之间的关联关系清晰，没有出现数据混乱或无法关联的情况。

其次，在业务流程测试中，我们按照“预约—就诊—缴费”的顺序进行验证。测试过程中，相关记录之间能够正确关联，业务状态也能随着流程推进发生变化，整体流程与医院门诊的实际情况基本一致，说明数据库设计能够支撑完整的门诊业务。

在统计功能方面，通过多条就诊和缴费数据进行汇总查询，统计结果能够正确反映门诊人次和收费情况，验证了数据库在数据分析方面的可用性。

最后，通过不同角色对数据库进行访问测试，确认权限控制能够生效，不同角色只能访问与自身职责相关的数据，避免了越权操作的问题。

实验体会：通过本次实验，我们完整经历了一个数据库系统从需求分析到设计、实现和测试的全过程。相比单纯地写几张数据表，本实验更强调将数据库与实际业务结合起来，使数据库真正服务于业务流程。

在实验过程中，我们对如何进行概念建模、如何将概念模型转换为关系模型，以及如何通过主键和外键来维护数据一致性有了更直观的理解。同时，通过将预约、就诊和缴费等环节进行关联，体会到了数据库在真实业务系统中的核心作用。

此外，在数据库层面引入权限控制和统计分析功能，也让我们认识到数据库不仅是“存数据”，还在系统安全和管理决策中发挥重要作用。整体来看，本实验加深了我们对数据库系统设计思想的理解，也提升了将理论知识应用到实际问题中的能力。