



# 資料庫期末專題

## 小型牙醫診所

第13組

盧冠寧 411630097 蔡翔宇411631145  
張蔡大鈞411631178 陳柏安412637141



# 目錄



專題背景與系統功能



ER 圖



正規化



資料表建立與操作語法



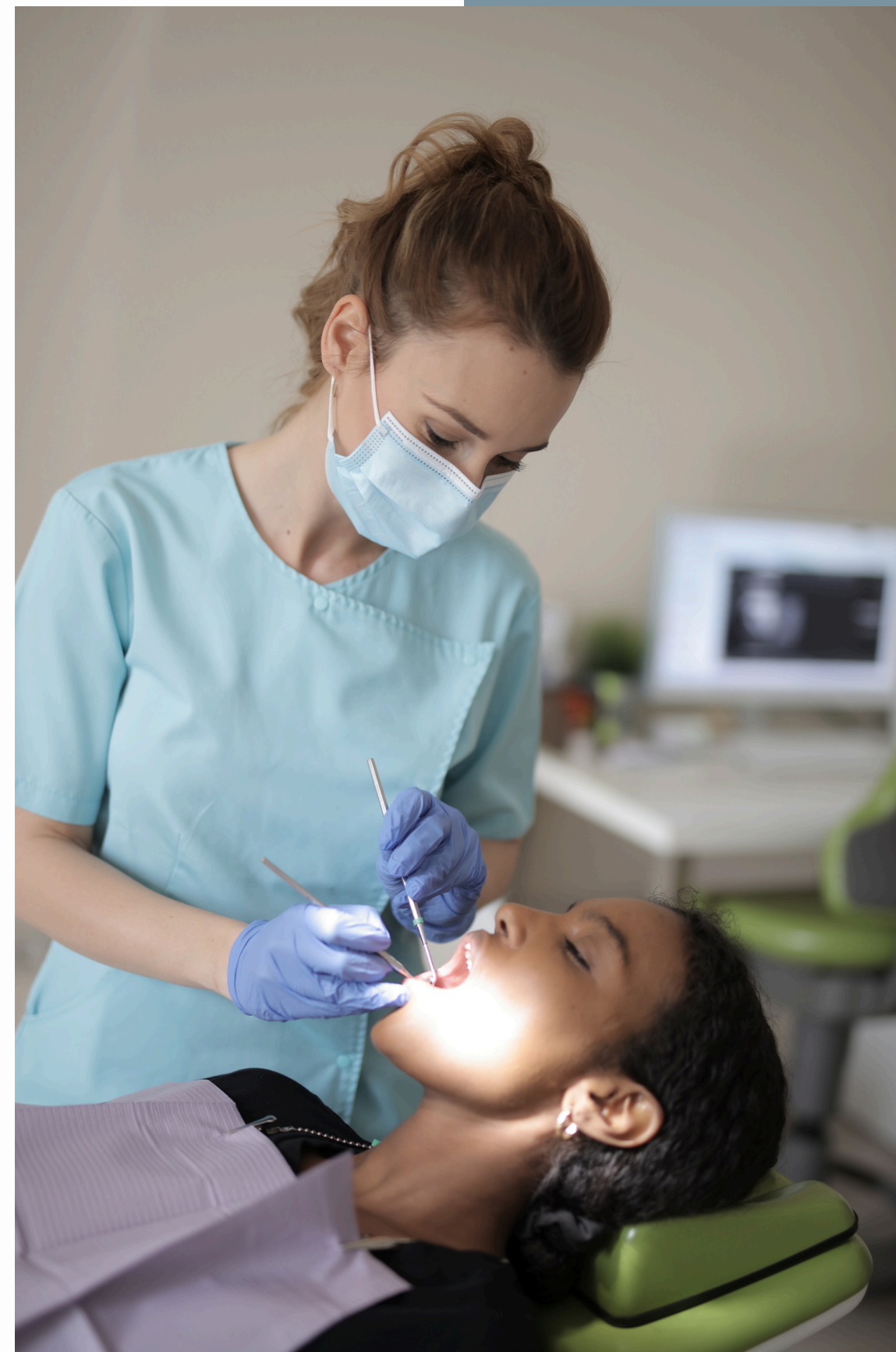
進階 SQL 應用與權限管理



分工表



DEMO



# 專題背景與系統功能

## 背景

我們以小型牙醫診所作為出發點，模擬實際診所在病患管理、排班、掛號、病歷紀錄等流程中會面臨的資料處理需求。

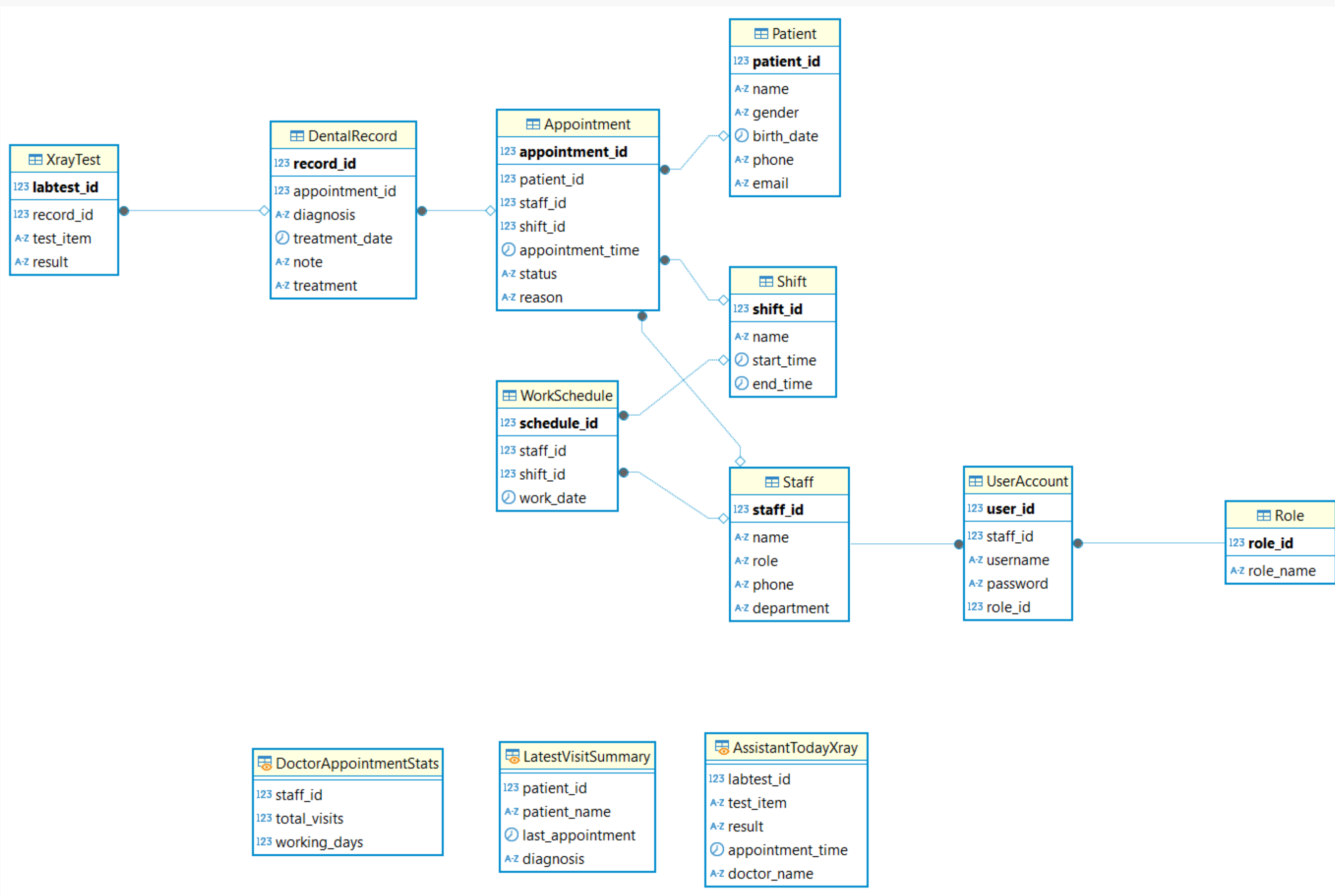
## 系統功能

- 病患管理：基本資料建檔
- 員工管理：分別有牙醫師、牙助、櫃台
- 值班排班：每日記錄輪班狀況（午/晚診）
- 掛號與自動指派醫師
- 病歷與 X 光檢查資料自動新增
- 帳號與權限控制（只能查詢、不能修改）





# ER圖



# 正規化設計

正規化階段	設計說明	SQL範例
1NF	每個欄位只存單一資料，例如 Patient.name、email、gender 都是單一值，沒有逗號合併等情況	<pre>-- 病患資料表 Patient CREATE TABLE Patient (   patient_id INT PRIMARY KEY AUTO_INCREMENT,   name VARCHAR(50) NOT NULL,   gender ENUM('男', '女') NOT NULL,   birth_date DATE,   phone VARCHAR(20),   email VARCHAR(100) );</pre>
2NF	所有表格主鍵皆為單欄 AUTO_INCREMENT，例如 patient_id、staff_id，避免複合主鍵與部分依賴	<pre>-- 員工資料表 Staff CREATE TABLE Staff (   staff_id INT PRIMARY KEY AUTO_INCREMENT,   name VARCHAR(50) NOT NULL,   role ENUM('牙醫師', '牙助', '櫃台') NOT NULL,   phone VARCHAR(20),   department VARCHAR(50) );</pre>
3NF	將重複資訊（如角色）抽出獨立成 Role 表格，UserAccount 用 role_id 作為外鍵	<pre>-- 建立帳號角色表 Role 與 UserAccount CREATE TABLE Role (   role_id INT AUTO_INCREMENT PRIMARY KEY,   role_name VARCHAR(50) UNIQUE NOT NULL );  CREATE TABLE UserAccount (   user_id INT AUTO_INCREMENT PRIMARY KEY,   staff_id INT NOT NULL,   username VARCHAR(50) UNIQUE NOT NULL,   password VARCHAR(255) NOT NULL,   role_id INT NOT NULL,   FOREIGN KEY (staff_id) REFERENCES Staff(staff_id),   FOREIGN KEY (role_id) REFERENCES Role(role_id) );</pre>

# 資料表建立與操作語法

```
-- 病患資料表 Patient
CREATE TABLE Patient (
  patient_id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(50) NOT NULL,
  gender ENUM('男', '女') NOT NULL,
  birth_date DATE,
  phone VARCHAR(20),
  email VARCHAR(100)
);

-- 員工資料表 Staff
CREATE TABLE Staff (
  staff_id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(50) NOT NULL,
  role ENUM('牙醫師', '牙助', '櫃台') NOT NULL,
  phone VARCHAR(20),
  department VARCHAR(50)
);

-- 診所時段表 Shift
CREATE TABLE Shift (
  shift_id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(20) NOT NULL, -- 午診 / 晚診
  start_time TIME NOT NULL,
  end_time TIME NOT NULL
);

-- 員工出勤紀錄表 WorkSchedule
CREATE TABLE WorkSchedule (
  schedule_id INT PRIMARY KEY AUTO_INCREMENT,
  staff_id INT,
  shift_id INT,
  work_date DATE,
  FOREIGN KEY (staff_id) REFERENCES Staff(staff_id),
  FOREIGN KEY (shift_id) REFERENCES Shift(shift_id)
);
```

## 表格設計語法 (Schema 建立)

我們總共建立了 8 張表格，對應診所裡的不同資料，例如病患、員工、排班、掛號、病歷等等。每張表都有主鍵 (PRIMARY KEY)、非空限制 (NOT NULL)，也有設定像 ENUM、AUTO\_INCREMENT 這些欄位屬性

## 關聯設計與外鍵 (FOREIGN KEY)

在左圖裡，是病患 (Patient)、員工 (Staff)、時段 (Shift) 與出勤 (WorkSchedule) 等表格。其中 WorkSchedule 是負責紀錄哪個員工在哪一天上哪一個班，因此它有兩個外鍵分別連到員工 (staff\_id) 與班別 (shift\_id)，這樣可以清楚知道誰什麼時間值班，也能保證資料一致性與不重複

# 常用操作語法

插入資料 (INSERT INTO): 我們會先把病患、醫師、牙助等資料一一建好, 再加上他們的值班時間, 這樣才能開始安排掛號流程

查詢資料 (SELECT)：透過查詢語法，我們可以找出今天哪位醫師上班、哪位病患已經完成看診，或是查看病歷與檢查資料

```

INSERT INTO Patient (name, gender, birth_date, phone, email) VALUES ('莊郁婷', '女', '1985-11-11', '0957796538',
INSERT INTO Patient (name, gender, birth_date, phone, email) VALUES ('饒志偉', '女', '2005-05-26', '0922204111',
INSERT INTO Patient (name, gender, birth_date, phone, email) VALUES ('黃靜怡', '男', '1968-06-26', '0981257623',
INSERT INTO Patient (name, gender, birth_date, phone, email) VALUES ('何婷婷', '男', '2001-03-06', '0933495822',
INSERT INTO Patient (name, gender, birth_date, phone, email) VALUES ('鞠佳慧', '男', '2002-04-03', '0982291445',

-- 插入 Staff 員工資料
INSERT INTO Staff (name, role, phone, department) VALUES ('李欣怡', '牙醫師', '0949672979', '牙科');
INSERT INTO Staff (name, role, phone, department) VALUES ('祝雅慧', '牙醫師', '0972568035', '牙科');
INSERT INTO Staff (name, role, phone, department) VALUES ('陳淑慧', '牙醫師', '0902502366', '牙科');
INSERT INTO Staff (name, role, phone, department) VALUES ('岑郁雯', '牙醫師', '0931219589', '牙科');
INSERT INTO Staff (name, role, phone, department) VALUES ('張志偉', '牙醫師', '0929035539', '牙科');

INSERT INTO Staff (name, role, phone, department) VALUES ('王雅文', '牙助', '0944039106', '牙科');
INSERT INTO Staff (name, role, phone, department) VALUES ('何龍', '牙助', '0915441906', '牙科');
INSERT INTO Staff (name, role, phone, department) VALUES ('劉依婷', '牙助', '0990252298', '牙科');
INSERT INTO Staff (name, role, phone, department) VALUES ('陳冠廷', '牙助', '0911016498', '牙科');
INSERT INTO Staff (name, role, phone, department) VALUES ('彭怡君', '牙助', '0947609744', '牙科');

INSERT INTO Staff (name, role, phone, department) VALUES ('王宜君', '櫃台', '0971404160', '櫃台');
INSERT INTO Staff (name, role, phone, department) VALUES ('邵龍', '櫃台', '0907812152', '櫃台');
SELECT * FROM Shift;

-- 插入 Shift 輪班資料
INSERT INTO Shift (shift_id, name, start_time, end_time) VALUES
(1, '午診', '10:00:00', '15:00:00'),
(2, '晚診', '17:00:00', '22:00:00');

-- 插入 WorkSchedule 員工出勤排班
INSERT INTO WorkSchedule (staff_id, shift_id, work_date) VALUES
(1, 1, '2025-06-02'),
(2, 1, '2025-06-02'),

```

- 指定日期每位醫師的看診次數  

```
SELECT S.name AS doctor_name, DATE(A.appointment_time) AS date, COUNT(*) AS  
FROM Appointment A  
JOIN Staff S ON A.staff_id = S.staff_id  
WHERE S.role = '牙醫師' AND DATE(A.appointment_time) = '2025-06-05'  
GROUP BY A.staff_id, DATE(A.appointment_time)  
ORDER BY visits DESC;
```
- 每位病患總共就診幾次（含已完成的）  

```
SELECT P.name AS patient_name, COUNT(*) AS total_visits  
FROM Appointment A  
JOIN Patient P ON A.patient_id = P.patient_id  
WHERE A.status = '完成'  
GROUP BY A.patient_id  
ORDER BY total_visits DESC;
```



# 進階 SQL 應用與權限管理

## View（檢視表）

- 1.查詢醫師每日看診人數平均
- 2.病患最近一次看診紀錄
- 3.過去七天還沒完成看診的名單

```
-- 建立 View：病患最近一次診斷紀錄
CREATE VIEW LatestVisitSummary AS
SELECT
    p.patient_id,
    p.name AS patient_name,
    MAX(a.appointment_time) AS last_appointment,
    d.diagnosis
FROM
    Patient p
JOIN
    Appointment a ON p.patient_id = a.patient_id
LEFT JOIN
    DentalRecord d ON a.appointment_id = d.appointment_id
GROUP BY
    p.patient_id, p.name, d.diagnosis;
```

## Trigger（觸發器）

- 1.當掛號狀態從其他狀態變成「完成」時，自動建立一筆病歷
- 2.減少忘記填寫病歷的風險，保障資料完整性

```
-- 建立 Trigger：當掛號完成時，自動新增病歷資料
DELIMITER $$
CREATE TRIGGER AddDentalRecordAfterAppointmentComplete
AFTER UPDATE ON Appointment
FOR EACH ROW
BEGIN
    IF NEW.status = '完成' AND OLD.status != '完成' THEN
        INSERT INTO DentalRecord (appointment_id, diagnosis, treatment)
        VALUES (NEW.appointment_id, '請補上診斷內容', '請補上治療內容');
    END IF;
END $$
DELIMITER ;
```

## Stored Procedure（儲存程序）

- 1.自動幫某位病患找一位當天值班的醫師
- 2.自動為病患建立掛號資料，系統會自動找當天值班的醫師，減少人工配對的負擔

```
-- 自動指派值班醫師後建立掛號資料
DELIMITER $$

CREATE PROCEDURE CreateAppointmentWithDoctor (
    IN p_patient_id INT,
    IN p_date DATE,
    IN p_time TIME,
    IN p_reason VARCHAR(255)
)
BEGIN
    DECLARE selected_doctor INT;

    -- 找當天值班的第1位醫師（非牙醫師）
    SELECT s.staff_id INTO selected_doctor
    FROM WorkSchedule ws
    JOIN Staff s ON ws.staff_id = s.staff_id
    WHERE ws.work_date = p_date AND s.role = '醫師'
    ORDER BY RAND()
    LIMIT 1;

    -- 插入掛號資料（狀態預設為「預約中」或「未完成」）
    INSERT INTO Appointment (patient_id, staff_id, appointment_time, reason, status)
    VALUES (p_patient_id, selected_doctor, TIMESTAMP(p_date, p_time), p_reason, '預約中');
END$$
DELIMITER ;
```

## 權限與使用者管理

- 1.建立了使用者 assistant，只能查看指定的 X 光檢查資料（View）
- 2.限制他不能操作其他表格，避免誤刪資料或造成安全風險
- 3.使用 GRANT 指令精準控制權限

```
-- 建立資料庫使用者
CREATE USER 'assistant'@'localhost' IDENTIFIED BY '123456';

-- 只開放 View 權限給 assistant
-- 這樣 assistant 只能查詢 View，不能操作其他表格
GRANT SELECT ON dental_clinic.AssistantTodayXray TO 'assistant'@'localhost';

-- 確認 assistant 的權限是否正確：
SHOW GRANTS FOR 'assistant'@'localhost';
```



# 分工表

姓名	負責項目
盧冠寧 411630097	資料表設計與正規化
陳柏安 412637141	進階功能應用
蔡翔宇 411631145	製作簡報、負責口頭報告
張蔡大鈞 411631178	撰寫文件與說明整理

# DBEAVER 操作



*Thank you*

