

這裡是一個適合本科生的 E-R 圖設計，主題為「學生課程管理系統」，並附上詳細的 SQL 語言檔案，以便學生在 phpMyAdmin 上操作。

實驗目標

- 1. 學習 E-R 圖的基本概念，並將其轉換為關聯式資料庫的表格設計。
- 2. 透過 phpMyAdmin 建立資料庫與資料表。
- 3. 使用 SQL 語法建立表格並插入資料。
- 4. 練習 SQL 查詢、插入、更新、刪除等基本操作。

步驟 1：設計 E-R 圖

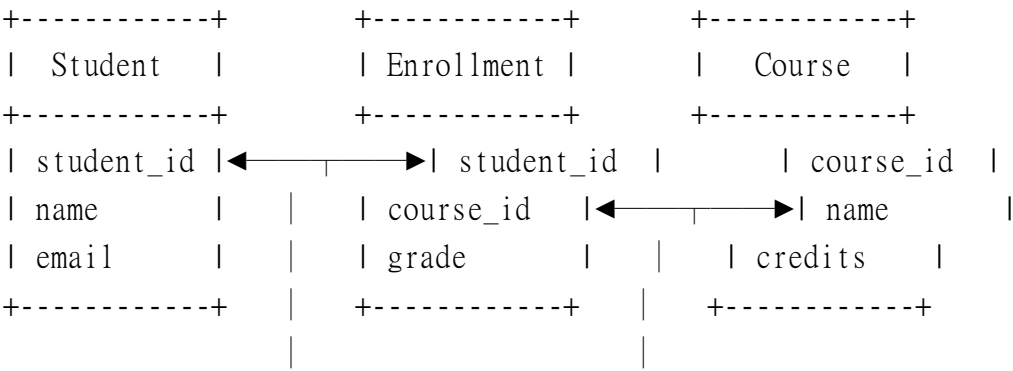
這個 學生課程管理系統 包含以下實體 (Entities)：

- 學生 (Student)：存放學生的基本資料。
- 課程 (Course)：存放課程資訊。
- 選課 (Enrollment)：存放學生選修的課程。

關聯 (Relationships)

- 一名學生 可選修多門課程 (Student → Enrollment → Course)
- 一門課程 可有多名學生選修 (Course → Enrollment → Student)
- Enrollment 作為多對多 (Many-to-Many) 關係的橋接表 (Bridge Table)

E-R 圖 (示意圖)



屬性說明

- **Student**
 - student_id (PK): 學生編號 (主鍵)
 - name: 學生姓名
 - email: 學生電子郵件
 - **Course**
 - course_id (PK): 課程編號 (主鍵)
 - name: 課程名稱
 - credits: 學分
 - **Enrollment**
 - student_id (FK): 學生編號 (外鍵)
 - course_id (FK): 課程編號 (外鍵)
 - grade: 成績 (可選)
-

步驟 2：在 phpMyAdmin 建立資料庫

1. 打開 phpMyAdmin。
 2. 在左側點擊「新建」並輸入資料庫名稱：university_db。
 3. 點擊「建立」。
-

步驟 3：執行 SQL 指令建立資料表

使用以下 SQL 指令，在 phpMyAdmin 的 SQL 查詢區 執行：

```
CREATE DATABASE university_db;
USE university_db;

-- 建立學生表
CREATE TABLE Student (
    student_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
```

```

        email VARCHAR(100) UNIQUE NOT NULL
    );

-- 建立課程表
CREATE TABLE Course (
    course_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    credits INT NOT NULL CHECK (credits > 0)
);

-- 建立選課表（橋接表）
CREATE TABLE Enrollment (
    student_id INT,
    course_id INT,
    grade CHAR(2), -- 例如: A, B+, C, F
    PRIMARY KEY (student_id, course_id),
    FOREIGN KEY (student_id) REFERENCES Student(student_id) ON DELETE
    CASCADE,
    FOREIGN KEY (course_id) REFERENCES Course(course_id) ON DELETE
    CASCADE
);

```

步驟 4：插入範例資料

讓學生在 phpMyAdmin 執行以下 SQL，填充一些測試資料：

```

-- 插入學生資料
INSERT INTO Student (name, email) VALUES
('張三', 'zhangsan@example.com'),
('李四', 'lisi@example.com'),
('王五', 'wangwu@example.com');

-- 插入課程資料
INSERT INTO Course (name, credits) VALUES
('數據庫管理', 3),
('計算機網絡', 4),
('人工智能導論', 3);

```

```
-- 插入選課資料 (學生與課程對應)
INSERT INTO Enrollment (student_id, course_id, grade) VALUES
(1, 1, 'A'),
(1, 2, 'B+'),
(2, 1, 'A-'),
(2, 3, 'B'),
(3, 2, 'C');
```

步驟 5：SQL 查詢與操作

學生可以嘗試以下查詢，檢查數據是否正確：

(1) 查看所有學生與選課資訊

```
SELECT Student.name, Student.email, Course.name AS course_name,
Enrollment.grade
FROM Enrollment
JOIN Student ON Enrollment.student_id = Student.student_id
JOIN Course ON Enrollment.course_id = Course.course_id;
```

(2) 查詢選修「數據庫管理」的學生

```
SELECT Student.name, Student.email
FROM Enrollment
JOIN Student ON Enrollment.student_id = Student.student_id
JOIN Course ON Enrollment.course_id = Course.course_id
WHERE Course.name = '數據庫管理';
```

(3) 更新某學生的成績

```
UPDATE Enrollment
SET grade = 'A+'
WHERE student_id = 3 AND course_id = 2;
```

(4) 刪除一位學生

```
DELETE FROM Student WHERE student_id = 2;
```

(此操作將自動刪除 Enrollment 表中與該學生相關的選課記錄)

步驟 6：學習心得與作業

任務

1. 增加一門新的課程，並讓所有學生選修該課程。
 2. 撰寫一個 SQL 查詢，找出所有學過「人工智能導論」且成績為 A 或 B 的學生。
-

這樣的 **實驗設計** 能夠引導學生逐步學習 **數據庫的設計、操作與管理**，同時透過 **phpMyAdmin** 提供直觀的操作方式，幫助他們快速掌握關鍵概念。

這樣的 SQL 檔案應該能夠讓學生順利地 **建立、填充與操作資料庫**，你覺得還需要增加哪些內容嗎？