

这里是一个适合本科生的 E-R 图设计，主题为「学生课程管理系统」，并附上详细的 SQL 语言档案，以便学生在 phpMyAdmin 上操作。

实验目标

- 1. 学习 E-R 图的基本概念，并将其转换为关联式数据库的表格设计。
- 2. 透过 phpMyAdmin 建立数据库与资料表。
- 3. 使用 SQL 语法建立表格并插入资料。
- 4. 练习 SQL 查询、插入、更新、删除等基本操作。

步骤 1：设计 E-R 图

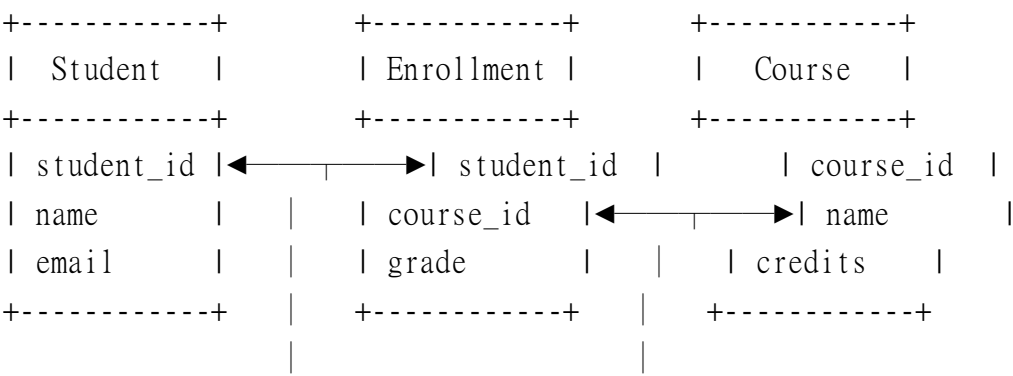
这个 学生课程管理系统 包含以下实体 (Entities)：

- 学生 (Student)：存放学生的基本资料。
- 课程 (Course)：存放课程信息。
- 选课 (Enrollment)：存放学生选修的课程。

关联 (Relationships)

- 一名学生 可选修多门课程 (Student → Enrollment → Course)
- 一门课程 可有多名学生选修 (Course → Enrollment → Student)
- Enrollment 作为多对多 (Many-to-Many) 关系的桥接表 (Bridge Table)

E-R 图 (示意图)



属性说明

- **Student**
 - student_id (PK): 学生编号 (主键)
 - name: 学生姓名
 - email: 学生电子邮件
- **Course**
 - course_id (PK): 课程编号 (主键)
 - name: 课程名称
 - credits: 学分
- **Enrollment**
 - student_id (FK): 学生编号 (外键)
 - course_id (FK): 课程编号 (外键)
 - grade: 成绩 (可选)

步骤 2：在 phpMyAdmin 建立数据库

1. 打开 phpMyAdmin。
2. 在左侧点击「新建」并输入数据库名称：university_db。
3. 点击「建立」。

步骤 3：执行 SQL 指令建立资料表

使用以下 SQL 指令，在 phpMyAdmin 的 SQL 查询区 执行：

```
CREATE DATABASE university_db;
USE university_db;

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

```

        email VARCHAR(100) UNIQUE NOT NULL
    );

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

-- 建立选课表（桥接表）
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
);

```

步骤 4：插入范例资料

让学生在 phpMyAdmin 执行以下 SQL，填充一些测试资料：

```

-- 插入学生资料
INSERT INTO Student (name, email) VALUES
('张三', 'zhangsan@example.com'),
('李四', 'lisi@example.com'),
('王五', 'wangwu@example.com');

-- 插入课程资料
INSERT INTO Course (name, credits) VALUES
('数据库管理', 3),
('计算机网络', 4),
('人工智能导论', 3);

```

```
-- 插入选课资料 (学生与课程对应)
INSERT INTO Enrollment (student_id, course_id, grade) VALUES
(1, 1, 'A'),
(1, 2, 'B+'),
(2, 1, 'A-'),
(2, 3, 'B'),
(3, 2, 'C');
```

步骤 5：SQL 查询与操作

学生可以尝试以下查询，检查数据是否正确：

(1) 查看所有学生与选课信息

```
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) 查询选修「数据库管理」的学生

```
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 = '数据库管理';
```

(3) 更新某学生的成绩

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

(4) 删除一位学生

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

(此操作将自动删除 Enrollment 表中与该学生相关的选课记录)

步骤 6：学习心得与作业

任务

1. 增加一门新的课程，并让所有学生选修该课程。
 2. 撰写一个 SQL 查询，找出所有学过「人工智能导论」且成绩为 A 或 B 的学生。
-

这样的 **实验设计** 能够引导学生逐步学习 **数据库的设计、操作与管理**，同时透过 **phpMyAdmin** 提供直观的操作方式，帮助他们快速掌握关键概念。

这样的 SQL 档案应该能够让学生顺利地 **建立、填充与操作数据库**，你觉得还需要增加哪些内容吗？