

Bài 1:

Thực thể chính

- Instructor(Mã giảng viên, Tên giảng viên, Email)
- Course(Mã khóa học, Tên khóa học, Mô tả)
- Student(Mã SV, Tên SV, Email)
- Enrollment(Mã đăng ký, mã SV, mã khóa học, ngày đăng, tiến độ hoàn thành(%))

Mối quan hệ:

- Instructor 1-N Course
- Student N-N Course
- Student 1-N Enrollment
- Course 1-N Enrollment
- Instructor 1-N Enrollment

Bài 2:

Enrollments_Temp(Mã Sv, Tên Sv, Mã khóa học, Tên khóa học, Tên giảng viên, Email giảng viên, ngày đăng ký)

Vấn đề / vi phạm:

- **1NF:** Có vẻ bảng ở 1NF nếu mỗi ô chứa giá trị nguyên tử (giả sử không có multi-value).
- **2NF:** Nếu PK giả định là (**StudentID**, **CourseID**), thì **StudentName** phụ thuộc hoàn toàn vào **StudentID** (partial? Actually phụ thuộc vào một phần của PK) và **CourseName**, **InstructorName**, **InstructorEmail** phụ thuộc vào **CourseID**. Vì vậy có **vi phạm 2NF** (dữ liệu lặp & phụ thuộc từng phần).
- **3NF:** **InstructorName**, **InstructorEmail** không trực tiếp phụ thuộc vào PK đăng ký mà phụ thuộc vào **CourseID** (và **CourseID** lại phụ thuộc vào PK). Nếu **Course** có thông tin giảng viên, thì **Instructor** là một thực thể riêng — có nghĩa bảng hiện tại có **phụ thuộc chuyển tiếp**, nên **vi phạm 3NF**.

Kết luận: Enrollments_Temp vi phạm **2NF** và **3NF**.

Tách về 3NF — các bảng sau:

1. Students

- StudentID (PK)
- FullName
- Email

2. Instructors

- InstructorID (PK)
- FullName
- Email

3. Courses

- CourseID (PK)
- CourseName
- InstructorID (FK → Instructors.InstructorID)

4. Enrollments

- EnrollmentID (PK) — hoặc composite PK (StudentID, CourseID)
- StudentID (FK → Students.StudentID)
- CourseID (FK → Courses.CourseID)
- EnrollmentDate
- ProgressPercent

Sau khi tách, mỗi thuộc tính chỉ nằm ở bảng nơi nó phụ thuộc hoàn toàn vào PK của bảng đó — đạt 3NF.

Bài 3:

1. Instructors

```
CREATE TABLE Instructors (  
    InstructorID INT AUTO_INCREMENT PRIMARY KEY,  
    FullName VARCHAR(255) NOT NULL,  
    Email VARCHAR(255) NOT NULL UNIQUE  
);
```

- PK: InstructorID
- Email UNIQUE

2. Courses

```
CREATE TABLE Courses (  
  
    CourseID INT AUTO_INCREMENT PRIMARY KEY,  
  
    CourseName VARCHAR(255) NOT NULL,  
  
    Description TEXT,  
  
    InstructorID INT NOT NULL,  
  
    CONSTRAINT fk_courses_instructor FOREIGN KEY (InstructorID) REFERENCES  
Instructors(InstructorID)  
);
```

- PK: CourseID
- FK: InstructorID → Instructors(InstructorID)
- Description dùng TEXT.

3. Students

```
CREATE TABLE Students (  
  
    StudentID INT AUTO_INCREMENT PRIMARY KEY,  
  
    FullName VARCHAR(255) NOT NULL,
```

```
Email VARCHAR(255) NOT NULL UNIQUE,  
  
JoinDate DATE DEFAULT NULL  
  
);
```

- PK: StudentID
- Email UNIQUE.

4. Enrollments

```
CREATE TABLE Enrollments (  
  
    EnrollmentID INT AUTO_INCREMENT PRIMARY KEY,  
  
    StudentID INT NOT NULL,  
  
    CourseID INT NOT NULL,  
  
    EnrollmentDate DATE NOT NULL,  
  
    ProgressPercent DECIMAL(5,2) DEFAULT 0.00, -- ex: 75.50 (%)  
  
    CONSTRAINT fk_enroll_student FOREIGN KEY (StudentID) REFERENCES  
Students(StudentID),  
  
    CONSTRAINT fk_enroll_course FOREIGN KEY (CourseID) REFERENCES  
Courses(CourseID),  
  
    UNIQUE KEY uq_student_course (StudentID, CourseID) -- tránh đăng ký trùng  
  
);
```

- ProgressPercent dùng DECIMAL(5,2) để lưu tới 100.00 (%).
- Khóa duy nhất (StudentID, CourseID) ngăn trùng đăng ký.