

BÀI THỰC HÀNH: XÂY DỰNG ỨNG DỤNG QUẢN LÝ SINH VIÊN

Mục tiêu:

- Thiết kế và triển khai các mối quan hệ phức tạp trong cơ sở dữ liệu: **One-to-Many** và **Many-to-Many**.
- Thành thạo việc sử dụng các annotation quan hệ của JPA (@OneToMany, @ManyToOne, @ManyToMany).
- Xây dựng giao diện dạng cây (tree list) để hiển thị dữ liệu có cấu trúc phân cấp bằng Thymeleaf.
- Hiểu cách truy vấn và quản lý dữ liệu qua các bảng có quan hệ với nhau.

Nội dung:

Thay vì chỉ quản lý sinh viên, hệ thống của bạn giờ đây sẽ quản lý một cấu trúc hoàn của trường học. Một trường có nhiều Khoa (Faculty), mỗi khoa có nhiều Môn học (Subject), mỗi môn học có thể mở nhiều Lớp học (Class), và một sinh viên có thể đăng ký học nhiều lớp khác nhau. Giao diện chính của ứng dụng sẽ hiển thị toàn bộ cấu trúc này dưới dạng cây phân cấp, đồng thời cung cấp các chức năng quản lý cần thiết.

Phần 1: Thiết kế Cơ sở dữ liệu

Để đáp ứng yêu cầu mới, sinh viên cần thiết kế một cơ sở dữ liệu có các bảng sau:

1. faculties (Khoa):

- id (PRIMARY KEY, BIGINT, AUTO_INCREMENT)
- name (VARCHAR)
- Mô tả: Lưu thông tin các khoa, ví dụ: "Công nghệ thông tin", "Quản trị kinh doanh".

2. subjects (Môn học):

- id (PRIMARY KEY, BIGINT, AUTO_INCREMENT)
- name (VARCHAR)
- faculty_id (FOREIGN KEY tham chiếu đến faculties.id)
- Mô tả: Lưu thông tin các môn học, ví dụ: "Lập trình Java", "Cơ sở dữ liệu". Mỗi môn học thuộc về một khoa.

3. student_classes (Lớp học):

- id (PRIMARY KEY, BIGINT, AUTO_INCREMENT)
- name (VARCHAR, ví dụ: "21CTT1", "Sáng T2-T4")
- subject_id (FOREIGN KEY tham chiếu đến subjects.id)
- Mô tả: Lưu thông tin các lớp học được mở cho một môn học cụ thể.

4. students (Sinh viên):

- id (PRIMARY KEY, BIGINT, AUTO_INCREMENT)
- first_name, last_name, email (VARCHAR)
- *Mô tả: Lưu thông tin cá nhân của sinh viên.*

5. enrollments (Bảng đăng ký học):

- student_id (FOREIGN KEY tham chiếu đến students.id)
- class_id (FOREIGN KEY tham chiếu đến student_classes.id)
- *Mô tả: Đây là bảng trung gian để tạo mối quan hệ Nhiều-Nhiều. Một dòng trong bảng này có nghĩa là một sinh viên đã đăng ký vào một lớp học.*

Sơ đồ quan hệ:

[Faculty] 1 <--- * [Subject] 1 <--- * [StudentClass] * <---> * [Student]

Phần 2: Xây dựng Chức năng Backend

Sinh viên cần xây dựng các lớp Model, Repository và Service để hiện thực hóa các chức năng sau:

1. Quản lý Khoa (Faculty Management):

- **Yêu cầu:** Xây dựng các trang giao diện và API để thực hiện đầy đủ các thao tác CRUD (Thêm, Xem, Sửa, Xóa) cho Khoa.

2. Quản lý Môn học (Subject Management):

- **Yêu cầu:** Xây dựng chức năng CRUD cho Môn học. Giao diện thêm/sửa một môn học phải cung cấp một danh sách dropdown để người dùng chọn Khoa mà môn học đó thuộc về.

3. Quản lý Lớp học (Class Management):

- **Yêu cầu:** Xây dựng chức năng CRUD cho Lớp học. Giao diện thêm/sửa một lớp học phải cho phép chọn Môn học tương ứng.

4. Quản lý Sinh viên (Student Management):

- **Yêu cầu:** Xây dựng chức năng CRUD cho Sinh viên.

5. Chức năng Đăng ký học (Enrollment):

- **Yêu cầu:** Xây dựng một giao diện cho phép người quản trị:
 - Chọn một Lớp học cụ thể.
 - Hiện thị danh sách các sinh viên đã đăng ký và danh sách các sinh viên chưa đăng ký vào lớp đó.
 - Cung cấp chức năng để thêm một hoặc nhiều sinh viên vào lớp học, hoặc xóa sinh viên ra khỏi lớp.
- Backend phải xử lý logic để tạo hoặc xóa các bản ghi tương ứng trong bảng enrollments.

Phần 3: Xây dựng Giao diện Hiển thị (UI)

Chức năng 1: Giao diện Dạng Cây (Tree View)

- **Yêu cầu:** Xây dựng một trang chính (/) hiển thị toàn bộ cấu trúc của trường học dưới dạng cây phân cấp. Cấu trúc cây phải rõ ràng theo thứ tự: **Khoa -> Môn học -> Lớp học -> Sinh viên**.
- **Gợi ý:** Sử dụng các vòng lặp `th:each` lồng nhau của Thymeleaf để render cấu trúc này. Controller cần chuẩn bị một danh sách các Khoa, trong đó mỗi Khoa đã chứa sẵn danh sách các Môn học, mỗi Môn học chứa các Lớp học, và mỗi Lớp học chứa các Sinh viên đã đăng ký.

Chức năng 2: Giao diện Quản lý Chi tiết

- **Yêu cầu:** Bên cạnh giao diện dạng cây, cần có các trang riêng biệt để thực hiện các thao tác quản lý (CRUD) cho từng đối tượng: Khoa, Môn học, Lớp học và Sinh viên.
- Ví dụ: một trang `/faculties` sẽ hiển thị bảng danh sách các khoa với các nút Thêm/Sửa/Xóa.

Giao diện thêm sinh viên vào lớp học:

Quản lý Đăng ký học của Sinh viên

Chọn sinh viên:

Đăng ký cho sinh viên:

Các lớp học hiện có

Các lớp đã đăng ký



Lưu thay đổi

Quay lại

Giao diện hiển thị sinh viên – lớp học:



Khoa:



Môn học:



Lớp:

— 👤 - ()

Chưa có sinh viên nào đăng ký.