

DÀI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



## HỆ CƠ SỞ DỮ LIỆU

### Báo cáo Bài tập lớn 2

Học kì 242 - L03 - Nhóm 10

# ỨNG DỤNG QUẢN LÍ KÍ TÚC XÁ

Giảng viên hướng dẫn: Msc. Lê Thị Bảo Thu

Sinh viên thực hiện: Doãn Phương Hùng Cường ID 2310381  
Nguyễn Khắc Khánh Huy ID 2311186  
Nguyễn Hữu Khang ID 2311449  
Đặng Quang Phong ID 2312613

THÀNH PHỐ HỒ CHÍ MINH, 28 THÁNG 4 2025



Nội dung	MSSV	Ghi chú
<b>Phần 1: TẠO CƠ SỞ DỮ LIỆU TRÊN MySQL</b>		
I. Tạo bảng dữ liệu (2đ)		
- Bảng 1-7	2312613	
- Bảng 8-14	2311449	
- Bảng 15-21	2311186	
- Bảng 22-27	2310381	
II. Insert dữ liệu (1.5đ)		
- Bảng 1-7	2312613	
- Bảng 8-14	2311449	
- Bảng 15-21	2311186	
- Bảng 22-27	2310381	
<b>Phần 2: PROCEDURE, FUNCTION, TRIGGER</b>		
3.1 Procedure		
3.1.1 Kiểm tra phòng đủ người	2312613	
3.1.2 Danh sách tòa chưa đủ người/phòng	2311449	
3.1.3 Danh sách phòng chưa đầy người	2311186	
3.1.4 Liệt kê thiết bị cần bảo trì	2310381	
3.1.5 Danh sách sinh viên không có người thân	2312613	
3.1.6 Thống kê kỷ luật sinh viên theo lỗi	2311449	
3.2 Function		
3.2.1 Số sinh viên bị kỷ luật trong khoảng thời gian (start-end)	2311186	
3.2.2 Tổng số sinh viên trong một tòa nhà	2310381	
3.2.3 Số thẻ nội trú chưa hết hạn	2312613	
3.3 Trigger		
3.3.1 Trigger tự động tạo emp_id khi thêm nhân viên mới	2311449	



3.3.2 Trigger kiểm tra và cập nhật khi thêm sinh viên mới	2311186	
3.3.3 Trigger cập nhật số lượng nhân viên khi thêm nhân viên quản lý	2310381	
3.3.4 Trigger cập nhật số lượng nhân viên khi thêm nhân viên y tế	2312613	
3.3.5 Trigger cập nhật số lượng nhân viên khi thêm nhân viên khác	2311449	
3.3.6 Trigger cập nhật khi xóa sinh viên khỏi hệ thống	2311186	
3.3.7 Trigger cập nhật khi xóa nhân viên khác khỏi hệ thống	2310381	
3.3.8 Trigger xóa bằng cấp nhân viên y tế khi xóa nhân viên	2312613	
3.3.9 Trigger xóa bằng cấp nhân viên quản lý khi xóa nhân viên quản lý	2311449	
3.3.10 Trigger khi chuyển phòng cho sinh viên	2311186	
3.3.11 Trigger xử lý sinh viên bị đuổi khỏi KTX (3 lần 'High')	2310381	
3.3.12 Trigger chuyển phòng cho nhân viên khác	2312613	
3.3.13 Trigger chuyển phòng cho nhân viên quản lý	2311449	
3.3.14 Trigger xử lý sinh viên bị đuổi khỏi KTX	2311186	
3.3.15 Trigger kiểm tra phòng không thể trùng loại phòng	2310381	
3.3.16 Trigger kiểm tra thời gian của phòng quản lý	2312613	
3.3.17 Trigger cập nhật tỷ lệ lấp đầy của phòng ở	2311449	



**Phần 3: XÂY DỰNG ỨNG DỤNG**

I. Tạo user, cấp quyền (0.5đ)	2311186	
II. Hiện thực chức năng		
2.1.1 Đăng nhập (0.25đ)	2310381	
2.1.2 Đăng xuất (0.25đ)	2310381	
2.2.1 Xem, thêm, sửa, xoá dữ liệu (1đ)	2312613	
2.2.2 Danh sách đối tượng, tìm kiếm, sắp xếp (1đ)	2311449	
2.2.3 Tính năng nghiệp vụ khác (0.5đ)	2311186	



## Mục lục

<b>1</b>	<b>Mở đầu</b>	<b>1</b>
<b>2</b>	<b>Tạo bảng cơ sở dữ liệu</b>	<b>1</b>
2.1	Tạo bảng dữ liệu . . . . .	1
2.1.1	Bảng người trong ký túc xá (People) . . . . .	1
2.1.2	Bảng số điện thoại (phone_number) . . . . .	2
2.1.3	Bảng địa chỉ người trong ký túc xá (Address) . . . . .	2
2.1.4	Bảng email người trong ký túc xá (Email) . . . . .	3
2.1.5	Bảng tòa nhà trong ký túc xá (Building) . . . . .	3
2.1.6	Bảng phòng trong ký túc xá (Room) . . . . .	4
2.1.7	Bảng nhân viên trong ký túc xá (Employee) . . . . .	4
2.1.8	Bảng phòng ở trong ký túc xá (Living_Room) . . . . .	4
2.1.9	Bảng sinh viên (Student) . . . . .	5
2.1.10	Bảng phòng quản lý (Management_Room) . . . . .	6
2.1.11	Bảng nhân viên quản lý (Manager_Employee) . . . . .	6
2.1.12	Bảng bằng cấp của nhân viên quản lý (Manager_Degree) . . . . .	7
2.1.13	Bảng phòng y tế (Health_Room) . . . . .	7
2.1.14	Bảng nhân viên y tế (Medical_Employee) . . . . .	8
2.1.15	Bảng bằng cấp nhân viên y tế (Medical_Degree) . . . . .	8
2.1.16	Bảng phòng khác (Other_Room) . . . . .	8
2.1.17	Bảng nhân viên phòng khác (Other_Employee) . . . . .	9
2.1.18	Bảng thời gian làm việc nhân viên phòng khác (Other_Employee_Time_Work) . . . . .	9
2.1.19	Bảng thiết bị lớn (Large_Equipment) . . . . .	10
2.1.20	Bảng thiết bị theo tòa nhà (Building_Equipment) . . . . .	10
2.1.21	Bảng người thân của sinh viên (Relative) . . . . .	11
2.1.22	Bảng kỷ luật (Disciplinary_Action) . . . . .	11
2.1.23	Bảng kỷ luật sinh viên (Student_Disipline) . . . . .	12
2.1.24	Bảng nhân viên quản lý phụ trách kỷ luật (Discipline_Staff) . . . . .	13
2.1.25	Bảng căn cước sinh viên (Identity_Card) . . . . .	13
2.1.26	Bảng thẻ ký túc xá (Dormitory_Card) . . . . .	13
2.1.27	Bảng thẻ nhân viên (Employee_Card) . . . . .	14
2.2	Thêm dữ liệu mẫu . . . . .	15
2.2.1	People . . . . .	15



2.2.2	Bảng phone_number . . . . .	17
2.2.3	Bảng address . . . . .	17
2.2.4	Bảng email . . . . .	19
2.2.5	Bảng building . . . . .	21
2.2.6	Bảng room . . . . .	21
2.2.7	Bảng employee . . . . .	22
2.2.8	Bảng living_room . . . . .	22
2.2.9	Bảng student . . . . .	23
2.2.10	Bảng management_room . . . . .	25
2.2.11	Bảng manager_employee . . . . .	26
2.2.12	Bảng manager_degree . . . . .	26
2.2.13	Bảng health_room . . . . .	26
2.2.14	Bảng medical_employee . . . . .	26
2.2.15	Bảng medical_degree . . . . .	27
2.2.16	Bảng other_room . . . . .	27
2.2.17	Bảng other_employee . . . . .	27
2.2.18	Bảng other_employee_time_work . . . . .	27
2.2.19	Bảng large_equipment . . . . .	28
2.2.20	Bảng building_equipment . . . . .	28
2.2.21	Bảng relative . . . . .	28
2.2.22	Bảng disciplinary_action . . . . .	31
2.2.23	Bảng student_discipline . . . . .	32
2.2.24	Bảng discipline_staff . . . . .	32
2.2.25	Bảng identity_card . . . . .	32
2.2.26	Bảng employee_card . . . . .	33
2.2.27	Bảng dormitory_card . . . . .	33
<b>3</b>	<b>Thủ tục - Hàm - Trigger</b>	<b>35</b>
3.1	Thủ tục . . . . .	35
3.1.1	Kiểm tra xem một phòng cụ thể đã đủ người chưa . . . . .	35
3.1.2	Thủ tục lấy thông tin sinh viên theo SSN . . . . .	36
3.1.3	Thống kê kỷ luật sinh viên theo từng lỗi vi phạm . . . . .	38
3.2	Hàm . . . . .	39
3.2.1	Số sinh viên bị kỉ luật trong khoảng thời gian (start-end) . . . . .	39
3.2.2	Tổng số sinh viên trong một tòa nhà . . . . .	40
3.2.3	Số thẻ nội trú chưa hết hạn . . . . .	41



---

3.3 Trigger . . . . .	42
3.3.1 Trigger khi chuyển phòng cho sinh viên . . . . .	42
3.3.2 Trigger kiểm tra và cập nhật khi thêm sinh viên mới . . . . .	44
3.3.3 Trigger xử lý sinh viên bị đuổi khỏi KTX . . . . .	46
3.3.4 Một số trigger khác . . . . .	48
<b>4 Triển khai ứng dụng</b>	<b>49</b>
4.1 Tạo user . . . . .	49
4.2 Giới thiệu môi trường lập trình . . . . .	49
4.3 Cách kết nối ứng dụng với CSDL MySQL . . . . .	50
4.3.1 Database Driver . . . . .	50
4.3.2 Cách kết nối . . . . .	51
4.4 Thiết kế hệ thống API . . . . .	52
4.5 Minh họa phần lập trình . . . . .	55
4.5.1 Mô tả lớp models (services) . . . . .	55
4.5.2 Minh họa code chức năng Select . . . . .	56
4.5.3 Minh họa code cho hàm hiện thực ở phần 2 . . . . .	56
4.6 Thiết kế giao diện web . . . . .	57

## Danh sách hình vẽ

2.1 Biểu đồ EERD cho hệ thống cơ sở dữ liệu ký túc xá . . . . .	15
3.1 Trước khi cập nhật . . . . .	43
3.2 Sau khi cập nhật . . . . .	44
3.3 Trước khi kỷ luật sinh viên . . . . .	47
3.4 Sau khi kỷ luật sinh viên . . . . .	48
4.1 Hệ thống API . . . . .	52
4.2 Trang chủ . . . . .	57
4.3 Trang đăng nhập . . . . .	57
4.4 Trang sau đăng nhập . . . . .	58
4.5 Trang quản lý sinh viên . . . . .	58
4.6 Trang thông tin sinh viên . . . . .	59
4.7 Trang chỉnh sửa thông tin sinh viên . . . . .	60
4.8 Xác nhận/Huỷ chỉnh sửa thông tin sinh viên . . . . .	61
4.9 Xoá sinh viên khỏi ký túc xá . . . . .	62
4.10 Trang quản lý phòng . . . . .	62



4.11	Trang thêm sinh viên vào ký túc xá . . . . .	63
4.12	Trang thống kê cho quản lý . . . . .	64
4.13	Trang thống kê với số liệu . . . . .	65



# 1 Mở đầu

Qua Assignment 1, nhóm đã xây dựng và mô tả một hệ cơ sở dữ liệu cho dự án ký túc xá của Trường Đại học Bách khoa - Đại học Quốc gia Thành phố Hồ Chí Minh với đầy đủ các yếu tố cốt lõi và chi tiết, cũng như thiết kế EERD và biểu diễn trực quan hóa các mối quan hệ bằng schema. Đó là tiền đề cần thiết để nhóm tiếp tục triển khai Assignment 2: thực hiện xây dựng một hệ thống cơ sở dữ liệu hoàn chỉnh dựa trên những tiêu chí đã chọn từ Assignment 1. Báo cáo bao gồm 3 phần chính:

- Tạo cơ sở dữ liệu với đầy đủ các bảng, ràng buộc và dữ liệu mẫu.
- Xây dựng các thủ tục lưu trữ, hàm và trigger để xử lý nghiệp vụ và kiểm soát dữ liệu
- Phát triển một ứng dụng kết nối với cơ sở dữ liệu, cho phép thực hiện các thao tác như đăng nhập, quản lý dữ liệu, truy vấn thông tin, và tương tác với các hàm/thủ tục đã xây dựng

Mục tiêu của báo cáo là thể hiện khả năng thiết kế, triển khai và ứng dụng hệ quản trị cơ sở dữ liệu vào thực tiễn, đồng thời đảm bảo hệ thống vận hành hiệu quả, đúng yêu cầu nghiệp vụ và thân thiện với người dùng.

## 2 Tạo bảng cơ sở dữ liệu

### 2.1 Tạo bảng dữ liệu

Trong bài báo cáo này, nhóm sử dụng ngôn ngữ MySQL để thực hiện các thao tác tạo bảng, ràng buộc và dữ liệu mẫu cho hệ thống cơ sở dữ liệu ký túc xá. Các bảng được tạo ra dựa trên thiết kế EERD đã được trình bày trong báo cáo Assignment 1. Đây là biểu đồ cơ sở dữ liệu theo kiểu EERD kết hợp với ERD, thể hiện các thực thể, thuộc tính và mối quan hệ giữa chúng.

#### 2.1.1 Bảng người trong ký túc xá (People)

- **ssn**: Căn cước công dân, khoá chính.
- **last\_name**: Tên người, không được rỗng.
- **first\_name**: Họ và tên lót, không được rỗng.
- **birthday**: Ngày sinh, không được rỗng.
- **sex**: Giới tính, Nam (M) hoặc Nữ (F).



- **health\_state**: Tình trạng sức khoẻ.
- **ethnic\_group**: Dân tộc, không được rỗng.

```
1 CREATE TABLE people (
2     ssn CHAR(8) PRIMARY KEY,
3     last_name VARCHAR(20) NOT NULL,
4     first_name VARCHAR(20) NOT NULL,
5     birthday DATE NOT NULL,
6     sex CHAR(1) CHECK (sex IN ('M', 'F')),
7     health_state VARCHAR(100),
8     ethnic_group VARCHAR(30)
9 );
```

### 2.1.2 Bảng số điện thoại (phone\_number)

- **ssn**: Căn cước công dân, khoá chính (cùng với số điện thoại).
- **phone\_number**: Số điện thoại, khoá chính (cùng với căn cước công dân).

```
1 CREATE TABLE phone_number (
2     ssn CHAR(8),
3     phone_number CHAR(10),
4     PRIMARY KEY (ssn, phone_number),
5     CONSTRAINT fk_phone_number FOREIGN KEY (ssn) REFERENCES people(ssn)
6         ON DELETE CASCADE ON UPDATE CASCADE
7 );
```

### 2.1.3 Bảng địa chỉ người trong ký túc xá (Address)

- **ssn**: Căn cước công dân, khoá chính (cùng với xã, huyện, tỉnh).
- **commune**: Xã/Phường, khoá chính (cùng với căn cước công dân).
- **district**: Quận/Huyện, khoá chính (cùng với căn cước công dân).
- **province**: Tỉnh/Thành phố, khoá chính (cùng với căn cước công dân).

```
1 CREATE TABLE address (
2     ssn CHAR(8),
3     commune VARCHAR(30),
4     district VARCHAR(30),
5     province VARCHAR(30),
6     PRIMARY KEY (ssn, commune, district, province),
```



```
7   CONSTRAINT fk_address FOREIGN KEY (ssn) REFERENCES people(ssn)
8     ON DELETE CASCADE ON UPDATE CASCADE
9 );
```

#### 2.1.4 Bảng email người trong ký túc xá (Email)

- **ssn**: Căn cước công dân, khoá chính (cùng với email).
- **email**: Địa chỉ email, khoá chính (cùng với căn cước công dân).

```
1 CREATE TABLE email (
2   ssn CHAR(8),
3   email VARCHAR(50),
4   PRIMARY KEY (ssn, email),
5   CONSTRAINT fk_email FOREIGN KEY (ssn) REFERENCES people(ssn)
6     ON DELETE CASCADE ON UPDATE CASCADE
7 );
```

#### 2.1.5 Bảng tòa nhà trong ký túc xá (Building)

- **building\_id**: Mã tòa nhà, khoá chính.
- **building\_name**: Tên tòa nhà, không được rỗng.
- **floors**: Số tầng, lớn hơn 0.
- **rooms**: Số lượng phòng, lớn hơn 0.
- **has\_air\_conditioner**: Có điều hoà hay không (TRUE/FALSE).
- **sponsor**: Đơn vị tài trợ (nếu có).
- **construction\_date**: Ngày xây dựng, không được rỗng.
- **last\_renovation**: Ngày sửa chữa gần nhất (nếu có).

```
1 CREATE TABLE building (
2   building_id CHAR(5) PRIMARY KEY,
3   building_name CHAR(5) NOT NULL,
4   floors INT NOT NULL CHECK (floors > 0),
5   rooms INT NOT NULL CHECK (rooms > 0),
6   has_air_conditioner BOOLEAN NOT NULL,
7   sponsor VARCHAR(100),
8   construction_date DATE NOT NULL,
9   last_renovation DATE
10 );
```



#### 2.1.6 Bảng phòng trong ký túc xá (Room)

- **building\_id**: Mã tòa nhà, khoá chính (cùng với mã phòng).
- **room\_id**: Mã phòng, khoá chính (cùng với mã tòa nhà).
- **room\_status**: Trạng thái phòng: Có thể là Available, Occupied hoặc Under Maintenance.
- **room\_area**: Diện tích phòng ( $m^2$ ), lớn hơn 0.

```
1 CREATE TABLE room (
2     building_id CHAR(5),
3     room_id CHAR(5),
4     room_status ENUM('Available', 'Occupied', 'Under Maintenance') NOT NULL,
5     room_area DECIMAL(10, 2) NOT NULL CHECK (room_area > 0),
6     PRIMARY KEY (building_id, room_id),
7     CONSTRAINT fk_room_building FOREIGN KEY (building_id) REFERENCES building(
8         building_id)
9         ON DELETE CASCADE ON UPDATE CASCADE
);
```

#### 2.1.7 Bảng nhân viên trong ký túc xá (Employee)

- **essn**: Căn cước công dân của nhân viên, khoá chính.
- **salary**: Mức lương của nhân viên.
- **emp\_id**: Mã nhân viên nội bộ, có thể rỗng.

```
1 CREATE TABLE employee (
2     essn CHAR(8) PRIMARY KEY,
3     salary DECIMAL(10, 0),
4     emp_id CHAR(7) NULL,
5     CONSTRAINT fk_essn FOREIGN KEY (essn) REFERENCES people(ssn)
6         ON DELETE CASCADE ON UPDATE CASCADE
7 );
```

#### 2.1.8 Bảng phòng ở trong ký túc xá (Living\_Room)

- **building\_id**: Mã tòa nhà, khoá chính (cùng với mã phòng).
- **room\_id**: Mã phòng, khoá chính (cùng với mã tòa nhà).
- **max\_num\_of\_students**: Số lượng sinh viên tối đa.
- **current\_num\_of\_students**: Số lượng sinh viên hiện tại (mặc định 0).
- **rental\_price**: Giá thuê phòng (đơn vị tiền tệ).



- occupancy\_rate: Tỷ lệ lấp đầy (%), từ 0 đến 100.

```

1 CREATE TABLE living_room (
2     building_id CHAR(5),
3     room_id CHAR(5),
4     max_num_of_students INT NOT NULL CHECK (max_num_of_students > 0),
5     current_num_of_students INT NOT NULL DEFAULT 0
6         CHECK (current_num_of_students >= 0),
7     rental_price DECIMAL(10, 2) NOT NULL CHECK (rental_price >= 0),
8     occupancy_rate DECIMAL(5, 2) NOT NULL DEFAULT 0
9         CHECK (occupancy_rate >= 0 AND occupancy_rate <= 100),
10    CHECK (current_num_of_students <= max_num_of_students),
11    PRIMARY KEY (building_id, room_id)
12);

```

### 2.1.9 Bảng sinh viên (Student)

- sssn: Căn cước công dân sinh viên, khoá chính.
- student\_id: Mã sinh viên, không được rỗng.
- has\_health\_insurance: Có bảo hiểm y tế hay không (TRUE/FALSE).
- study\_status: Tình trạng học tập (đang học, bảo lưu, nghỉ học...).
- class\_name: Tên lớp.
- faculty: Khoa.
- building\_id: Toà nhà sinh viên ở.
- room\_id: Phòng sinh viên ở.

```

1 CREATE TABLE student (
2     sssn CHAR(8) PRIMARY KEY,
3     student_id CHAR(7) NOT NULL,
4     has_health_insurance BOOLEAN NOT NULL,
5     study_status VARCHAR(20),
6     class_name VARCHAR(20),
7     faculty VARCHAR(50),
8     building_id CHAR(5),
9     room_id CHAR(5),
10    CONSTRAINT fk_student_p FOREIGN KEY (sssn) REFERENCES people(ssn)
11        ON UPDATE CASCADE ON DELETE CASCADE,
12    CONSTRAINT fk_student_room FOREIGN KEY (building_id, room_id)

```



```
13     REFERENCES living_room(building_id, room_id)
14 );
```

### 2.1.10 Bảng phòng quản lý (Management\_Room)

- building\_id: Mã tòa nhà, khoá chính (cùng với room\_id).
- room\_id: Mã phòng, khoá chính (cùng với building\_id).
- num\_of\_staff: Số lượng nhân viên quản lý (mặc định 0).
- start\_time: Giờ bắt đầu làm việc.
- end\_time: Giờ kết thúc làm việc.

```
1 CREATE TABLE management_room (
2     building_id CHAR(5),
3     room_id CHAR(5),
4     num_of_staff INT NOT NULL DEFAULT 0 CHECK (num_of_staff >= 0),
5     start_time TIME NOT NULL,
6     end_time TIME NOT NULL,
7     PRIMARY KEY (building_id, room_id),
8     CONSTRAINT fk_mgmt_room FOREIGN KEY (building_id, room_id) REFERENCES room(
9         building_id, room_id)
10    ON DELETE CASCADE ON UPDATE CASCADE
);
```

### 2.1.11 Bảng nhân viên quản lý (Manager\_Employee)

- essn: Căn cước nhân viên, khoá chính.
- room\_id: Phòng mà nhân viên quản lý.
- building\_id: Tòa nhà mà nhân viên quản lý.
- experience: Số năm kinh nghiệm.

```
1 CREATE TABLE manager_employee (
2     essn CHAR(8) PRIMARY KEY,
3     room_id CHAR(5),
4     building_id CHAR(5),
5     experience DECIMAL(2,0),
6     CONSTRAINT fk_mgr_essn FOREIGN KEY (essn)
7         REFERENCES employee(essn)
8         ON DELETE CASCADE ON UPDATE CASCADE,
```



```

9   CONSTRAINT fk_manager_employee_room FOREIGN KEY (building_id, room_id)
10  REFERENCES management_room(building_id, room_id)
11 );

```

### 2.1.12 Bảng bằng cấp của nhân viên quản lý (Manager\_Degree)

- essn: Căn cước nhân viên.
- degree: Tên bằng cấp.

```

1 CREATE TABLE manager_degree (
2   essn CHAR(8),
3   degree VARCHAR(50),
4   PRIMARY KEY (essn, degree),
5   CONSTRAINT fk_mgr_deg FOREIGN KEY (essn) REFERENCES manager_employee(essn)
6     ON DELETE CASCADE ON UPDATE CASCADE
7 );

```

### 2.1.13 Bảng phòng y tế (Health\_Room)

- building\_id: Mã tòa nhà, khoá chính (cùng với room\_id).
- room\_id: Mã phòng, khoá chính (cùng với building\_id).
- num\_of\_staff: Số lượng nhân viên y tế (mặc định 0).
- start\_time: Giờ bắt đầu hoạt động.
- end\_time: Giờ kết thúc hoạt động.
- equipment\_status: Tình trạng thiết bị ('Good', 'Under Repair', 'Not Available').

```

1 CREATE TABLE health_room (
2   building_id CHAR(5),
3   room_id CHAR(5),
4   num_of_staff INT NOT NULL DEFAULT 0 CHECK (num_of_staff >= 0),
5   start_time TIME NOT NULL,
6   end_time TIME NOT NULL,
7   equipment_status ENUM('Good', 'Under Repair', 'Not Available') NOT NULL,
8   PRIMARY KEY (building_id, room_id),
9   CONSTRAINT fk_health_room FOREIGN KEY (building_id, room_id)
10    REFERENCES room(building_id, room_id)
11      ON DELETE CASCADE ON UPDATE CASCADE
12 );

```



#### 2.1.14 Bảng nhân viên y tế (Medical\_Employee)

- **essn:** Căn cước nhân viên y tế, khoá chính.
- **room\_id:** Phòng làm việc.
- **building\_id:** Toà nhà làm việc.
- **experience:** Số năm kinh nghiệm.

```
1 CREATE TABLE medical_employee (
2     essn CHAR(8) PRIMARY KEY,
3     room_id CHAR(5),
4     building_id CHAR(5),
5     experience DECIMAL(2,0),
6     CONSTRAINT fk_med_essn FOREIGN KEY (essn) REFERENCES employee(essn)
7         ON DELETE CASCADE ON UPDATE CASCADE,
8     CONSTRAINT fk_medical_employee_room FOREIGN KEY (building_id, room_id)
9         REFERENCES health_room(building_id, room_id)
10 );
```

#### 2.1.15 Bảng bằng cấp nhân viên y tế (Medical\_Degree)

- **essn:** Căn cước nhân viên y tế.
- **degree:** Tên bằng cấp.

```
1 CREATE TABLE medical_degree (
2     essn CHAR(8),
3     degree VARCHAR(50),
4     PRIMARY KEY (essn, degree),
5     CONSTRAINT fk_med_deg FOREIGN KEY (essn) REFERENCES medical_employee(essn)
6         ON DELETE CASCADE ON UPDATE CASCADE
7 );
```

#### 2.1.16 Bảng phòng khác (Other\_Room)

- **building\_id:** Mã toà nhà, khoá chính (cùng với room\_id).
- **room\_id:** Mã phòng, khoá chính (cùng với building\_id).
- **room\_type:** Loại phòng (ví dụ: phòng học, phòng hội họp, phòng giải trí,...).
- **start\_time:** Giờ bắt đầu sử dụng.
- **end\_time:** Giờ kết thúc sử dụng.
- **num\_of\_staff:** Số lượng nhân viên quản lý phòng (mặc định 0).



```
1 CREATE TABLE other_room (
2     building_id CHAR(5),
3     room_id CHAR(5),
4     room_type VARCHAR(100) NOT NULL,
5     start_time TIME NOT NULL,
6     end_time TIME NOT NULL,
7     num_of_staff INT NOT NULL DEFAULT 0 CHECK (num_of_staff >= 0),
8     PRIMARY KEY (building_id, room_id),
9     CONSTRAINT fk_other_room FOREIGN KEY (building_id, room_id)
10    REFERENCES room(building_id, room_id)
11    ON DELETE CASCADE ON UPDATE CASCADE
12 );
```

### 2.1.17 Bảng nhân viên phòng khác (Other\_Employee)

- **essn**: Căn cước nhân viên, khoá chính.
- **room\_id**: Phòng làm việc.
- **building\_id**: Toà nhà làm việc.
- **type\_work**: Loại công việc (ví dụ: giảng viên, tư vấn, giám sát,...).

```
1 CREATE TABLE other_employee (
2     essn CHAR(8) PRIMARY KEY,
3     room_id CHAR(5),
4     building_id CHAR(5),
5     type_work VARCHAR(20),
6     CONSTRAINT fk_oth_essn FOREIGN KEY (essn) REFERENCES employee(essn)
7         ON DELETE CASCADE ON UPDATE CASCADE,
8     CONSTRAINT fk_other_employee_room FOREIGN KEY (building_id, room_id)
9         REFERENCES other_room(building_id, room_id)
10);
```

### 2.1.18 Bảng thời gian làm việc nhân viên phòng khác (Other\_Employee\_Time\_Work)

- **essn**: Căn cước nhân viên.
- **date\_work**: Ngày làm việc (các giá trị: Monday, Tuesday, ..., Other).
- **time\_start**: Giờ bắt đầu làm việc.



- time\_end: Giờ kết thúc làm việc.

```
1 CREATE TABLE other_employee_time_work (
2     essn CHAR(8),
3     date_work ENUM('monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday', 'other'),
4     time_start TIME NOT NULL,
5     time_end TIME NOT NULL,
6     PRIMARY KEY (essn, date_work, time_start, time_end),
7     CONSTRAINT fk_other_time FOREIGN KEY (essn) REFERENCES other_employee(essn)
8 );
```

#### 2.1.19 Bảng thiết bị lớn (Large\_Equipment)

- equipment\_id: Mã thiết bị, khoá chính.
- equipment\_name: Tên thiết bị.
- equipment\_type: Loại thiết bị.
- equipment\_status: Tình trạng thiết bị (Working, Damaged, Under Repair).
- purchase\_date: Ngày mua thiết bị.
- purchase\_location: Nơi mua thiết bị.

```
1 CREATE TABLE large_equipment (
2     equipment_id CHAR(10) PRIMARY KEY,
3     equipment_name VARCHAR(100) NOT NULL,
4     equipment_type VARCHAR(50) NOT NULL,
5     equipment_status ENUM('Working', 'Damaged', 'Under Repair') NOT NULL,
6     purchase_date DATE NOT NULL,
7     purchase_location VARCHAR(255)
8 );
```

#### 2.1.20 Bảng thiết bị theo tòa nhà (Building\_Equipment)

- building\_id: Mã tòa nhà, khoá chính (cùng với equipment\_id).
- equipment\_id: Mã thiết bị, khoá chính (cùng với building\_id).
- quantity: Số lượng thiết bị.

```
1 CREATE TABLE building_equipment (
2     building_id CHAR(5),
3     equipment_id CHAR(10),
```



```

4   quantity INT NOT NULL CHECK (quantity > 0),
5   PRIMARY KEY (building_id, equipment_id),
6   CONSTRAINT fk_building_equipment_building FOREIGN KEY (building_id)
7     REFERENCES building(building_id) ON DELETE CASCADE ON UPDATE CASCADE,
8   CONSTRAINT fk_building_equipment_equipment FOREIGN KEY (equipment_id)
9     REFERENCES large_equipment(equipment_id)
10    ON DELETE CASCADE ON UPDATE CASCADE
11 );

```

### 2.1.21 Bảng người thân của sinh viên (Relative)

- **sssn:** Căn cước sinh viên, khoá chính (kết hợp fname và lname).
- **fname:** Tên người thân, không được rỗng.
- **lname:** Họ người thân, không được rỗng.
- **birthday:** Ngày sinh của người thân.
- **relationship:** Quan hệ với sinh viên.
- **address:** Địa chỉ người thân.
- **phone\_number:** Số điện thoại (nếu có).
- **job:** Nghề nghiệp của người thân.

```

1 CREATE TABLE relative (
2   sssn CHAR(8),
3   fname VARCHAR(20) NOT NULL,
4   lname VARCHAR(20) NOT NULL,
5   birthday DATE NOT NULL,
6   relationship VARCHAR(50) NOT NULL,
7   address VARCHAR(255) NOT NULL,
8   phone_number CHAR(10),
9   job VARCHAR(50),
10  PRIMARY KEY (sssn, fname, lname),
11  CONSTRAINT fk_relative FOREIGN KEY (sssn) REFERENCES student(sssn)
12    ON DELETE CASCADE ON UPDATE CASCADE
13 );

```

### 2.1.22 Bảng kỷ luật (Disciplinary\_Action)

- **action\_id:** Mã hành động kỷ luật, khoá chính.
- **action\_type:** Loại hình kỷ luật.



- **reason:** Lý do kỷ luật.
- **decision\_date:** Ngày ra quyết định.
- **effective\_from:** Ngày hiệu lực bắt đầu.
- **effective\_to:** Ngày hiệu lực kết thúc (có thể NULL).
- **severity\_level:** Mức độ vi phạm (low, medium, high, expulsion).
- **status:** Trạng thái (pending, active, completed, cancelled).

```
1 CREATE TABLE disciplinary_action (
2     action_id VARCHAR(20) PRIMARY KEY,
3     action_type VARCHAR(50) NOT NULL,
4     reason TEXT NOT NULL,
5     decision_date DATE NOT NULL,
6     effective_from DATE NOT NULL,
7     effective_to DATE,
8     severity_level VARCHAR(20) CHECK (
9         severity_level IN ('low', 'medium', 'high', 'expulsion')
10    ),
11    status VARCHAR(20) CHECK (
12        status IN ('pending', 'active', 'completed', 'cancelled')
13    )
14);
```

### 2.1.23 Bảng kỷ luật sinh viên (Student\_Discipline)

- **action\_id:** Mã hành động kỷ luật.
- **sssn:** Căn cước sinh viên.

```
1 CREATE TABLE student_discipline (
2     action_id VARCHAR(20),
3     sssn CHAR(8),
4     PRIMARY KEY (action_id, sssn),
5     CONSTRAINT fk_student_discipline_d FOREIGN KEY (action_id)
6         REFERENCES disciplinary_action(action_id)
7         ON DELETE CASCADE ON UPDATE CASCADE,
8     CONSTRAINT fk_student_discipline_s FOREIGN KEY (sssn)
9         REFERENCES student(sssn) ON DELETE CASCADE ON UPDATE CASCADE
10);
```



#### 2.1.24 Bảng nhân viên quản lý phụ trách kỷ luật (Discipline\_Staff)

- action\_id: Mã hành động kỷ luật.
- essn: Căn cước nhân viên quản lý.

```
1 CREATE TABLE discipline_staff (
2     action_id VARCHAR(20),
3     essn CHAR(8),
4     PRIMARY KEY (action_id, essn),
5     CONSTRAINT fk_disciplineStaff_d FOREIGN KEY (action_id)
6         REFERENCES disciplinary_action(action_id)
7         ON DELETE CASCADE ON UPDATE CASCADE,
8     CONSTRAINT fk_disciplineStaff_m FOREIGN KEY (essn)
9         REFERENCES manager_employee(essn)
10        ON DELETE CASCADE ON UPDATE CASCADE
11 );
```

#### 2.1.25 Bảng căn cước sinh viên (Identity\_Card)

- id: Mã thẻ căn cước, khoá chính.
- issued\_date: Ngày cấp thẻ căn cước.

```
1 CREATE TABLE identity_card (
2     id CHAR(7) NOT NULL PRIMARY KEY,
3     issued_date DATE NOT NULL
4 );
```

#### 2.1.26 Bảng thẻ ký túc xá (Dormitory\_Card)

- number: Mã số thẻ ký túc xá, khoá chính.
- start\_date: Ngày bắt đầu hiệu lực thẻ.
- end\_date: Ngày kết thúc hiệu lực thẻ.
- validity: Trạng thái thẻ (còn hiệu lực hay không).
- id\_card: Mã căn cước sinh viên (tham chiếu đến bảng identity\_card).

```
1 CREATE TABLE dormitory_card (
2     number CHAR(7) NOT NULL PRIMARY KEY,
3     start_date DATE,
4     end_date DATE,
```

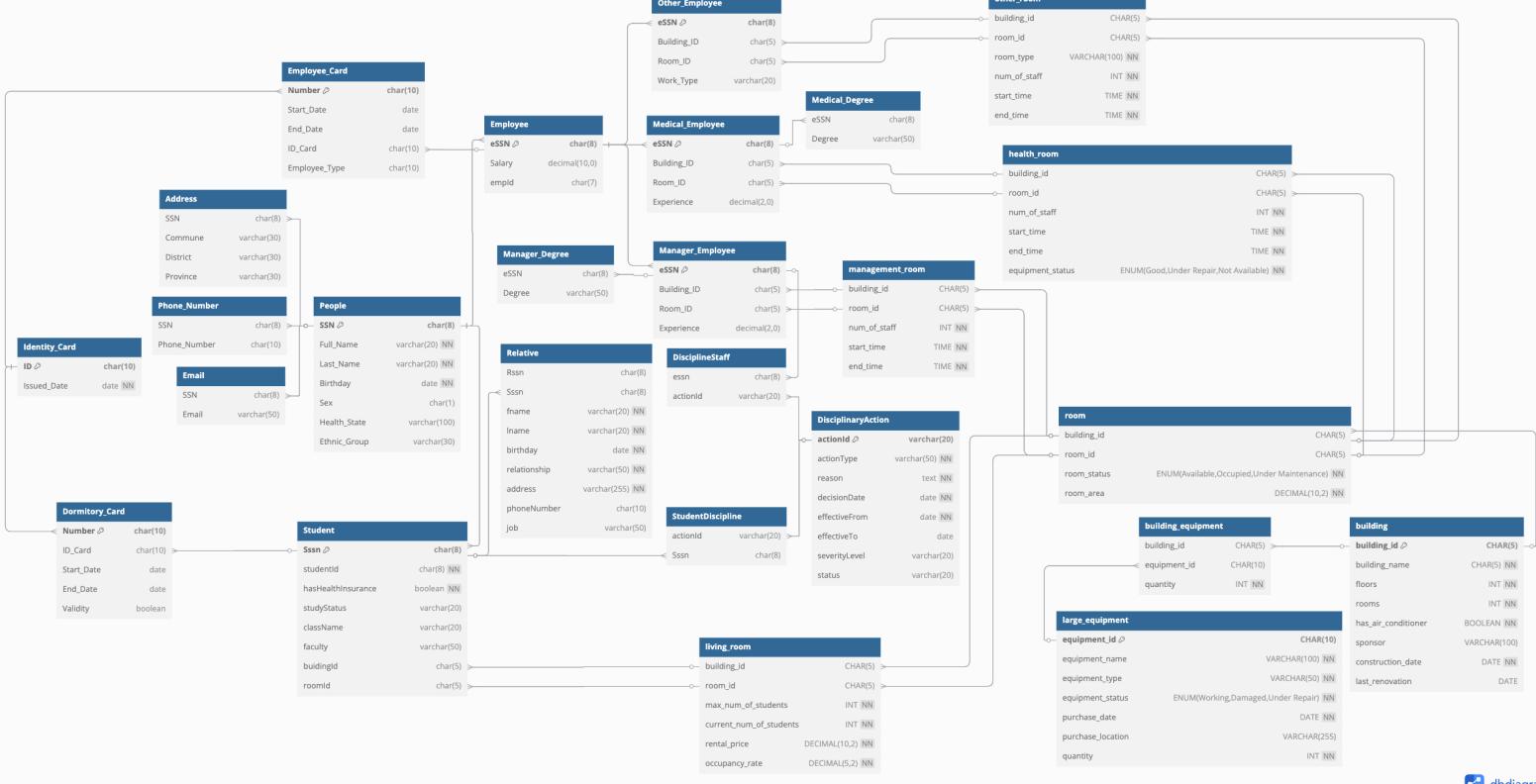


```
5      CHECK (start_date <= end_date),
6      id_card CHAR(8) NOT NULL,
7      validity BOOLEAN DEFAULT TRUE,
8      FOREIGN KEY (number) REFERENCES identity_card(id)
9          ON UPDATE CASCADE ON DELETE CASCADE,
10     FOREIGN KEY (id_card) REFERENCES student(ssn)
11         ON UPDATE CASCADE ON DELETE CASCADE
12 );
```

### 2.1.27 Bảng thẻ nhân viên (Employee\_Card)

- **number:** Mã số thẻ nhân viên, khoá chính.
- **start\_date:** Ngày bắt đầu hiệu lực thẻ nhân viên.
- **end\_date:** Ngày kết thúc hiệu lực thẻ nhân viên.
- **employee\_type:** Loại nhân viên (Janitor, Security, Reception).
- **id\_card:** Mã căn cước nhân viên (tham chiếu đến bảng identity\_card).

```
1 CREATE TABLE employee_card (
2     number CHAR(7) NOT NULL PRIMARY KEY,
3     start_date DATE,
4     end_date DATE,
5     CHECK (start_date <= end_date),
6     id_card CHAR(8) NOT NULL,
7     employee_type CHAR(10) CHECK (
8         employee_type IN ('Janitor', 'Security', 'Reception')
9     ),
10    FOREIGN KEY (number) REFERENCES identity_card(id)
11        ON UPDATE CASCADE ON DELETE CASCADE,
12    FOREIGN KEY (id_card) REFERENCES employee(essn)
13        ON UPDATE CASCADE ON DELETE CASCADE
14 );
```



Hình 2.1: Biểu đồ EERD cho hệ thống cơ sở dữ liệu ký túc xá

## 2.2 Thêm dữ liệu mẫu

### 2.2.1 People

```

1  INSERT INTO people VALUE
2  ('05620501', 'An', 'Nguyen Van', '1992-11-1', 'M', 'Good', 'Kinh'),
3  ('05620502', 'Anh', 'Nguyen Thi', '1996-10-1', 'F', 'Good', 'Kinh'),
4  ('05620503', 'Binh', 'Nguyen An', '1990-12-31', 'M', 'Good', 'Kinh'),
5  ('05620504', 'Tuan', 'Trinh Tran Phuong', '1997-4-12', 'M', 'Good', 'Kinh'),
6  ('05620505', 'Hoang', 'Dang Tien', '1988-5-8', 'M', 'Good', 'Kinh'),
7  ('05620506', 'Dat', 'Nguyen Tan', '1995-4-17', 'M', 'Good', 'Kinh'),
8  ('05620507', 'Bao', 'Nguyen Van', '1996-1-1', 'M', 'Good', 'Hmong'),
9  ('05620508', 'Hung', 'Tran', '1989-8-1', 'M', 'Good', 'Kinh'),
10 ('05620509', 'Huyen', 'Nguyen Ngoc', '1988-2-6', 'F', 'Good', 'Kinh'),
11 ('05620510', 'Hoa', 'Bui Ngoc', '1980-5-1', 'M', 'Good', 'Kinh'),
12 ('05620511', 'Hieu', 'Tran Minh', '1999-10-30', 'M', 'Good', 'Kinh'),

```



13 ('05620512', 'Khanh', 'Tran Gia', '1988-9-1', 'M', 'Good', 'Kinh'),  
14 ('05620513', 'Khoi', 'Nguyen Minh', '2003-6-3', 'M', 'Good', 'Kinh'),  
15 ('05620514', 'Mai', 'Nguyen Phuong', '2006-3-9', 'F', 'Good', 'Kinh'),  
16 ('05620515', 'Ngoc', 'Nguyen Minh', '2003-12-1', 'F', 'Good', 'Kinh'),  
17 ('05620516', 'Nam', 'Nguyen Hao', '2006-3-2', 'M', 'Good', 'Kinh'),  
18 ('05620517', 'Ngan', 'Pham Kim', '2003-10-20', 'F', 'Good', 'Kinh'),  
19 ('05620518', 'Phuong', 'Tran Ngoc', '2005-3-26', 'F', 'Good', 'Kinh'),  
20 ('05620519', 'Phu', 'Nguyen Quoc', '2005-7-23', 'M', 'Good', 'Kinh'),  
21 ('05620520', 'Phung', 'Nguyen Minh', '2004-5-1', 'F', 'Good', 'Kinh'),  
22 ('05620521', 'Quan', 'Vo Anh', '2005-9-6', 'M', 'Good', 'Kinh'),  
23 ('05620522', 'Phuoc', 'Nguyen Thien', '2005-10-18', 'M', 'Good', 'Kinh'),  
24 ('05620523', 'Quynh', 'Nguyen Ngoc Diem', '2006-2-1', 'F', 'Good', 'Kinh'),  
25 ('05620524', 'Quynh', 'Nguyen Thi Diem', '2004-5-1', 'F', 'Good', 'Kinh'),  
26 ('05620525', 'Phat', 'Vo Tan', '2004-5-1', 'M', 'Good', 'Kinh'),  
27 ('05620526', 'Son', 'Bui Ngoc', '2003-12-12', 'M', 'Good', 'Kinh'),  
28 ('05620527', 'Tai', 'Nguyen Duc', '2003-9-30', 'M', 'Good', 'Thai'),  
29 ('05620528', 'Tan', 'Nguyen Nhat', '2004-1-31', 'M', 'Good', 'Kinh'),  
30 ('05620529', 'Tien', 'Phan Ngoc', '2005-9-17', 'M', 'Good', 'Kinh'),  
31 ('05620530', 'Tu', 'Nguyen Anh', '2003-2-12', 'M', 'Good', 'Kinh'),  
32 ('05620531', 'Tuong', 'Nguyen Ngoc', '2005-3-14', 'M', 'Good', 'Kinh'),  
33 ('05620532', 'Thao', 'Nguyen Ngoc Thanh', '2006-2-4', 'F', 'Good', 'Kinh'),  
34 ('05620533', 'Thu', 'Nguyen Ngoc Minh', '2005-3-31', 'F', 'Good', 'Kinh'),  
35 ('05620534', 'Thuy', 'Nguyen Thi Thanh', '2003-2-15', 'F', 'Good', 'Kinh'),  
36 ('05620535', 'Uyen', 'Nguyen Ngoc', '2005-2-18', 'M', 'Good', 'Kinh'),  
37 ('05620536', 'Vy', 'Nguyen Tuong', '2004-6-13', 'M', 'Good', 'Kinh'),  
38 ('05620537', 'Vu', 'Tran Van', '2003-2-12', 'M', 'Good', 'Kinh'),  
39 ('05620538', 'Duc', 'Nguyen Minh', '2003-3-15', 'M', 'Good', 'Tay'),  
40 ('05620539', 'Toan', 'Bui Duc', '2006-1-16', 'M', 'Good', 'Kinh'),  
41 ('05620540', 'Tram', 'Nguyen Thuy', '2004-8-1', 'F', 'Good', 'Kinh'),  
42 ('05620541', 'Truc', 'Pham Ngoc', '2006-1-24', 'M', 'Good', 'Kinh'),  
43 ('05620542', 'Hau', 'Nguyen Phuc', '2005-6-1', 'F', 'Good', 'Kinh'),  
44 ('05620543', 'Thuan', 'Luong Minh', '2003-12-5', 'M', 'Good', 'Kinh'),  
45 ('05620544', 'Nhi', 'Nguyen Ngoc Thuy', '2006-3-20', 'F', 'Good', 'Kinh'),  
46 ('05620545', 'Dai', 'Nguyen Van', '2005-2-17', 'M', 'Good', 'Kinh'),  
47 ('05620546', 'Khiem', 'Pham Gia', '2003-9-23', 'M', 'Good', 'Kinh'),  
48 ('05620547', 'Tran', 'Nguyen Ngoc Thao', '2005-5-29', 'F', 'Good', 'Kinh'),  
49 ('05620548', 'Tien', 'Nguyen Thuy', '2004-3-19', 'F', 'Good', 'Kinh'),  
50 ('05620549', 'Tuan', 'Nguyen Quoc', '2005-9-12', 'M', 'Good', 'Kinh'),



```

51 ('05620550', 'Chi', 'Bui Ngoc Kim', '2003-9-15', 'F', 'Good', 'Hoa'),
52 ('05620551', 'Chi', 'Pham Minh', '2006-4-1', 'M', 'Good', 'Kinh'),
53 ('05620552', 'Chien', 'Tran Duc', '2002-1-26', 'M', 'Good', 'Kinh');

```

## 2.2.2 Bảng phone\_number

```

1 INSERT INTO phone_number VALUES
2 ('05620501', '0328100000'), ('05620501', '0328100543'), ('05620502', '0328100001'),
3 ('05620503', '0328100002'), ('05620503', '0328234213'), ('05620504', '0328100003'),
4 ('05620505', '0328100004'), ('05620506', '0328100005'), ('05620507', '0328100245'),
5 ('05620507', '0322452343'), ('05620508', '0328105631'), ('05620509', '0323453455'),
6 ('05620510', '0337493258'), ('05620511', '0329798089'), ('05620511', '0354234349'),
7 ('05620512', '0334574368'), ('05620513', '0389162347'), ('05620514', '0328190284'),
8 ('05620515', '0323824785'), ('05620516', '0309238478'), ('05620516', '0302455638'),
9 ('05620517', '0320245757'), ('05620518', '0329287547'), ('05620518', '0329309458'),
10 ('05620519', '0320918267'), ('05620520', '0387326653'), ('05620521', '0320975643'),
11 ('05620521', '0329235432'), ('05620522', '0399706545'), ('05620523', '0328100038'),
12 ('05620524', '0391203470'), ('05620525', '0328906742'), ('05620525', '0392348324'),
13 ('05620526', '0328109200'), ('05620527', '0328102980'), ('05620527', '0328239804'),
14 ('05620528', '0322134783'), ('05620529', '0320975423'), ('05620530', '0328100347'),
15 ('05620531', '0326784592'), ('05620531', '0322349823'), ('05620532', '0326735204'),
16 ('05620533', '0327843792'), ('05620534', '0328129343'), ('05620534', '0339852345'),
17 ('05620535', '0393280450'), ('05620536', '0924723455'), ('05620536', '0959873235'),
18 ('05620537', '0739012012'), ('05620538', '0928761451'), ('05620539', '0322934734'),
19 ('05620540', '0329235483'), ('05620540', '0325359783'), ('05620541', '0328198534'),
20 ('05620542', '0327684529'), ('05620543', '0320394754'), ('05620544', '0309237432'),
21 ('05620544', '0329483452'), ('05620545', '0320293745'), ('05620546', '0302348523'),
22 ('05620546', '0339854982'), ('05620547', '0920975343'), ('05620548', '0322934523'),
23 ('05620548', '0339845723'), ('05620549', '0328102409'), ('05620550', '0328724234'),
24 ('05620550', '0339785329'), ('05620551', '0329235798'), ('05620552', '0329459838');
```

## 2.2.3 Bảng address

```

1 INSERT INTO address VALUES
2 ('05620501', 'Dien Lam', 'Dien Khanh', 'Khanh Hoa'),
3 ('05620501', 'Nam Dong', 'Nam Dong', 'Thua Thien Hue'),
4 ('05620502', 'Phu Tan', 'Tan Phu', 'Dong Nai'),

```



5 ('05620503', 'Loc An', 'Dat Do', 'Ba Ria - Vung Tau'),  
6 ('05620504', 'An Binh', 'Long Thanh', 'Dong Nai'),  
7 ('05620505', 'Tan Xuan', 'Dong Xuan', 'Phu Yen'),  
8 ('05620506', 'Tuy An', 'Tuy Hoa', 'Phu Yen'),  
9 ('05620507', 'Hai Van', 'Hai Chau', 'Da Nang'),  
10 ('05620508', 'Hoa Khanh', 'Lien Chieu', 'Da Nang'),  
11 ('05620509', 'Dong Giang', 'Tay Giang', 'Quang Nam'),  
12 ('05620510', 'Huong An', 'Que Son', 'Quang Nam'),  
13 ('05620511', 'Nong Son', 'Nong Son', 'Quang Nam'),  
14 ('05620512', 'Binh Nam', 'Thang Binh', 'Quang Nam'),  
15 ('05620513', 'Tam Dan', 'Phu Ninh', 'Quang Nam'),  
16 ('05620514', 'Son Tra', 'Son Tra', 'Da Nang'),  
17 ('05620515', 'Cam Le', 'Cam Le', 'Da Nang'),  
18 ('05620516', 'Thanh Khe', 'Thanh Khe', 'Da Nang'),  
19 ('05620517', 'Hoa Vang', 'Hoa Vang', 'Da Nang'),  
20 ('05620518', 'Hoa An', 'Lien Chieu', 'Da Nang'),  
21 ('05620518', 'Thanh Khe', 'Thanh Khe', 'Da Nang'),  
22 ('05620519', 'Cau Moi', 'Ha Long', 'Quang Ninh'),  
23 ('05620520', 'Bai Chay', 'Ha Long', 'Quang Ninh'),  
24 ('05620521', 'Quang Trung', 'Quang Yen', 'Quang Ninh'),  
25 ('05620522', 'Yen Thanh', 'Yen Hung', 'Quang Ninh'),  
26 ('05620522', 'Nam Dong', 'Nam Dong', 'Thua Thien Hue'),  
27 ('05620523', 'Mong Duong', 'Cam Pha', 'Quang Ninh'),  
28 ('05620524', 'Uong Bi', 'Uong Bi', 'Quang Ninh'),  
29 ('05620525', 'Dong Tien', 'Luong Son', 'Hoa Binh'),  
30 ('05620526', 'Mai Chau', 'Mai Chau', 'Hoa Binh'),  
31 ('05620527', 'Dinh Hoa', 'Dinh Hoa', 'Thai Nguyen'),  
32 ('05620528', 'Son Cam', 'Thai Nguyen', 'Thai Nguyen'),  
33 ('05620529', 'Hoa Binh', 'Hoa Binh', 'Hoa Binh'),  
34 ('05620530', 'Lang Son', 'Lang Son', 'Lang Son'),  
35 ('05620531', 'Phu Tan', 'Tan Phu', 'Dong Nai'),  
36 ('05620532', 'Bac Giang', 'Bac Giang', 'Bac Giang'),  
37 ('05620533', 'Tuy An', 'Tuy Hoa', 'Phu Yen'),  
38 ('05620533', 'Tan Xuan', 'Dong Xuan', 'Phu Yen'),  
39 ('05620534', 'Phu Ly', 'Phu Ly', 'Ha Nam'),  
40 ('05620535', 'Duc Tho', 'Duc Tho', 'Ha Tinh'),  
41 ('05620536', 'Quynh Luu', 'Quynh Luu', 'Nghe An'),  
42 ('05620537', 'Vinh', 'Vinh', 'Nghe An'),



```
43 ('05620537', 'Dong Tien', 'Luong Son', 'Hoa Binh'),  
44 ('05620538', 'Nam Dan', 'Nam Dan', 'Nghe An'),  
45 ('05620539', 'Thanh Chuong', 'Thanh Chuong', 'Nghe An'),  
46 ('05620540', 'Cua Lo', 'Cua Lo', 'Nghe An'),  
47 ('05620541', 'Hung Nguyen', 'Hung Nguyen', 'Nghe An'),  
48 ('05620542', 'Kien Thuy', 'Kien Thuy', 'Hai Phong'),  
49 ('05620543', 'Ngo Quyen', 'Ngo Quyen', 'Hai Phong'),  
50 ('05620544', 'An Lao', 'An Lao', 'Hai Phong'),  
51 ('05620545', 'Van Lam', 'Van Lam', 'Hung Yen'),  
52 ('05620545', 'Yen Thanh', 'Yen Hung', 'Quang Ninh'),  
53 ('05620546', 'Cua Lo', 'Cua Lo', 'Nghe An'),  
54 ('05620546', 'Phan Thiet', 'Phan Thiet', 'Binh Thuan'),  
55 ('05620547', 'Tan Nghia', 'Tan Nghia', 'Binh Thuan'),  
56 ('05620547', 'Hung Nguyen', 'Hung Nguyen', 'Nghe An'),  
57 ('05620548', 'Dong Ha', 'Dong Ha', 'Quang Tri'),  
58 ('05620549', 'Huong Thuy', 'Huong Thuy', 'Thua Thien Hue'),  
59 ('05620550', 'Phu Bai', 'Phu Bai', 'Thua Thien Hue'),  
60 ('05620551', 'Nam Dong', 'Nam Dong', 'Thua Thien Hue'),  
61 ('05620552', 'Huong Tra', 'Huong Tra', 'Thua Thien Hue');
```

#### 2.2.4 Bảng email

```
1 INSERT INTO email VALUES  
2 ('05620501', 'annguyen01@gmail.com'),  
3 ('05620502', 'anhnguyen02@gmail.com'),  
4 ('05620503', 'binhnguyen03@gmail.com'),  
5 ('05620504', 'tuantrinh04@gmail.com'),  
6 ('05620504', 'tuantrinhtran03@gmail.com'),  
7 ('05620505', 'hoangdang05@gmail.com'),  
8 ('05620506', 'datnguyen06@gmail.com'),  
9 ('05620507', 'baonguyen07@gmail.com'),  
10 ('05620508', 'hungtran08@gmail.com'),  
11 ('05620509', 'huyennguyen09@gmail.com'),  
12 ('05620510', 'hoabui10@gmail.com'),  
13 ('05620511', 'hieutran11@gmail.com'),  
14 ('05620512', 'khanhtran12@gmail.com'),  
15 ('05620513', 'khoi.nguyenminh03@hcmut.edu.vn'),  
16 ('05620513', 'khoinguyen13@gmail.com'),
```



```
17 ('05620514', 'mai.nguyenphuong06@hcmut.edu.vn'),  
18 ('05620515', 'ngoc.nguyenminh03@hcmut.edu.vn'),  
19 ('05620516', 'nam.nguyenhao06@hcmut.edu.vn'),  
20 ('05620516', 'namhao987@gmail.com'),  
21 ('05620517', 'ngan.phamkim03@hcmut.edu.vn'),  
22 ('05620518', 'phuong.trangoc05@hcmut.edu.vn'),  
23 ('05620519', 'phu.nguyenquoc05@hcmut.edu.vn'),  
24 ('05620520', 'phung.nguyenminh04@hcmut.edu.vn'),  
25 ('05620521', 'quan.voanh05@hcmut.edu.vn'),  
26 ('05620521', 'quanvo05@gmail.com'),  
27 ('05620522', 'phuoc.nguyenthien05@hcmut.edu.vn'),  
28 ('05620523', 'quynh.nguyenngocdiem06@hcmut.edu.vn'),  
29 ('05620524', 'quynh.nguyenthidiem04@hcmut.edu.vn'),  
30 ('05620525', 'phat.vo10@hcmut.edu.vn'),  
31 ('05620525', 'phatvo@gmail.com'),  
32 ('05620526', 'son.buingoc03@hcmut.edu.vn'),  
33 ('05620527', 'tai.nguyenduc03@hcmut.edu.vn'),  
34 ('05620528', 'tan.nguyennhat04@hcmut.edu.vn'),  
35 ('05620528', 'tannhat@gmail.com'),  
36 ('05620529', 'tien.phanngoc05@hcmut.edu.vn'),  
37 ('05620529', 'tienphan@gmail.com'),  
38 ('05620530', 'tu.nguyenanh03@hcmut.edu.vn'),  
39 ('05620531', 'tuong.nguyenngoc05@hcmut.edu.vn'),  
40 ('05620532', 'thao.nguyenengocthanh06@hcmut.edu.vn'),  
41 ('05620533', 'thu.nguyenngocminh05@hcmut.edu.vn'),  
42 ('05620533', 'thuminhnguyen123@gmail.com'),  
43 ('05620534', 'thuy.nguyenthithanh03@hcmut.edu.vn'),  
44 ('05620535', 'uyen.nguyenngoc05@hcmut.edu.vn'),  
45 ('05620536', 'vy.nguyenthuong04@hcmut.edu.vn'),  
46 ('05620537', 'vu.tranvan03@hcmut.edu.vn'),  
47 ('05620538', 'duc.nguyenminh03@hcmut.edu.vn'),  
48 ('05620539', 'toan.buiduc06@hcmut.edu.vn'),  
49 ('05620539', 'toanducr@gmail.com'),  
50 ('05620540', 'tram.nguyenthuy04@hcmut.edu.vn'),  
51 ('05620541', 'truc.phamngoc06@hcmut.edu.vn'),  
52 ('05620542', 'hau.nguyenphuc05@hcmut.edu.vn'),  
53 ('05620543', 'thuan.luongminh03@hcmut.edu.vn'),  
54 ('05620544', 'nhi.nguyenngocthuy06@hcmut.edu.vn'),
```



```

55 ('05620545','dai.nguyenvan05@hcmut.edu.vn'),
56 ('05620545','daivan@gmail.com'),
57 ('05620546','khiem.phamgia03@hcmut.edu.vn'),
58 ('05620547','tran.nguyenngocthao05@hcmut.edu.vn'),
59 ('05620548','tien.nguyenthuy04@hcmut.edu.vn'),
60 ('05620549','tuan.nguyenquoc05@hcmut.edu.vn'),
61 ('05620549','tuannguyen@gmail.com'),
62 ('05620550','chi.buingockim03@hcmut.edu.vn'),
63 ('05620551','chi.phamminh06@hcmut.edu.vn'),
64 ('05620551','chipham00@gmail.com'),
65 ('05620552','chien.tranduc02@hcmut.edu.vn');

```

## 2.2.5 Bảng building

```

1 INSERT INTO building
2 VALUES
3 ('BK001','BK001',4, 16, TRUE, 'HCMUT', '2015-01-01', '2022-06-01'),
4 ('BK002','BK002',4, 16, FALSE, 'HCMUT', '2016-03-15', '2023-02-10'),
5 ('BK003','BK003',4, 16, TRUE, 'HCMUT', '2017-05-10', '2021-09-20'),
6 ('BK004','BK004',4, 16, TRUE, 'HCMUT', '2018-07-25', NULL);

```

## 2.2.6 Bảng room

```

1 INSERT INTO room VALUES
2 -- BK001's rooms
3 ('BK001','P.101','Available',25.0),('BK001','P.102','Available',25.0),
4 ('BK001','P.103','Available',25.0),('BK001','P.104','Available',25.0),
5 ('BK001','P.201','Occupied',25.0),('BK001','P.202','Occupied',25.0),
6 ('BK001','P.203','Occupied',25.0),('BK001','P.204','Occupied',25.0),
7 ('BK001','P.301','Under Maintenance',25.0),('BK001','P.302','Available',25.0),
8 ('BK001','P.303','Available',25.0),('BK001','P.304','Available',25.0),
9 ('BK001','P.401','Available',25.0),('BK001','P.402','Occupied',25.0),
10 ('BK001','P.403','Occupied',25.0),('BK001','P.404','Available',25.0),
11 -- BK002's rooms
12 ('BK002','P.101','Available',25.0),('BK002','P.102','Available',25.0),
13 ('BK002','P.103','Available',25.0),('BK002','P.104','Available',25.0),
14 ('BK002','P.201','Occupied',25.0),('BK002','P.202','Occupied',25.0),

```



```
15 ('BK002', 'P.203', 'Occupied', 25.0), ('BK002', 'P.204', 'Occupied', 25.0),
16 ('BK002', 'P.301', 'Under Maintenance', 25.0), ('BK002', 'P.302', 'Available', 25.0),
17 ('BK002', 'P.303', 'Available', 25.0), ('BK002', 'P.304', 'Available', 25.0),
18 ('BK002', 'P.401', 'Available', 25.0), ('BK002', 'P.402', 'Occupied', 25.0),
19 ('BK002', 'P.403', 'Occupied', 25.0), ('BK002', 'P.404', 'Available', 25.0),
20 -- BK003's rooms
21 ('BK003', 'P.101', 'Available', 25.0), ('BK003', 'P.102', 'Available', 25.0),
22 ('BK003', 'P.103', 'Available', 25.0), ('BK003', 'P.104', 'Available', 25.0),
23 ('BK003', 'P.201', 'Occupied', 25.0), ('BK003', 'P.202', 'Occupied', 25.0),
24 ('BK003', 'P.203', 'Occupied', 25.0), ('BK003', 'P.204', 'Occupied', 25.0),
25 ('BK003', 'P.301', 'Under Maintenance', 25.0), ('BK003', 'P.302', 'Available', 25.0),
26 ('BK003', 'P.303', 'Available', 25.0), ('BK003', 'P.304', 'Available', 25.0),
27 ('BK003', 'P.401', 'Available', 25.0), ('BK003', 'P.402', 'Occupied', 25.0),
28 ('BK003', 'P.403', 'Occupied', 25.0), ('BK003', 'P.404', 'Available', 25.0),
29 -- BK004's rooms
30 ('BK004', 'P.101', 'Available', 25.0), ('BK004', 'P.102', 'Available', 25.0),
31 ('BK004', 'P.103', 'Available', 25.0), ('BK004', 'P.104', 'Available', 25.0),
32 ('BK004', 'P.201', 'Occupied', 25.0), ('BK004', 'P.202', 'Occupied', 25.0),
33 ('BK004', 'P.203', 'Occupied', 25.0), ('BK004', 'P.204', 'Occupied', 25.0),
34 ('BK004', 'P.301', 'Under Maintenance', 25.0), ('BK004', 'P.302', 'Available', 25.0),
35 ('BK004', 'P.303', 'Available', 25.0), ('BK004', 'P.304', 'Available', 25.0),
36 ('BK004', 'P.401', 'Available', 25.0), ('BK004', 'P.402', 'Occupied', 25.0),
37 ('BK004', 'P.403', 'Occupied', 25.0), ('BK004', 'P.404', 'Available', 25.0);
```

## 2.2.7 Bảng employee

```
1 INSERT INTO employee (essn, salary) VALUES
2 ('05620501',10000000), ('05620502',9000000), ('05620503',11000000),
3 ('05620504',30000), ('05620505',30000), ('05620506',30000),
4 ('05620507',8000000), ('05620508',15000000), ('05620509',14000000),
5 ('05620510',20000000), ('05620511',12000000), ('05620512',31000);
```

## 2.2.8 Bảng living\_room

```
1 INSERT INTO living_room VALUES
2 -- BK001's rooms
3 ('BK001', 'P.104', 6, 1500000.00), ('BK001', 'P.201', 6, 1500000.00),
```



```

4 ('BK001', 'P.202', 6, 1500000.00), ('BK001', 'P.203', 6, 1500000.00),
5 ('BK001', 'P.204', 6, 1500000.00), ('BK001', 'P.301', 6, 1500000.00),
6 ('BK001', 'P.302', 6, 1500000.00), ('BK001', 'P.303', 6, 1500000.00),
7 ('BK001', 'P.304', 6, 1500000.00), ('BK001', 'P.401', 6, 1500000.00),
8 ('BK001', 'P.402', 6, 1500000.00), ('BK001', 'P.403', 6, 1500000.00),
9 ('BK001', 'P.404', 6, 1500000.00),
10 -- BK002's rooms
11 ('BK002', 'P.102', 6, 1500000.00), ('BK002', 'P.104', 6, 1500000.00),
12 ('BK002', 'P.201', 6, 1500000.00), ('BK002', 'P.202', 6, 1500000.00),
13 ('BK002', 'P.203', 6, 1500000.00), ('BK002', 'P.204', 6, 1500000.00),
14 ('BK002', 'P.301', 6, 1500000.00), ('BK002', 'P.302', 6, 1500000.00),
15 ('BK002', 'P.303', 6, 1500000.00), ('BK002', 'P.304', 6, 1500000.00),
16 ('BK002', 'P.401', 6, 1500000.00), ('BK002', 'P.402', 6, 1500000.00),
17 ('BK002', 'P.403', 6, 1500000.00), ('BK002', 'P.404', 6, 1500000.00),
18 -- BK003's rooms
19 ('BK003', 'P.102', 6, 1500000.00), ('BK003', 'P.104', 6, 1500000.00),
20 ('BK003', 'P.201', 6, 1500000.00), ('BK003', 'P.202', 6, 1500000.00),
21 ('BK003', 'P.203', 6, 1500000.00), ('BK003', 'P.204', 6, 1500000.00),
22 ('BK003', 'P.301', 6, 1500000.00), ('BK003', 'P.302', 6, 1500000.00),
23 ('BK003', 'P.303', 6, 1500000.00), ('BK003', 'P.304', 6, 1500000.00),
24 ('BK003', 'P.401', 6, 1500000.00), ('BK003', 'P.402', 6, 1500000.00),
25 ('BK003', 'P.403', 6, 1500000.00), ('BK003', 'P.404', 6, 1500000.00),
26 -- BK004's rooms
27 ('BK004', 'P.102', 6, 1500000.00), ('BK004', 'P.104', 6, 1500000.00),
28 ('BK004', 'P.201', 6, 1500000.00), ('BK004', 'P.202', 6, 1500000.00),
29 ('BK004', 'P.203', 6, 1500000.00), ('BK004', 'P.204', 6, 1500000.00),
30 ('BK004', 'P.301', 6, 1500000.00), ('BK004', 'P.302', 6, 1500000.00),
31 ('BK004', 'P.303', 6, 1500000.00), ('BK004', 'P.304', 6, 1500000.00),
32 ('BK004', 'P.401', 6, 1500000.00), ('BK004', 'P.402', 6, 1500000.00),
33 ('BK004', 'P.403', 6, 1500000.00), ('BK004', 'P.404', 6, 1500000.00);

```

## 2.2.9 Bảng student

```

1 INSERT INTO student VALUES
2 ('05620513', '2312613', true, 'Active', 'KHMT1', 'Computer Science', 'BK001', 'P.104')
3 ,
4 ('05620514', '2212345', true, 'Active', 'KHMT2', 'Computer Science', 'BK001', 'P.104')
5 ,

```



```
4 ('05620515','2012346',true,'Active','CNTT1','Information Technology','BK003','P
.204'),
5 ('05620516','2112347',true,'Active','CNTT2','Information Technology','BK001','P
.104'),
6 ('05620517','2312348',true,'Active','DTVT1','Electronics and Telecommunications
Engineering','BK002','P.102'),
7 ('05620518','2412349',true,'Active','DTVT2','Electronics and Telecommunications
Engineering','BK002','P.104'),
8 ('05620519','2212350',true,'Active','CK1','Mechanical Engineering','BK002','P
.203'),
9 ('05620520','2012351',true,'Non_Active','CK2','Mechanical Engineering','BK002',
'P.102'),
10 ('05620521','2112352',true,'Active','D1','Electrical Engineering','BK003','P
.104'),
11 ('05620522','2312353',true,'Active','D2','Electrical Engineering','BK003','P
.104'),
12 ('05620523','2412354',true,'Active','HTTT1','Information Security','BK003','P
.104'),
13 ('05620524','2212355',true,'Active','HTTT2','Information Security','BK003','P
.104'),
14 ('05620525','2012356',true,'Active','XD1','Civil Engineering','BK003','P.104'),
15 ('05620526','2112357',true,'Active','XD2','Civil Engineering','BK003','P.104'),
16 ('05620527','2312358',true,'Active','H1','Chemical Engineering','BK004','P.104
'),
17 ('05620528','2412359',true,'Active','H2','Chemical Engineering','BK004','P.104
'),
18 ('05620529','2212360',true,'Non_Active','MT1','Environmental Engineering','
BK001','P.204'),
19 ('05620530','2012361',true,'Active','MT2','Environmental Engineering','BK001','
P.204'),
20 ('05620531','2112362',true,'Active','DK1','Control and Automation Engineering',
'BK001','P.204'),
21 ('05620532','2312363',true,'Active','DK2','Control and Automation Engineering',
'BK001','P.204'),
22 ('05620533','2412364',true,'Active','QL1','Industrial Management','BK002','P
.204'),
23 ('05620534','2212365',true,'Active','QL2','Industrial Management','BK002','P
.204'),
```



```

24 ('05620535','2012366',true,'Active','SCM1','Logistics and Supply Chain
     Management','BK002','P.204'),
25 ('05620536','2112367',true,'Active','SCM2','Logistics and Supply Chain
     Management','BK002','P.204'),
26 ('05620537','2312368',true,'Active','VL1','Materials Engineering','BK003','P
     .204'),
27 ('05620538','2412369',true,'Active','VL2','Materials Engineering','BK003','P
     .204'),
28 ('05620539','2212370',true,'Active','SH1','Biotechnology','BK003','P.204'),
29 ('05620540','2012371',true,'Non_Active','SH2','Biotechnology','BK003','P.204'),
30 ('05620541','2412374',true,'Active','KT1','Computer Engineering','BK004','P.204
     '),
31 ('05620542','2212375',true,'Active','KT2','Computer Engineering','BK004','P.204
     '),
32 ('05620543','2112382',true,'Active','XD3','Civil Engineering','BK004','P.204'),
33 ('05620544','2312383',true,'Active','H3','Chemical Engineering','BK004','P.204
     '),
34 ('05620545','2012376',true,'Active','KHMT3','Computer Science','BK001','P.304')
     ,
35 ('05620546','2112377',true,'Active','CNTT3','Information Technology','BK001','P
     .304'),
36 ('05620547','2312378',true,'Active','DTVT3','Electronics and Telecommunications
     Engineering','BK001','P.304'),
37 ('05620548','2412379',true,'Active','CK3','Mechanical Engineering','BK001','P
     .304'),
38 ('05620549','2212380',true,'Active','D3','Electrical Engineering','BK003','P
     .304'),
39 ('05620550','2012381',true,'Active','HTTT3','Information Security','BK003','P
     .304'),
40 ('05620551','2112372',true,'Active','PM1','Software Engineering','BK003','P.304
     '),
41 ('05620552','2312373',true,'Active','PM2','Software Engineering','BK003','P.304
     ');

```

### 2.2.10 Bảng management\_room

```

1 INSERT INTO management_room (building_id, room_id, start_time, end_time)
2 VALUES

```



```
3 ('BK001', 'P.101', '08:00:00', '17:00:00'), ('BK002', 'P.101', '08:00:00', '17:00:00')  
    ,  
4 ('BK003', 'P.101', '08:00:00', '17:00:00'), ('BK004', 'P.101', '08:00:00', '17:00:00')  
    ;
```

### 2.2.11 Bảng manager\_employee

```
1 INSERT INTO manager_employee VALUES  
2 ('05620501', 'P.101', 'BK001', 5), ('05620502', 'P.101', 'BK002', 3),  
3 ('05620503', 'P.101', 'BK003', 10), ('05620507', 'P.101', 'BK004', 2);
```

### 2.2.12 Bảng manager\_degree

```
1 INSERT INTO manager_degree VALUES  
2 ('05620501', 'Bachelor of Human Resource Management'),  
3 ('05620501', 'Communication and Soft Skills Certificate'),  
4 ('05620502', 'Bachelor of Administrative Management'),  
5 ('05620503', 'Facility Management Certificate'),  
6 ('05620503', 'Fire Safety Certificate'),  
7 ('05620503', 'Bachelor of Administrative Management'),  
8 ('05620507', 'Bachelor of Administrative Management');
```

### 2.2.13 Bảng health\_room

```
1 INSERT INTO health_room (building_id, room_id, start_time, end_time,  
    equipment_status)  
VALUES  
3 ('BK001', 'P.102', '08:00:00', '18:00:00', 'Good');
```

### 2.2.14 Bảng medical\_employee

```
1 INSERT INTO medical_employee VALUES  
2 ('05620508', 'P.102', 'BK001', 10), ('05620509', 'P.102', 'BK001', 20),  
3 ('05620510', 'P.102', 'BK001', 15), ('05620511', 'P.102', 'BK001', 17);
```



### 2.2.15 Bảng medical\_degree

```
1 INSERT INTO medical_degree VALUES
2 ('05620508', 'Bachelor of Nursing'),
3 ('05620509', 'Intermediate Diploma in Medicine'),
4 ('05620509', 'First Aid Certification'),
5 ('05620510', 'School Health Management Certificate'),
6 ('05620511', 'First Aid Certification'),
7 ('05620511', 'Bachelor of Nursing');
```

### 2.2.16 Bảng other\_room

```
1 INSERT INTO other_room (building_id, room_id, room_type, start_time, end_time)
2 VALUES
3 ('BK001', 'P.103', 'Meeting Room', '09:00:00', '17:00:00'),
4 ('BK002', 'P.103', 'Meeting Room', '09:00:00', '17:00:00'),
5 ('BK003', 'P.103', 'Meeting Room', '09:00:00', '17:00:00'),
6 ('BK004', 'P.103', 'Meeting Room', '09:00:00', '17:00:00');
```

### 2.2.17 Bảng other\_employee

```
1 INSERT INTO other_employee VALUES
2 ('05620504', 'P.103', 'BK002', 'Janitor'), ('05620505', 'P.103', 'BK003', 'Security'),
3 ('05620506', 'P.103', 'BK001', 'Reception'), ('05620512', 'P.103', 'BK004', 'Security')
4 );
```

### 2.2.18 Bảng other\_employee\_time\_work

```
1 INSERT INTO other_employee_time_work VALUES
2 ('05620504', 'monday', '06:00:00', '18:00:00'),
3 ('05620504', 'tuesday', '18:00:00', '06:00:00'),
4 ('05620504', 'other', '06:00:00', '18:00:00'),
5 ('05620505', 'monday', '06:00:00', '18:00:00'),
6 ('05620505', 'other', '18:00:00', '06:00:00'),
7 ('05620506', 'thursday', '06:00:00', '12:00:00'),
8 ('05620506', 'other', '12:00:00', '18:00:00'),
9 ('05620512', 'monday', '18:00:00', '06:00:00'),
```



10 ('05620512', 'other', '06:00:00', '18:00:00');

### 2.2.19 Bảng large\_equipment

```
1 INSERT INTO large_equipment (equipment_id, equipment_name, equipment_type,
2   equipment_status, purchase_date, purchase_location)
3 VALUES
4 ('EQ_AIR_01', 'Air Conditioner', 'Cooling', 'Working', '2021-05-01', 'HCM'),
5 ('EQ_TV_01', 'Television', 'Entertainment', 'Working', '2020-03-15', 'HCM'),
6 ('EQ_PC_01', 'Computer', 'IT', 'Under Repair', '2022-08-20', 'HN'),
7 ('EQ_FRG_01', 'Fridge', 'Appliance', 'Damaged', '2019-11-10', 'DN');
```

### 2.2.20 Bảng building\_equipment

```
1 INSERT INTO building_equipment (building_id, equipment_id, quantity)
2 VALUES
3 ('BK001', 'EQ_AIR_01', 2),
4 ('BK002', 'EQ_TV_01', 4),
5 ('BK003', 'EQ_PC_01', 6),
6 ('BK004', 'EQ_FRG_01', 8);
```

### 2.2.21 Bảng relative

```
1 INSERT INTO relative VALUES
2 ('05620513', 'An', 'Nguyen Van', '1975-05-15', 'Father', 'Tam Dan, Phu Ninh, Quang Nam',
3   ', '0389162345', 'Teacher'),
4 ('05620513', 'Hoa', 'Tran Thi', '1978-08-20', 'Mother', 'Tam Dan, Phu Ninh, Quang Nam',
5   ', '0389162346', 'Nurse'),
6 ('05620514', 'Phong', 'Dang Quang', '1976-03-10', 'Father', 'Son Tra, Son Tra, Da Nang',
7   ', '0328190285', 'Engineer'),
8 ('05620514', 'Linh', 'Nguyen Thi', '1980-11-25', 'Mother', 'Son Tra, Son Tra, Da Nang',
9   ', '0328190286', 'Accountant'),
10 ('05620515', 'Hung', 'Le Van', '1973-07-18', 'Father', 'Cam Le, Cam Le, Da Nang',
11   ', '0323824786', 'Doctor'),
12 ('05620515', 'Thuy', 'Tran Thi', '1975-09-30', 'Mother', 'Cam Le, Cam Le, Da Nang',
13   ', '0323824787', 'Teacher'),
```



8	(‘05620516’, ‘Tuan’, ‘Nguyen Anh’, ‘1972-12-05’, ‘Father’, ‘Thanh Khe, Thanh Khe, Da Nang’, ‘0309238479’, ‘Businessman’),
9	(‘05620516’, ‘Mai’, ‘Pham Thi’, ‘1976-04-15’, ‘Mother’, ‘Thanh Khe, Thanh Khe, Da Nang’, ‘0309238480’, ‘Housewife’),
10	(‘05620517’, ‘Cuong’, ‘Vo Van’, ‘1970-06-20’, ‘Father’, ‘Hoa Vang, Hoa Vang, Da Nang’, ‘0320245758’, ‘Police Officer’),
11	(‘05620517’, ‘Lan’, ‘Nguyen Thi’, ‘1974-10-10’, ‘Mother’, ‘Hoa Vang, Hoa Vang, Da Nang’, ‘0320245759’, ‘Nurse’),
12	(‘05620518’, ‘Dung’, ‘Tran Van’, ‘1971-01-15’, ‘Father’, ‘Hoa An, Lien Chieu, Da Nang’, ‘0329287548’, ‘Electrician’),
13	(‘05620518’, ‘Huong’, ‘Le Thi’, ‘1975-05-25’, ‘Mother’, ‘Hoa An, Lien Chieu, Da Nang’, ‘0329287549’, ‘Teacher’),
14	(‘05620519’, ‘Hai’, ‘Nguyen Van’, ‘1969-08-12’, ‘Father’, ‘Cau Moi, Ha Long, Quang Ninh’, ‘0320918268’, ‘Fisherman’),
15	(‘05620519’, ‘Hong’, ‘Pham Thi’, ‘1973-11-28’, ‘Mother’, ‘Cau Moi, Ha Long, Quang Ninh’, ‘0320918269’, ‘Tailor’),
16	(‘05620520’, ‘Khanh’, ‘Nguyen Van’, ‘1974-02-14’, ‘Father’, ‘Bai Chay, Ha Long, Quang Ninh’, ‘0387326654’, ‘Tour Guide’),
17	(‘05620520’, ‘Nga’, ‘Tran Thi’, ‘1978-07-19’, ‘Mother’, ‘Bai Chay, Ha Long, Quang Ninh’, ‘0387326655’, ‘Receptionist’),
18	(‘05620521’, ‘Long’, ‘Vo Van’, ‘1972-09-03’, ‘Father’, ‘Quang Trung, Quang Yen, Quang Ninh’, ‘0320975644’, ‘Construction Worker’),
19	(‘05620521’, ‘Thao’, ‘Nguyen Thi’, ‘1976-12-08’, ‘Mother’, ‘Quang Trung, Quang Yen, Quang Ninh’, ‘0320975645’, ‘Housewife’),
20	(‘05620522’, ‘Manh’, ‘Nguyen Van’, ‘1970-04-17’, ‘Father’, ‘Yen Thanh, Yen Hung, Quang Ninh’, ‘0399706546’, ‘Miner’),
21	(‘05620522’, ‘Nhung’, ‘Nguyen Thi’, ‘1974-10-22’, ‘Mother’, ‘Yen Thanh, Yen Hung, Quang Ninh’, ‘0399706547’, ‘Teacher’),
22	(‘05620523’, ‘Nam’, ‘Nguyen Hoang’, ‘1973-01-30’, ‘Father’, ‘Mong Duong, Cam Pha, Quang Ninh’, ‘0328100039’, ‘Driver’),
23	(‘05620523’, ‘Oanh’, ‘Le Thi’, ‘1977-06-15’, ‘Mother’, ‘Mong Duong, Cam Pha, Quang Ninh’, ‘0328100040’, ‘Shopkeeper’),
24	(‘05620524’, ‘Phuc’, ‘Nguyen Van’, ‘1971-03-25’, ‘Father’, ‘Uong Bi, Uong Bi, Quang Ninh’, ‘0391203471’, ‘Electrician’),
25	(‘05620524’, ‘Quyen’, ‘Tran Thi’, ‘1975-08-10’, ‘Mother’, ‘Uong Bi, Uong Bi, Quang Ninh’, ‘0391203472’, ‘Nurse’),
26	(‘05620525’, ‘Rinh’, ‘Vo Van’, ‘1968-07-12’, ‘Father’, ‘Dong Tien, Luong Son, Hoa Binh’, ‘0328906743’, ‘Farmer’),



27 ('05620525', 'Suong', 'Nguyen Thi', '1972-12-18', 'Mother', 'Dong Tien,Luong Son,Hoa  
Binh', '0328906744', 'Housewife'),  
28 ('05620526', 'Thang', 'Bui Van', '1974-05-20', 'Father', 'Mai Chau,Mai Chau,Hoa Binh  
, '0328109201', 'Tour Guide'),  
29 ('05620526', 'Trang', 'Nguyen Thi', '1978-10-05', 'Mother', 'Mai Chau,Mai Chau,Hoa  
Binh', '0328109202', 'Weaver'),  
30 ('05620527', 'Tung', 'Nguyen Duc', '1970-11-15', 'Father', 'Dinh Hoa,Dinh Hoa,Thai  
Nguyen', '0328102981', 'Tea Farmer'),  
31 ('05620527', 'Uyen', 'Phan Thi', '1974-04-20', 'Mother', 'Dinh Hoa,Dinh Hoa,Thai  
Nguyen', '0328102982', 'Teacher'),  
32 ('05620528', 'Vinh', 'Nguyen Van', '1973-02-28', 'Father', 'Son Cam,Thai Nguyen,Thai  
Nguyen', '0322134784', 'Miner'),  
33 ('05620528', 'Xuan', 'Nguyen Thi', '1977-07-15', 'Mother', 'Son Cam,Thai Nguyen,Thai  
Nguyen', '0322134785', 'Housewife'),  
34 ('05620529', 'Yen', 'Phan Van', '1971-08-10', 'Father', 'Hoa Binh,Hoa Binh,Hoa Binh'  
, '0320975424', 'Construction Worker'),  
35 ('05620529', 'Zung', 'Nguyen Thi', '1975-01-25', 'Mother', 'Hoa Binh,Hoa Binh,Hoa  
Binh', '0320975425', 'Tailor'),  
36 ('05620530', 'Anh', 'Nguyen Van', '1974-06-18', 'Father', 'Lang Son,Lang Son,Lang  
Son', '0328100348', 'Shopkeeper'),  
37 ('05620530', 'Bich', 'Tran Thi', '1978-11-30', 'Mother', 'Lang Son,Lang Son,Lang Son  
, '0328100349', 'Housewife'),  
38 ('05620531', 'Chien', 'Vo Van', '1972-03-15', 'Father', 'Phu Tan,Tan Phu,Dong Nai',  
'0326784593', 'Factory Worker'),  
39 ('05620531', 'Dung', 'Nguyen Thi', '1976-08-20', 'Mother', 'Phu Tan,Tan Phu,Dong Nai  
, '0326784594', 'Housewife'),  
40 ('05620532', 'Giang', 'Nguyen Van', '1970-05-10', 'Father', 'Bac Giang,Bac Giang,Bac  
Giang', '0326735205', 'Farmer'),  
41 ('05620532', 'Hanh', 'Nguyen Thi', '1974-10-15', 'Mother', 'Bac Giang,Bac Giang,Bac  
Giang', '0326735206', 'Teacher'),  
42 ('05620533', 'Khoa', 'Nguyen Van', '1973-01-20', 'Father', 'Tuy An,Tuy Hoa,Phu Yen',  
'0327843793', 'Fisherman'),  
43 ('05620533', 'Lien', 'Nguyen Thi', '1977-06-25', 'Mother', 'Tuy An,Tuy Hoa,Phu Yen',  
'0327843794', 'Seamstress'),  
44 ('05620534', 'Minh', 'Nguyen Van', '1971-04-15', 'Father', 'Phu Ly,Phu Ly,Ha Nam',  
'0328129344', 'Driver'),  
45 ('05620534', 'Nga', 'Tran Thi', '1975-09-20', 'Mother', 'Phu Ly,Phu Ly,Ha Nam',  
'0328129345', 'Shopkeeper'),



```

46 ('05620535', 'Phu', 'Nguyen Van', '1974-07-25', 'Father', 'Duc Tho,Duc Tho,Ha Tinh',
     '0393280451', 'Teacher'),
47 ('05620535', 'Quynh', 'Nguyen Thi', '1978-12-30', 'Mother', 'Duc Tho,Duc Tho,Ha Tinh
     , '0393280452', 'Nurse'),
48 ('05620536', 'Son', 'Nguyen Van', '1972-02-10', 'Father', 'Quynh Luu,Quynh Luu,Nghe
     An', '0924723456', 'Farmer'),
49 ('05620536', 'Tam', 'Tran Thi', '1976-07-15', 'Mother', 'Quynh Luu,Quynh Luu,Nghe An
     , '0924723457', 'Housewife'),
50 ('05620537', 'Thanh', 'Tran Van', '1970-05-20', 'Father', 'Vinh,Vinh,Nghe An', '
     0739012013', 'Construction Worker'),
51 ('05620537', 'Uyen', 'Nguyen Thi', '1974-10-25', 'Mother', 'Vinh,Vinh,Nghe An', '
     0739012014', 'Teacher'),
52 ('05620538', 'Viet', 'Nguyen Van', '1973-08-15', 'Father', 'Nam Dan,Nam Dan,Nghe An'
     , '0928761452', 'Teacher'),
53 ('05620538', 'Xuan', 'Nguyen Thi', '1977-01-20', 'Mother', 'Nam Dan,Nam Dan,Nghe An'
     , '0928761453', 'Nurse'),
54 ('05620539', 'Yen', 'Bui Van', '1971-04-10', 'Father', 'Thanh Chuong,Thanh Chuong,
     Nghe An', '0322934735', 'Farmer'),
55 ('05620540', 'Anh', 'Nguyen Van', '1974-12-20', 'Father', 'Cua Lo,Cua Lo,Nghe An', '
     0329235484', 'Fisherman'),
56 ('05620540', 'Binh', 'Nguyen Thi', '1978-05-25', 'Mother', 'Cua Lo,Cua Lo,Nghe An', '
     0329235485', 'Seamstress'),
57 ('05620541', 'Cuong', 'Tran Van', '1972-03-15', 'Father', 'Dien Chau,Dien Chau,Nghe
     An', '0391287452', 'Teacher');

```

### 2.2.22 Bảng disciplinary\_action

```

1 INSERT INTO disciplinary_action VALUES
2 ('DA001', 'Cleaning Duty', 'Violation of quiet hours', '2023-03-15', '2023-03-16', '
     2023-06-16', 'low', 'cancelled'),
3 ('DA002', 'Community Service', 'Violation of quiet hours', '2023-04-20', '2023-04-21', NULL, 'expulsion', 'active'),
4 ('DA003', 'Expulsion', 'Smoking in non-designated areas', '2023-02-10', '2023-02-11
     ', '2023-04-11', 'high', 'active'),
5 ('DA004', 'Expulsion', 'Physical assault or fighting', '2023-01-05', '2023-01-06', NULL, 'expulsion', 'active'),
6 ('DA005', 'Yard Cleaning', 'Unauthorized guests', '2023-05-12', '2023-05-13', '2023-08-13', 'low', 'completed'),

```



```
7 ('DA006', 'Classroom Setup', 'Physical assault or fighting', '2023-08-01', ,
 2023-08-02', '2023-11-03', 'high', 'completed'),
8 ('DA007', 'Dorm Cleaning', 'Physical assault or fighting', '2023-09-01', ,
 2023-09-02', '2023-10-02', 'high', 'completed'),
9 ('DA008', 'Cafeteria Duty', 'Harassment or bullying', '2025-02-20', '2025-02-21', ,
 2025-05-21', 'medium', 'active'),
10 ('DA009', 'Library Service', 'Unauthorized access to restricted areas', ,
 2025-03-10', '2025-03-11', '2025-06-11', 'medium', 'active'),
11 ('DA010', 'Hall Monitoring', 'Disregard for dormitory staff instructions', ,
 2025-03-15', '2025-03-16', '2025-07-16', 'low', 'active');
```

### 2.2.23 Bảng student\_discipline

```
1 INSERT INTO student_discipline VALUES
2 ('DA001', '05620513'), ('DA002', '05620520'), ('DA003', '05620525'),
3 ('DA004', '05620529'), ('DA005', '05620530'), ('DA006', '05620525'),
4 ('DA007', '05620525'), ('DA008', '05620531'), ('DA009', '05620515'),
5 ('DA010', '05620537');
```

### 2.2.24 Bảng discipline\_staff

```
1 INSERT INTO discipline_staff VALUES
2 ('DA001', '05620501'), ('DA002', '05620502'), ('DA003', '05620503'),
3 ('DA004', '05620507'), ('DA005', '05620501'), ('DA006', '05620502'),
4 ('DA007', '05620507'), ('DA008', '05620502'), ('DA009', '05620501'),
5 ('DA010', '05620507');
```

### 2.2.25 Bảng identity\_card

```
1 INSERT INTO identity_card (ID, Issued_Date)
2 SELECT CONCAT('ID', LPAD(ROW_NUMBER() OVER (ORDER BY sssn), 5, '0')) AS ID,
3 -- Random date within the last 5 years
4 DATE_ADD(CURDATE(), INTERVAL(RAND() * 1825) DAY) AS Issued_Date
5 FROM student UNION ALL
6 SELECT CONCAT('ID',
7 LPAD(ROW_NUMBER()
8 OVER (ORDER BY essn) + (SELECT COUNT(*) FROM student),
```



```

9          5, '0'
10         )
11     ) AS ID,
12 -- Random date within the last 5 years
13   DATE_ADD(CURDATE(), INTERVAL (RAND() * 1825) DAY) AS Issued_Date
14 FROM employee;

```

## 2.2.26 Bảng employee\_card

```

1 INSERT INTO employee_card VALUES
2 ('ID00041', '2025-04-14', '2026-03-14', '05620501', NULL),
3 ('ID00042', '2025-04-14', '2026-05-20', '05620502', NULL),
4 ('ID00043', '2025-04-14', '2026-02-28', '05620503', NULL),
5 ('ID00044', '2025-04-14', '2026-06-15', '05620504', NULL),
6 ('ID00045', '2025-04-14', '2026-04-10', '05620505', NULL),
7 ('ID00046', '2025-04-14', '2026-07-01', '05620506', NULL),
8 ('ID00047', '2025-04-14', '2026-03-30', '05620507', NULL),
9 ('ID00048', '2025-04-14', '2026-08-25', '05620508', NULL),
10 ('ID00049', '2025-04-14', '2026-01-12', '05620509', NULL),
11 ('ID00050', '2025-04-14', '2026-09-05', '05620510', NULL),
12 ('ID00051', '2025-04-14', '2026-03-18', '05620511', NULL),
13 ('ID00052', '2025-04-14', '2026-06-22', '05620512', NULL);

```

## 2.2.27 Bảng dormitory\_card

```

1 INSERT INTO dormitory_card VALUES
2 ('ID00001', '2025-04-14', '2026-01-15', '05620513', TRUE),
3 ('ID00002', '2025-04-14', '2026-02-20', '05620514', TRUE),
4 ('ID00003', '2025-04-14', '2026-03-10', '05620515', TRUE),
5 ('ID00004', '2025-04-14', '2026-04-05', '05620516', TRUE),
6 ('ID00005', '2025-04-14', '2026-05-25', '05620517', TRUE),
7 ('ID00006', '2025-04-14', '2026-06-15', '05620518', TRUE),
8 ('ID00007', '2025-04-14', '2026-07-01', '05620519', TRUE),
9 ('ID00008', '2025-04-14', '2026-08-10', '05620520', TRUE),
10 ('ID00009', '2025-04-14', '2026-09-05', '05620521', TRUE),
11 ('ID00010', '2025-04-14', '2026-10-20', '05620522', TRUE),
12 ('ID00011', '2025-04-14', '2026-11-15', '05620523', TRUE),

```



```
13 ('ID00012', '2025-04-14', '2026-12-01', '05620524', TRUE),  
14 ('ID00013', '2025-04-14', '2026-01-25', '05620525', TRUE),  
15 ('ID00014', '2025-04-14', '2026-02-18', '05620526', TRUE),  
16 ('ID00015', '2025-04-14', '2026-03-22', '05620527', TRUE),  
17 ('ID00016', '2025-04-14', '2026-04-30', '05620528', TRUE),  
18 ('ID00017', '2025-04-14', '2026-05-12', '05620529', TRUE),  
19 ('ID00018', '2025-04-14', '2026-06-08', '05620530', TRUE),  
20 ('ID00019', '2025-04-14', '2026-07-25', '05620531', TRUE),  
21 ('ID00020', '2025-04-14', '2026-08-14', '05620532', TRUE),  
22 ('ID00021', '2025-04-14', '2026-09-18', '05620533', TRUE),  
23 ('ID00022', '2025-04-14', '2026-10-05', '05620534', TRUE),  
24 ('ID00023', '2025-04-14', '2026-11-22', '05620535', TRUE),  
25 ('ID00024', '2025-04-14', '2026-12-10', '05620536', TRUE),  
26 ('ID00025', '2025-04-14', '2026-01-30', '05620537', TRUE),  
27 ('ID00026', '2025-04-14', '2026-02-25', '05620538', TRUE),  
28 ('ID00027', '2025-04-14', '2026-03-15', '05620539', TRUE),  
29 ('ID00028', '2025-04-14', '2026-04-20', '05620540', TRUE),  
30 ('ID00029', '2025-04-14', '2026-05-18', '05620541', TRUE),  
31 ('ID00030', '2025-04-14', '2026-06-22', '05620542', TRUE),  
32 ('ID00031', '2025-04-14', '2026-07-30', '05620543', TRUE),  
33 ('ID00032', '2025-04-14', '2026-08-25', '05620544', TRUE),  
34 ('ID00033', '2025-04-14', '2026-09-12', '05620545', TRUE),  
35 ('ID00034', '2025-04-14', '2026-10-08', '05620546', TRUE),  
36 ('ID00035', '2025-04-14', '2026-11-05', '05620547', TRUE),  
37 ('ID00036', '2025-04-14', '2026-12-20', '05620548', TRUE),  
38 ('ID00037', '2025-04-14', '2026-01-18', '05620549', TRUE),  
39 ('ID00038', '2025-04-14', '2026-02-28', '05620550', TRUE),  
40 ('ID00039', '2025-04-14', '2026-03-25', '05620551', TRUE),  
41 ('ID00040', '2025-04-14', '2026-04-15', '05620552', TRUE);
```



### 3 Thủ tục - Hàm - Trigger

#### 3.1 Thủ tục

##### 3.1.1 Kiểm tra xem một phòng cụ thể đã đủ người chưa

###### Tham số đầu vào

- p\_building\_id (CHAR(5)): mã tòa nhà cần kiểm tra.
- p\_room\_id (CHAR(5)): mã phòng cần kiểm tra.

###### Logic

Thực hiện truy vấn bảng living\_room để kiểm tra một phòng cụ thể trong một tòa nhà có đang bị thiếu người ở hay không. Nếu số lượng sinh viên hiện tại trong phòng nhỏ hơn sức chứa tối đa, thủ tục sẽ trả về thông tin về phòng đó cùng với tỷ lệ lấp đầy dưới dạng phần trăm. Nếu phòng đã đầy hoặc không tồn tại, thủ tục không trả về kết quả nào. Nếu nhập mã tòa nhà và mã phòng không đúng 5 ký tự, sẽ trả về thông báo lỗi.

###### Kết quả đầu ra

- Mã tòa nhà (building\_id).
- Mã phòng (room\_id).
- Số sinh viên hiện tại (current\_num\_of\_students).
- Sức chứa tối đa (max\_num\_of\_students).
- Tỷ lệ lấp đầy (occupancy\_rate).

```
1 CREATE PROCEDURE check_one_room_underoccupied(
2     IN p_building_id CHAR(5),
3     IN p_room_id CHAR(5)
4 )
5 BEGIN
6     -- Check p_building_id
7     IF LENGTH(REPLACE(TRIM(p_building_id), ' ', '')) != 5 THEN
8         SIGNAL SQLSTATE '45000'
9         SET MESSAGE_TEXT = 'Building ID must be exactly 5 characters long.';
10    END IF;
11    -- Check p_room_id
12    IF LENGTH(REPLACE(TRIM(p_room_id), ' ', '')) != 5 THEN
13        SIGNAL SQLSTATE '45000'
14        SET MESSAGE_TEXT = 'Room ID must be exactly 5 characters long.';
15    END IF;
16    SELECT
```



```
17     building_id,  
18     room_id,  
19     current_num_of_students,  
20     max_num_of_students,  
21     CONCAT(occupancy_rate, '%') AS occupancy_rate  
22 FROM living_room  
23 WHERE building_id = p_building_id  
24     AND room_id = p_room_id  
25     AND current_num_of_students < max_num_of_students;  
26 END;
```

```
1 •  use final;  
2 •  CALL check_one_room_underoccupied('BK001','P.104');
```

Result Grid					
	building_id	room_id	current_num_of_students	max_num_of_students	occupancy_rate
▶	BK001	P.104	3	6	50.00%

### 3.1.2 Thủ tục lấy thông tin sinh viên theo SSN

#### Logic

Thủ tục này nhận vào một SSN và thực hiện:

- Kiểm tra SSN phải đúng 8 ký tự (không tính khoảng trắng).
- Truy vấn và trả về thông tin cá nhân, sinh viên, số điện thoại, địa chỉ, và email liên kết với SSN đó.

```
1 CREATE PROCEDURE get_student_by_ssn(IN p_ssn CHAR(8))  
2 BEGIN  
3     -- Check if SSN is exactly 8 characters long  
4     IF LENGTH(REPLACE(TRIM(p_ssn), ' ', '')) != 8 THEN
```



```
5      SIGNAL SQLSTATE '45000'  
6      SET MESSAGE_TEXT = 'SSN must be exactly 8 characters long.';  
7  END IF;  
8  
9  -- Query to retrieve student's information  
10 SELECT  
11     p.ssn,  
12     first_name,  
13     last_name,  
14     p.birthday,  
15     p.sex,  
16     p.health_state,  
17     p.ethnic_group,  
18     s.student_id,  
19     s.has_health_insurance,  
20     s.study_status,  
21     s.class_name,  
22     s.faculty,  
23     s.building_id,  
24     s.room_id,  
25     -- Concatenate multiple phone numbers into one field  
26     GROUP_CONCAT(DISTINCT ph.phone_number SEPARATOR ';') AS phone_numbers,  
27     -- Concatenate multiple addresses into one field  
28     GROUP_CONCAT(  
29         DISTINCT CONCAT_WS(' ', ' , a.commune, a.district, a.province)  
30         SEPARATOR ';' )  
31     ) AS addresses,  
32     -- Concatenate multiple emails into one field  
33     GROUP_CONCAT(DISTINCT e.email SEPARATOR ';') AS emails  
34 FROM people p  
35 JOIN student s ON p.ssn = s.sssn  
36 LEFT JOIN phone_number ph ON p.ssn = ph.ssn  
37 LEFT JOIN address a ON p.ssn = a.ssn  
38 LEFT JOIN email e ON p.ssn = e.ssn  
39 WHERE p.ssn = p_ssn  
40 GROUP BY  
41     p.ssn,  
42     p.last_name,
```



```
43     p.first_name,  
44     p.birthday,  
45     p.sex,  
46     p.health_state,  
47     p.ethnic_group,  
48     s.student_id,  
49     s.has_health_insurance,  
50     s.study_status,  
51     s.class_name,  
52     s.faculty,  
53     s.building_id,  
54     s.room_id;  
55 END;
```

```
1 •  use final;  
2 •  CALL get_student_by_ssn('05620513');
```

ssn	first_name	last_name	birthday	sex	health_state	ethnic_group	student_id	has_health_insurance	study_status	class_name	faculty	building_id	room_id
05620513	Khoi	Nguyen Minh	2003-06-03	M	Good	Kinh	2312613	1	Active	KHMT1	Computer Science	BK001	P.104

### 3.1.3 Thống kê kỷ luật sinh viên theo từng lỗi vi phạm

#### Logic

Thủ tục thống kê số lần sinh viên vi phạm theo từng loại lỗi (lý do vi phạm) dựa trên bảng kỷ luật sinh viên. Chỉ những lý do có số lần vi phạm lớn hơn hoặc bằng `min_count` mới được hiển thị. Kết quả được sắp xếp theo số lượng vi phạm giảm dần.

```
1 CREATE PROCEDURE getViolationStatisticsByType(IN min_count INT)  
2 BEGIN  
3     SELECT reason, COUNT(*) AS violation_count  
4     FROM student_discipline sd  
5     JOIN disciplinary_action da ON sd.action_id = da.action_id  
6     GROUP BY reason  
7     HAVING COUNT(*) >= min_count  
8     ORDER BY violation_count DESC;
```



9 END;

```
1 •  use final;
2 •  CALL get_student_by_ssn('05620513');
```

ssn	first_name	last_name	birthday	sex	health_state	ethnic_group	student_id	has_health_insurance	study_status	class_name	faculty	building_id	room_id
05620513	Khoi	Nguyen Minh	2003-06-03	M	Good	Kinh	2312613	1	Active	KHMT1	Computer Science	BK001	P.104

## 3.2 Hàm

### 3.2.1 Số sinh viên bị kỷ luật trong khoảng thời gian (start-end)

#### Tham số đầu vào

- start\_date (DATE): ngày bắt đầu khoảng thời gian cần thống kê.
- end\_date (DATE): ngày kết thúc khoảng thời gian cần thống kê.

#### Logic

Hàm trả về tổng số sinh viên bị kỷ luật trong khoảng thời gian được chỉ định. Chỉ tính các hành vi kỷ luật có hiệu lực trong khoảng thời gian đó và loại trừ những trường hợp bị đuổi học ('expulsion'). Kết quả là số lượng sinh viên khác nhau (distinct) có hành vi vi phạm trong khoảng thời gian được chỉ định.

```

1 CREATE FUNCTION count_disciplined_students(start_date Date, end_date Date)
2 RETURNS INT
3 DETERMINISTIC
4 BEGIN
5     DECLARE student_count INT;
6     IF start_date > end_date THEN
7         SIGNAL SQLSTATE '45000'
8         SET MESSAGE_TEXT = 'START DATE MUST BE LESS THAN END DATE';
9     END IF;
10    SELECT COUNT(distinct sd.ssn) INTO student_count
11    FROM student_discipline sd
12    JOIN disciplinary_action da ON sd.action_id = da.action_id
13    WHERE da.effective_from <= end_date AND da.effective_to >= start_date
14        AND da.action_type NOT IN ('expulsion');
```



```
15    RETURN student_count;  
16 END;
```

```
1 •  use final;  
2 •  SELECT count_disciplined_students('2023-01-01', '2023-12-31') AS disciplined_students;
```

Result Grid	
disciplined_students	3
▶	3

### 3.2.2 Tổng số sinh viên trong một tòa nhà

#### Tham số đầu vào

- bid (CHAR(5)): mã tòa nhà cần thống kê.

#### Logic

Hàm tính tổng số sinh viên hiện đang ở trong tất cả các phòng của một tòa nhà cụ thể. Dữ liệu được truy vấn từ bảng living\_room và sử dụng SUM trên trường current\_num\_of\_students. Nếu không có sinh viên nào, hàm trả về giá trị 0.

```
1 CREATE FUNCTION total_students_by_building(bid CHAR(5))  
2 RETURNS INT  
3 DETERMINISTIC  
4 READS SQL DATA  
5 BEGIN  
6     DECLARE total INT  
7     IF LENGTH(REPLACE(TRIM(bid), ' ', '')) != 5 THEN  
8         SIGNAL SQLSTATE '45000'  
9         SET MESSAGE_TEXT = 'Building ID must be exactly 5 characters long.';  
10    END IF;  
11    SELECT SUM(current_num_of_students)  
12    INTO total  
13    FROM living_room  
14    WHERE building_id = bid;
```



```
15    RETURN IFNULL(total, 0);
16 END;
```

```
1 •  use final;
2 •  CALL get_student_by_ssn('05620513');
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:										
ssn	first_name	last_name	birthday	sex	health_state	ethnic_group	student_id	has_health_insurance	study_status	class_name	faculty	building_id	room_id
05620513	Khoi	Nguyen Minh	2003-06-03	M	Good	Kinh	2312613	1	Active	KHMT1	Computer Science	BK001	P.104

### 3.2.3 Số thẻ nội trú chưa hết hạn

#### Logic

Hàm trả về số lượng thẻ nội trú đang còn hiệu lực (Validity = 1) từ bảng dormitory\_card. Sử dụng hàm COUNT(\*) để đếm số dòng thỏa điều kiện.

```
1 CREATE FUNCTION num_validity_dormitory_card() RETURNS INT DETERMINISTIC
2 BEGIN
3     DECLARE num INT;
4     SELECT COUNT(*) INTO num FROM dormitory_card WHERE Validity = 1;
5     RETURN num;
6 END;
```

```
1 •  use final;
2 •  SELECT num_validity_dormitory_card() AS valid_cards;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
valid_cards	40		



### 3.3 Trigger

#### 3.3.1 Trigger khi chuyển phòng cho sinh viên

##### Logic

Trigger được kích hoạt **sau khi** có cập nhật dữ liệu trên bảng **student**. Áp dụng cho từng dòng dữ liệu được cập nhật (**FOR EACH ROW**). Trigger đảm bảo rằng dữ liệu về số lượng sinh viên và tình trạng của phòng luôn được đồng bộ mỗi khi có sinh viên chuyển phòng.

- Trigger chỉ thực hiện khi sinh viên thực sự chuyển phòng (thay đổi **room\_id** hoặc **building\_id**).
- Giảm số lượng sinh viên và cập nhật tỷ lệ lấp đầy (**occupancy\_rate**) của phòng cũ.
- Nếu phòng cũ không còn sinh viên nào sau khi trừ, thì cập nhật trạng thái phòng thành '**Available**' trong bảng **room**.
- Tăng số lượng sinh viên và cập nhật lại tỷ lệ lấp đầy của phòng mới.

```
1 CREATE TRIGGER after_update_student_room
2 AFTER UPDATE ON student
3 FOR EACH ROW
4 BEGIN
5     DECLARE old_now INT;
6     DECLARE old_max INT;
7     DECLARE new_now INT;
8     DECLARE new_max INT;
9     -- Update only if the room has changed
10    IF OLD.room_id IS NOT NULL AND OLD.building_id IS NOT NULL AND
11        (OLD.room_id != NEW.room_id OR OLD.building_id != NEW.building_id)
12    THEN
13        -- Subtract the student from the old room
14        SELECT current_num_of_students, max_num_of_students
15        INTO old_now, old_max
16        FROM living_room
17        WHERE room_id = OLD.room_id AND building_id = OLD.building_id;
18        -- Update the old room
19        UPDATE living_room
20        SET current_num_of_students = old_now - 1,
21            occupancy_rate = ROUND(100.0 * (old_now - 1) / old_max, 2)
22        WHERE room_id = OLD.room_id AND building_id = OLD.building_id;
23        -- If the old room has no students left -> set status to Available
```



```

24      IF old_now - 1 = 0 THEN
25          UPDATE room
26              SET room_status = 'Available'
27              WHERE room_id = OLD.room_id AND building_id = OLD.building_id;
28      END IF;
29
30      -- Add the student to the new room
31      SELECT current_num_of_students, max_num_of_students
32          INTO new_now, new_max
33          FROM living_room
34          WHERE room_id = NEW.room_id AND building_id = NEW.building_id;
35
36      -- Update the registered room
37      UPDATE living_room
38          SET current_num_of_students = new_now + 1,
39              occupancy_rate = ROUND(100.0 * (new_now + 1) / new_max, 2)
40          WHERE room_id = NEW.room_id AND building_id = NEW.building_id;
41      END IF;
42
43  END;

```

1 • select\*from living\_room

	building_id	room_id	max_num_of_students	current_num_of_students	rental_price	occupancy_rate
▶	BK001	P.104	6	3	1500000.00	50.00
	BK001	P.201	6	0	1500000.00	0.00
	BK001	P.202	6	0	1500000.00	0.00
	BK001	P.203	6	0	1500000.00	0.00
	BK001	P.204	6	3	1500000.00	50.00
	BK001	P.301	6	0	1500000.00	0.00
	BK001	P.302	6	0	1500000.00	0.00
	BK001	P.303	6	0	1500000.00	0.00
	BK001	P.304	6	4	1500000.00	66.67
	BK001	P.401	6	0	1500000.00	0.00

Hình 3.1: Trước khi cập nhật



```
1 •  use final;
2   -- Check trigger after_update_student_room()
3   -- Chuyển sinh viên từ 204(BK001)->104(BK001)
4 •  UPDATE student
5   SET building_id = 'BK001', room_id = 'P.104'
6   WHERE sssn = '05620531';
7
8 •  select*from living_room
9
```

	building_id	room_id	max_num_of_students	current_num_of_students	rental_price	occupancy_rate
▶	BK001	P.104	6	4	1500000.00	66.67
	BK001	P.201	6	0	1500000.00	0.00
	BK001	P.202	6	0	1500000.00	0.00
	BK001	P.203	6	0	1500000.00	0.00
	BK001	P.204	6	2	1500000.00	33.33
	BK001	P.301	6	0	1500000.00	0.00
	BK001	P.302	6	0	1500000.00	0.00
	BK001	P.303	6	0	1500000.00	0.00
	BK001	P.304	6	4	1500000.00	66.67

Hình 3.2: Sau khi cập nhật

### 3.3.2 Trigger kiểm tra và cập nhật khi thêm sinh viên mới

#### Logic

Trigger được kích hoạt **trước khi** thêm một dòng dữ liệu mới vào bảng student. Áp dụng cho từng dòng dữ liệu được chèn vào (FOR EACH ROW). Trigger đảm bảo rằng việc thêm sinh viên mới chỉ thực hiện nếu phòng còn chỗ trống, đồng thời cập nhật số lượng sinh viên và tỷ lệ lấp đầy của phòng.

- Kiểm tra số lượng sinh viên hiện tại trong phòng so với số lượng tối đa.
- Nếu phòng đã đầy, trigger sẽ chặn việc thêm mới và đưa ra thông báo lỗi 'The room is full of students.'
- Nếu còn chỗ, tăng số lượng sinh viên hiện tại thêm 1 và cập nhật lại tỷ lệ lấp đầy (occupancy\_rate).

```
1 CREATE TRIGGER before_insert_student
2 BEFORE INSERT ON student
3 FOR EACH ROW
4 BEGIN
5   DECLARE v_now_num INT;
```



```

6   DECLARE v_max_num INT;
7   DECLARE v_new_now_num INT;
8   --
9   SELECT current_num_of_students, max_num_of_students
10  INTO v_now_num, v_max_num
11  FROM living_room
12  WHERE room_id = NEW.room_id
13    AND building_id = NEW.building_id;
14  -- Check whether the room is full
15  IF v_now_num >= v_max_num THEN
16    SIGNAL SQLSTATE '45000'
17    SET MESSAGE_TEXT = 'The room is full of students.';
18  END IF;
19
20  SET v_new_now_num = v_now_num + 1;
21  -- Update the room's values
22  UPDATE living_room
23  SET current_num_of_students = v_new_now_num,
24    occupancy_rate = v_new_now_num / max_num_of_students * 100
25  WHERE room_id = NEW.room_id
26    AND building_id = NEW.building_id;
27 END;

```

```

1 • use final;
2   -- Check trigger before_insert_student()
3 • SELECT building_id, room_id, current_num_of_students, max_num_of_students, occupancy_rate
4   FROM living_room
5   WHERE building_id = 'BK001' AND room_id = 'P.104';

```

building_id	room_id	current_num_of_students	max_num_of_students	occupancy_rate
BK001	P.104	3	6	50.00
*	NULL	NULL	NULL	NULL

① 1172 20:29:11 INSERT INTO student VALUES ('05620513', '2312613', true, 'Active', 'KHMT1... Error Code: 1644. The room is full of students. 0.000 sec



### 3.3.3 Trigger xử lý sinh viên bị đuổi khỏi KTX

#### Logic

Trigger này được kích hoạt **sau khi** một bản ghi về kỷ luật sinh viên (disciplinary action) được thêm vào bảng `student_discipline`. Nếu sinh viên bị kỷ luật với mức độ `expulsion` và hành động đang `active`, trigger sẽ:

- Kiểm tra phòng của sinh viên bị đuổi và cập nhật thông tin phòng của sinh viên thành `NULL`.
- Giảm số lượng sinh viên trong phòng tương ứng và tính lại tỷ lệ lấp đầy phòng.

```
1 CREATE TRIGGER after_expulsion
2 AFTER INSERT ON student_discipline
3 FOR EACH ROW
4 BEGIN
5     DECLARE severity ENUM('low', 'medium', 'high', 'expulsion');
6     DECLARE action_status ENUM('pending', 'active', 'completed', 'cancelled');
7     DECLARE old_room CHAR(5);
8     DECLARE old_building CHAR(5);
9     -- Get severity and action status
10    SELECT severity_level, status INTO severity, action_status
11    FROM disciplinary_action
12    WHERE action_id = NEW.action_id;
13    -- If the severity is 'expulsion' and action is active
14    IF severity = 'expulsion' AND action_status = 'active' THEN
15        -- Get current room and building of the student
16        SELECT room_id, building_id INTO old_room, old_building
17        FROM student
18        WHERE sssn = NEW.sssn;
19        -- Update student's room information
20        UPDATE student
21        SET room_id = NULL, building_id = NULL
22        WHERE sssn = NEW.sssn;
23        -- Update living room data
24        UPDATE living_room
25        SET current_num_of_students = current_num_of_students - 1,
26            occupancy_rate = 100.0 * (current_num_of_students - 1) /
27            max_num_of_students
28        WHERE room_id = old_room AND building_id = old_building;
END IF;
```



29 END;

```

1 •  use final;
2   -- Check trigger after_expulsion()
3   -- check actionId(DA002) của sinh viên(05620520) bị expulsion
4
5 •  select*from student
6

```

Result Grid								
sssn	student_id	has_health_insurance	study_status	class_name	faculty	building_id	room_id	
05620516	2112347	1	Active	CNTT2	Information Technology	BK001	P.104	
05620517	2312348	1	Active	DTV1	Electronics and Telecommunications Engineering	BK002	P.102	
05620518	2412349	1	Active	DTV2	Electronics and Telecommunications Engineering	BK002	P.104	
05620519	2212350	1	Active	CK1	Mechanical Engineering	BK002	P.203	
05620520	2012351	1	Non_Active	CK2	Mechanical Engineering	NULL	NULL	
05620521	2112352	1	Active	D1	Electrical Engineering	BK003	P.104	
05620522	2312353	1	Active	D2	Electrical Engineering	BK003	P.104	
05620523	2412354	1	Active	HTTT1	Information Security	BK003	P.104	
05620524	2212355	1	Active	HTTT2	Information Security	BK003	P.104	

```

INSERT INTO student VALUES
('05620513', '2312613', true, 'Active', 'KHMT1', 'Computer Science','BK001', 'P.104'),
('05620514', '2212345', true, 'Active', 'KHMT2', 'Computer Science', 'BK001', 'P.104'),
('05620515', '2012346', true, 'Active', 'CNTT1', 'Information Technology', 'BK003', 'P.204'),
('05620516', '2112347', true, 'Active', 'CNTT2', 'Information Technology', 'BK001', 'P.104'),
('05620517', '2312348', true, 'Active', 'DTV1', 'Electronics and Telecommunications Engineering', 'BK002', 'P.102'),
('05620518', '2412349', true, 'Active', 'DTV2', 'Electronics and Telecommunications Engineering', 'BK002', 'P.104'),
('05620519', '2212350', true, 'Active', 'CK1', 'Mechanical Engineering', 'BK002', 'P.203'),
('05620520', '2012351', true, 'Non_Active', 'CK2', 'Mechanical Engineering', 'BK002', 'P.102'),
('05620521', '2112352', true, 'Active', 'D1', 'Electrical Engineering', 'BK003', 'P.104'),
('05620522', '2312353', true, 'Active', 'D2', 'Electrical Engineering', 'BK003', 'P.104'),
('05620523', '2412354', true, 'Active', 'HTTT1', 'Information Security', 'BK003', 'P.104'),
('05620524', '2212355', true, 'Active', 'HTTT2', 'Information Security', 'BK003', 'P.104'),

```

Hình 3.3: Trước khi ký luật sinh viên



```
1 •  select*from living_room  
2
```

Result Grid						
	building_id	room_id	max_num_of_students	current_num_of_students	rental_price	occupancy_rate
	BK001	P.402	6	0	1500000.00	0.00
	BK001	P.403	6	0	1500000.00	0.00
	BK001	P.404	6	0	1500000.00	0.00
▶	BK002	P.102	6	1	1500000.00	16.67
	BK002	P.104	6	1	1500000.00	16.67
	BK002	P.201	6	0	1500000.00	0.00
	BK002	P.202	6	0	1500000.00	0.00
	BK002	P.203	6	1	1500000.00	16.67
	BK002	P.204	6	4	1500000.00	16.67

Hình 3.4: Sau khi ký luật sinh viên

### 3.3.4 Một số trigger khác

1. Trigger tự động tạo emp\_id khi thêm nhân viên mới
2. Trigger cập nhật số lượng nhân viên khi thêm nhân viên quản lý
3. Trigger cập nhật số lượng nhân viên khi thêm nhân viên y tế
4. Trigger cập nhật số lượng nhân viên khi thêm nhân viên khác
5. Trigger cập nhật khi xóa sinh viên khỏi hệ thống
6. Trigger cập nhật khi xóa nhân viên khác khỏi hệ thống
7. Trigger xóa bằng cấp nhân viên y tế khi xóa nhân viên
8. Trigger xóa bằng cấp của nhân viên quản lý khi xóa nhân viên quản lý
9. Trigger xử lý sinh viên bị đuổi khỏi KTX (3 lần 'High')
10. Trigger chuyển phòng cho nhân viên khác
11. Trigger chuyển phòng cho nhân viên quản lý
12. Trigger kiểm tra phòng không thể trùng loại phòng
13. Trigger kiểm tra thời gian của phòng quản lý
14. Trigger cập nhật tỷ lệ lấp đầy của phòng ở



## 4 Triển khai ứng dụng

### 4.1 Tạo user

Tạo user

```
1 CREATE USER 'sManager'@'%' IDENTIFIED BY 'admin123';
```

Phân quyền

```
1 GRANT ALL PRIVILEGES ON railway.* TO 'sManager'@'%';
```

### 4.2 Giới thiệu môi trường lập trình

Ứng dụng được phát triển dưới dạng **Web Application** theo mô hình **Client-Server**, đồng thời tổ chức backend theo kiến trúc **MVC (Model-View-Controller)** nhằm tăng tính rõ ràng và dễ bảo trì. Hệ thống bao gồm hai thành phần chính:

#### 1. Frontend (Giao diện người dùng)

- **Ngôn ngữ lập trình:** TypeScript
- **Thư viện và Framework sử dụng:**
  - **ReactJS:** Thư viện xây dựng giao diện người dùng theo kiến trúc *component-based*, giúp tối ưu hiệu suất và khả năng tái sử dụng mã nguồn.
  - **TailwindCSS:** Framework CSS theo hướng *utility-first*, hỗ trợ thiết kế giao diện nhanh chóng, thống nhất và tối ưu responsive trên nhiều thiết bị.
- **Công cụ hỗ trợ phát triển:** *Visual Studio Code*, hệ thống xây dựng dựa trên *Vite* (giúp tối ưu tốc độ khởi động và hot reload).

#### 2. Backend (Xử lý nghiệp vụ và API)

- **Ngôn ngữ lập trình:** TypeScript (chạy trên nền tảng Node.js)
- **Nền tảng:**
  - **Node.js:** Môi trường thực thi JavaScript phía server, giúp xây dựng backend hiệu quả, hỗ trợ lập trình bất đồng bộ (asynchronous).
- **Framework và Thư viện sử dụng:**
  - **ExpressJS:** Framework web cho Node.js, hỗ trợ xây dựng hệ thống API theo mô hình RESTful, quản lý routing và middleware.



- **mysql2**: Thư viện hỗ trợ kết nối và thực thi truy vấn cơ sở dữ liệu MySQL.
- **Cơ sở dữ liệu:**
  - **MySQL**: Hệ quản trị cơ sở dữ liệu quan hệ (RDBMS), lưu trữ toàn bộ dữ liệu của hệ thống.

Backend ExpressJS được tổ chức theo kiến trúc mở rộng từ MVC:

- **Model**: Chịu trách nhiệm thực thi các truy vấn cơ sở dữ liệu và gọi *stored procedures* đã được định nghĩa sẵn trong hệ quản trị cơ sở dữ liệu MySQL.
- **Service**: Xử lý toàn bộ logic nghiệp vụ, điều phối dữ liệu giữa *Controller* và *Model*.
- **Controller**: Tiếp nhận *request* từ frontend, gọi *Service* để thực hiện nghiệp vụ và gửi *response* về cho *client*.
- **View**: Đại diện bởi giao diện người dùng xây dựng bằng ReactJS, hiển thị dữ liệu và cho phép người dùng tương tác với hệ thống.

Kiến trúc MVC giúp phân tách rõ ràng các tầng chức năng, đồng thời đảm bảo khả năng mở rộng và bảo trì cho hệ thống. Đồng thời, việc bổ sung lớp *Service* giúp hệ thống tách biệt rõ ràng giữa các tầng, tăng tính mở rộng, dễ bảo trì và dễ dàng kiểm thử hơn so với mô hình MVC truyền thống.

### 3. Kiến trúc kết nối

- **Backend (ExpressJS)** thực hiện kết nối đến CSDL MySQL để thao tác dữ liệu thông qua thư viện **mysql2**.
- **Frontend (ReactJS)** giao tiếp với **Backend** bằng cách gửi các yêu cầu HTTP (REST API) để hiển thị và cập nhật dữ liệu.

Các thành phần hoạt động độc lập, kết nối với nhau qua giao thức HTTP, đảm bảo khả năng mở rộng, linh hoạt và dễ bảo trì trong tương lai.

## 4.3 Cách kết nối ứng dụng với CSDL MySQL

Trong quá trình phát triển hệ thống, nhóm lựa chọn sử dụng **MySQL** làm hệ quản trị cơ sở dữ liệu chính. Để kết nối ứng dụng với CSDL, nhóm thực hiện các bước sau:

### 4.3.1 Database Driver

Đối với phía backend (ứng dụng Node.js), nhóm sử dụng thư viện **mysql2** — một database driver phổ biến và tối ưu cho MySQL trong môi trường Node.js. Thư viện này hỗ trợ Promise API, giúp việc xử lý truy vấn dễ dàng hơn trong lập trình bất đồng bộ.



```
1 import mysql from 'mysql2/promise';
2 import dotenv from 'dotenv';
3
4 dotenv.config();
5
6 const pool = mysql.createPool({
7   host: process.env.DB_HOST,
8   user: process.env.DB_USER,
9   password: process.env.DB_PASSWORD,
10  port: Number(process.env.DB_PORT),
11  database: process.env.DB_NAME,
12  waitForConnections: true,
13  connectionLimit: 10,
14  queueLimit: 0,
15 });
16
17 export default pool;
```

#### 4.3.2 Cách kết nối

Ban đầu, nhóm đã sử dụng dịch vụ **Railway** để **deploy** cơ sở dữ liệu MySQL lên nền tảng đám mây. Railway cung cấp thông tin kết nối dạng *URI*, ví dụ:

```
mysql://username:password@host:port/database
```

Nhóm lấy thông tin này và sử dụng để kết nối vào phần mềm quản lý cơ sở dữ liệu **MySQL Workbench** để thao tác trực tiếp.

Với các thông tin sau:

```
1 VITE_APP_API_URL='https://dormbk-be-production.up.railway.app'
2 DATABASE_URL="mysql://root:SympYtUKAztewpakanIMZJucyoohxiTMm@hopper.proxy.rlwy.
   net:52246/railway"
3 DB_HOST=hopper.proxy.rlwy.net
4 DB_USER=root
5 DB_PASSWORD=SympYtUKAztewpakanIMZJucyoohxiTMm
6 DB_PORT=52246
7 DB_NAME=railway
8 JWT_SECRET=group08
```

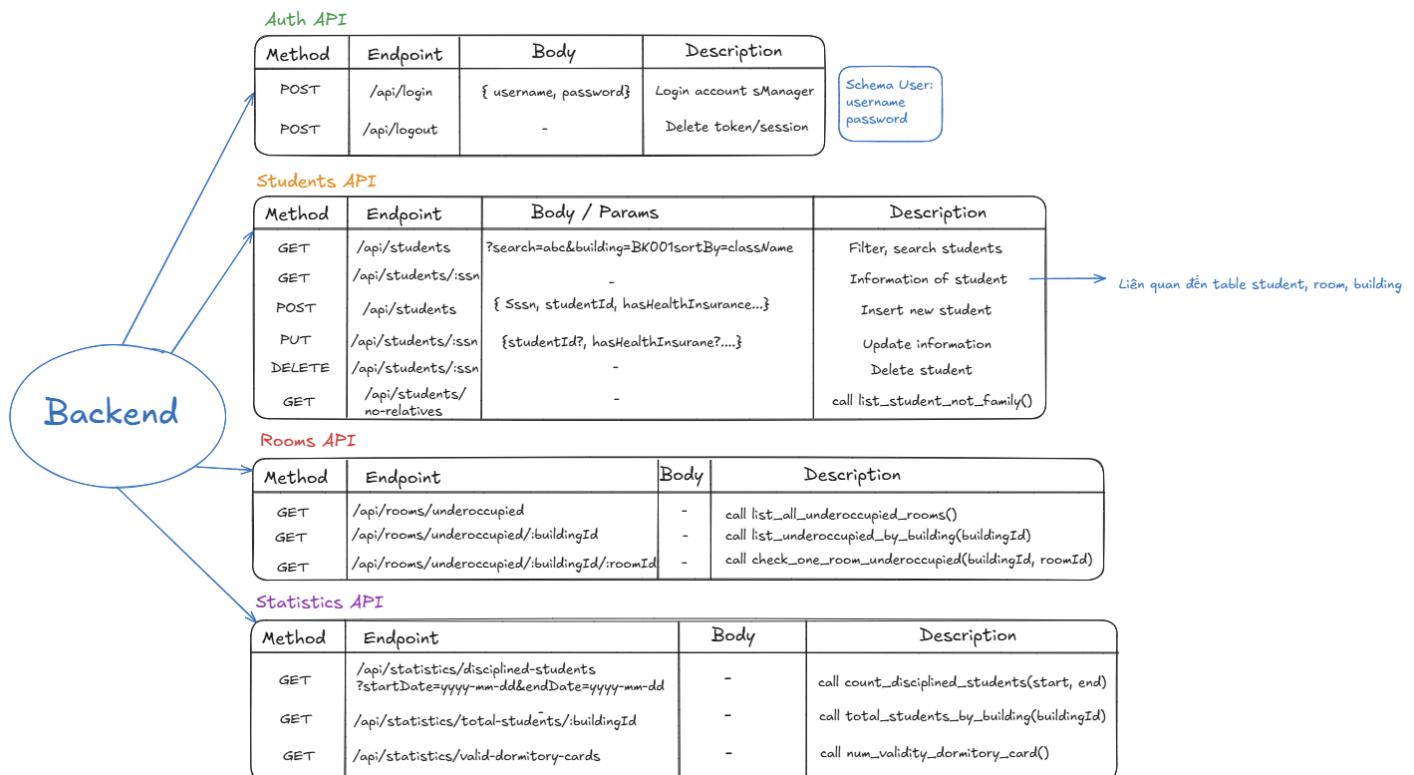


## Kết nối MySQL Workbench

Trong MySQL Workbench:

- Chọn Database → Connect to Database.
- Diền thông tin:
  - Hostname: hopper.proxy.rlwy.net
  - Port: 52246
  - Username: sManager
  - Password: admin123
- Sau đó nhấn OK để kết nối.

## 4.4 Thiết kế hệ thống API



Hình 4.1: Hệ thống API



## 1. Nhóm API xác thực người dùng (Auth API)

Cung cấp chức năng đăng nhập và đăng xuất cho tài khoản quản trị viên (user sManager).

- **POST /api/login:** Đăng nhập bằng tài khoản sManager, truyền username và password.
- **POST /api/logout:** Đăng xuất và xóa session/token đang hoạt động.

## 2. Nhóm API sinh viên (Students API)

Thực hiện các thao tác CRUD đối với sinh viên, đồng thời hỗ trợ truy vấn dữ liệu nâng cao.

- **GET /api/students:** Tìm kiếm, lọc và sắp xếp danh sách sinh viên (có hỗ trợ query parameters như search, building, sortBy).
- **GET /api/students/:ssn:** Lấy thông tin chi tiết của sinh viên theo mã số SSN.
- **POST /api/students:** Thêm mới sinh viên vào hệ thống.
- **PUT /api/students/:ssn:** Cập nhật thông tin sinh viên.
- **DELETE /api/students/:ssn:** Xóa sinh viên khỏi hệ thống.
- **GET /api/students/no-relatives:** Gọi stored procedure list\_student\_not\_family() để lấy danh sách sinh viên không có người thân.

Các API này liên quan đến các bảng student, room, building trong CSDL.

## 3. Nhóm API phòng (Rooms API)

Hỗ trợ truy vấn các phòng đang bị dưới mức sử dụng.

- **GET /api/rooms:** Gọi stored procedure list\_all\_rooms() để lấy danh sách toàn bộ các phòng.
- **GET /api/rooms/:buildingId:**  
Lọc theo tòa nhà bằng list\_rooms\_building(buildingId).
- **GET /api/rooms/underoccupied:**  
Gọi stored procedure list\_all\_underoccupied\_rooms() để lấy danh sách toàn bộ các phòng dưới mức sử dụng.
- **GET /api/rooms/underoccupied/:buildingId:**  
Lọc theo tòa nhà bằng list\_underoccupied\_by\_building(buildingId).

## 4. Nhóm API thống kê (Statistics API)

Thống kê các chỉ số liên quan đến sinh viên và thẻ ký túc xá.



- **GET /api/statistics/disciplined-students?startDate=...&endDate=...:**

Thống kê số lượng sinh viên bị kỷ luật theo khoảng thời gian, gọi hàm `count_disciplined_students()`.

- **GET /api/statistics/total-students/:buildingId:**

Thống kê số sinh viên theo từng tòa nhà, gọi `total_students_by_building()`.

- **GET /api/statistics/valid-dormitory-cards:**

Thống kê số lượng thẻ ký túc xá còn hiệu lực, gọi `num_validity_dormitory_card()`.

Tất cả các API được tổ chức rõ ràng, dễ mở rộng, dễ tích hợp với frontend ReactJS, đồng thời đảm bảo tính bảo mật và kiểm soát quyền truy cập thông qua tài khoản quản trị viên.



## 4.5 Minh họa phần lập trình

### 4.5.1 Mô tả lớp models (services)

```
import pool from '@/Config/db.config';
import { Room } from '@/Interfaces/rooms.interface';

You, 9 hours ago | 2 authors (You and one other)
export class RoomsService {
  constructor() {
    console.log('RoomsService initialized');
  }

  async getAllRooms(): Promise<Room[]> {
    const result = await pool.query('CALL list_all_rooms()');
    const rows = result[0];

    if (Array.isArray(rows) && Array.isArray(rows[0])) {
      return rows[0] as Room[];
    } else {
      throw new Error('Unexpected result format');
    }
  }

  async getRoomsByBuildingId(buildingId: string): Promise<Room[]> {
    if (buildingId.length > 5) {
      throw new Error('Building ID is exactly 5 characters long');
    }
    const result = await pool.query('CALL list_rooms_building(?)', [
      buildingId,
    ]);
    const rows = result[0];

    if (Array.isArray(rows) && Array.isArray(rows[0])) {
      return rows[0] as Room[];
    } else {
      throw new Error('Unexpected result format');
    }
  }

  async getUnderoccupiedRooms(): Promise<Room[]> {
    const result = await pool.query('CALL list_all_underoccupied_rooms()');
    const rows = result[0];

    if (Array.isArray(rows) && Array.isArray(rows[0])) {
      return rows[0] as Room[];
    } else {
      throw new Error('Unexpected result format');
    }
  }

  async getUnderoccupiedRoomsByBuildingId(buildingId: string): Promise<Room[]> {
    if (buildingId.length > 5) {
      throw new Error('Building ID is exactly 5 characters long');
    }
    const result = await pool.query('CALL list_underoccupied_by_building(?)', [
      buildingId,
    ]);
    const rows = result[0];
    You, 9 hours ago | 2 authors (You and one other)
    return rows as Room[];
  }
}
```



#### 4.5.2 Minh họa code chức năng Select

```
static async getAllStudents(): Promise<Student[]> {
  const result = await pool.query('CALL get_student()');
  const rows = result[0];

  if (Array.isArray(rows) && Array.isArray(rows[0])) {
    return rows[0] as Student[];
  } else {
    throw new Error('Unexpected result format');
  }
}
```

NguyenHuukhang, 2 days ago · Handle auth and st

#### 4.5.3 Minh họa code cho hàm hiện thực ở phần 2

```
async getDisciplinedStudents(
  startDate: string,
  endDate: string,
): Promise<DisciplinedStudentsResponse> {
  if (this.isDate(startDate) || !this.isDate(endDate)) {
    throw new Error('Invalid date format');
  }

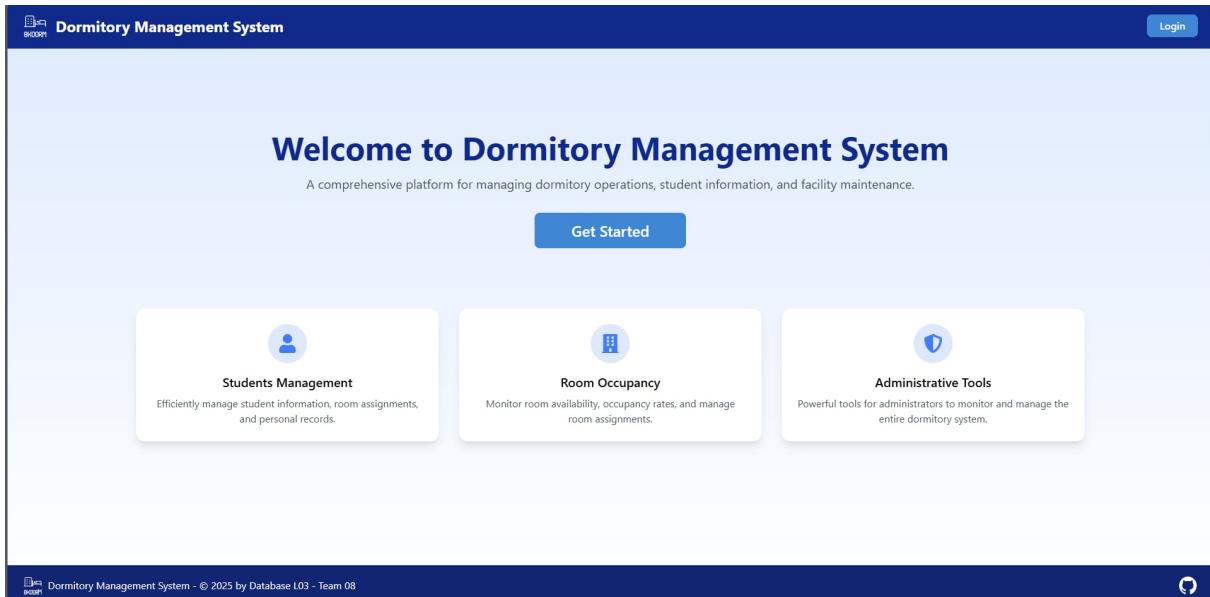
  const result = await pool.query(
    'SELECT count_disciplined_students(?, ?) AS totalDisciplinedStudents',
    [startDate, endDate],
  );

  const totalDisciplinedStudents = result[0] as DisciplinedStudentsResponse[];
  return totalDisciplinedStudents[0];
}
```

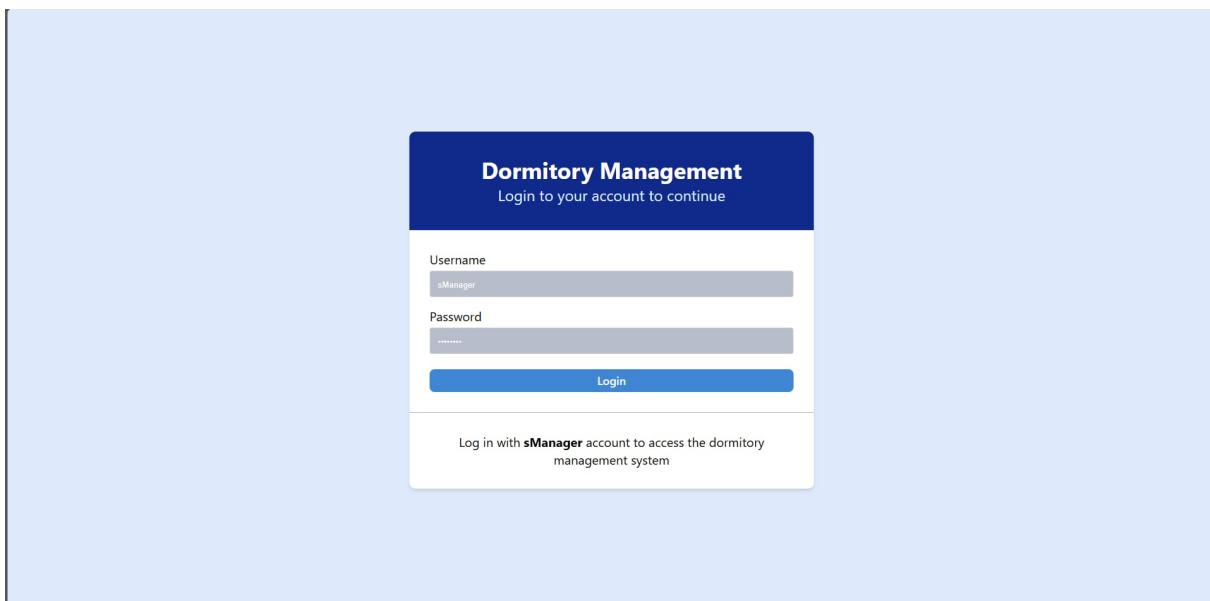


Các procedure/function hiện thực thêm có trong bảng thiết kế API

## 4.6 Thiết kế giao diện web



Hình 4.2: Trang chủ



Hình 4.3: Trang đăng nhập



TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

The screenshot shows the 'Dormitory Management System' dashboard. On the left is a sidebar with 'Dashboard' (selected), 'Students', 'Rooms', and 'Statistics'. The main area has three cards: 'Students Management' (View), 'Room Occupancy' (View), and 'Statistics' (View). Below them is a 'Quick Actions' button labeled 'Add new student'.

Hình 4.4: Trang sau đăng nhập

The screenshot shows the 'Students Management' page of the system. It features a sidebar with 'Dashboard', 'Students' (selected), 'Rooms', and 'Statistics'. The main content area includes sections for 'Students Management' (View), 'Room Occupancy' (View), and 'Statistics' (View). A 'Quick Actions' button for adding new students is also present.

Hình 4.5: Trang quản lý sinh viên



**Dormitory Management System**

- [Dashboard](#)
- [Students](#)
- [Rooms](#)
- [Statistics](#)

**Students Manager**  
All students' information in one place

SSN	Student ID	Full Name	Faculty	Study Status	Room	Building	Actions
05620513	2312613	Khoi Nguyen Minh	Computer Science	Active	P.104	BK001	<a href="#">View</a> <a href="#">Delete</a>
05620514	2212345	M	M	Building	P.104	BK001	<a href="#">View</a> <a href="#">Delete</a>
05620515	2012346	N	N	Building	P.204	BK003	<a href="#">View</a> <a href="#">Delete</a>
05620516	2112347	N	N	Building	P.104	BK001	<a href="#">View</a> <a href="#">Delete</a>
05620517	2312348	N	N	Building	P.102	BK002	<a href="#">View</a> <a href="#">Delete</a>
05620518	2412349	P	P	Building	P.104	BK002	<a href="#">View</a> <a href="#">Delete</a>
05620519	2212350	Phu Nguyen Quoc	Mechanical Engineering	Active	P.203	BK002	<a href="#">View</a> <a href="#">Delete</a>

**Student Information**

SSN: 05620513 Student ID: 2312613

Full Name: Khoi Nguyen Minh Birthday: 2003-06-02

Sex: Male Has health insurance: Yes

Health State: Good Ethnic Group: Kinh

Faculty: Computer Science Class: KHMT1

Study Status: Active Building: BK001

Room: P.104

Addresses: Tam Dan Phu Ninh Quang Nam

Emails: khoi.nguyenminh03@hcmut.edu.vn khoi.nguyen13@gmail.com

Phone Numbers: 0389162347

[Close](#) [Edit](#)

Room	Building	Study Status	Actions
P.104	BK001	Active	<a href="#">View</a> <a href="#">Delete</a>
P.104	BK001	Active	<a href="#">View</a> <a href="#">Delete</a>
P.204	BK003	Active	<a href="#">View</a> <a href="#">Delete</a>
P.104	BK001	Active	<a href="#">View</a> <a href="#">Delete</a>
P.102	BK002	Active	<a href="#">View</a> <a href="#">Delete</a>
P.104	BK002	Active	<a href="#">View</a> <a href="#">Delete</a>
P.203	BK002	Active	<a href="#">View</a> <a href="#">Delete</a>

Hình 4.6: Trang thông tin sinh viên



TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

The screenshot displays the 'Dormitory Management System' interface. On the left, there's a sidebar with 'Dashboard', 'Students' (which is selected), 'Rooms', and 'Statistics'. The main area shows a 'Students Manager' section with a 'Students List' table and a search bar. Two modals are overlaid on the page:

- Edit Student Information Modal:** This modal is centered and contains fields for SSN (05620513), Student ID (2312613), First name (Khoi), Last name (Nguyen Minh), Birthday (06/02/2003), Sex (Male), Has health insurance (Yes), Health State (Good), Ethnic Group (Kinh), Faculty (Computer Science), Class (KHMT1), Study Status (Active), Building (BK001), Room (P.104), and Addresses (Phu Nguyen Quoc). A 'Save' button at the bottom right is highlighted in green.
- Student Details Modal:** This modal is partially visible behind the first one. It shows the same student information but includes additional fields for Addresses (Tam Dan, Phu Ninh, Quang Nam) and Email (khoi.nguyenminh03@hcmut.edu.vn, khoinguyen13@gmail.com). It also has '+ Add Address' and '+ Add Emails' buttons. A 'Save' button at the bottom right is highlighted in green.

On the right side of the main area, there's a table titled 'Active' showing room assignments with columns for Room, Building, Study Status, and Actions (View, Delete).

Hình 4.7: Trang chỉnh sửa thông tin sinh viên



Hình 4.8: Xác nhận/Huỷ chỉnh sửa thông tin sinh viên



TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

The screenshot shows the 'Students Management' section of the system. A modal dialog titled 'Confirm Deletion' asks if the user wants to delete a student with SSN: 05620513. The dialog has 'Cancel' and 'Yes' buttons. In the background, a table lists student records. One record for student 05620513 is highlighted.

SSN	Student ID	Name	Major	Room	Building	Study Status	Actions
05620513	2312613	Khoa	Information Technology	P.104	BK001	Active	<button>View</button> <button>Delete</button>
05620514	2212345	Mai	Information Technology	P.104	BK001	Active	<button>View</button> <button>Delete</button>
05620515	2012346	Ngoc Nguyen Minh	Information Technology	P.204	BK003	Active	<button>View</button> <button>Delete</button>
05620516	2112347	Nam Nguyen Hao	Information Technology	P.104	BK001	Active	<button>View</button> <button>Delete</button>
05620517	2312348	Ngan Pham Kim	Electronics and Telecommunications Engineering	P.102	BK002	Active	<button>View</button> <button>Delete</button>
05620518	2412349	Phuong Tran Ngoc	Electronics and Telecommunications Engineering	P.104	BK002	Active	<button>View</button> <button>Delete</button>
05620519	2212350	Phu Nguyen Quoc	Mechanical Engineering	P.203	BK002	Active	<button>View</button> <button>Delete</button>

Hình 4.9: Xoá sinh viên khỏi ký túc xá

The screenshot shows the 'Rooms Management' section. It displays a table of room details, including Room ID, Building ID, maximum students, current students, occupancy rate, and actions. A button 'Show Underoccupied Rooms' is visible at the top right.

Room ID	Building ID	Max Students	Current Students	Occupancy Rate	Actions
P.104	BK001	6	3	50.00%	<button>View</button>
P.201	BK001	6	0	0.00%	<button>View</button>
P.202	BK001	6	0	0.00%	<button>View</button>
P.203	BK001	6	0	0.00%	<button>View</button>
P.204	BK001	6	3	50.00%	<button>View</button>
P.301	BK001	6	0	0.00%	<button>View</button>
P.302	BK001	6	0	0.00%	<button>View</button>

Hình 4.10: Trang quản lý phòng



**Dormitory Management System**

**Add new student**

Add a new student to the dormitory management system. Required fields are marked with asterisk (\*).

SSN (*):	Student ID (*):
First name (*):	Last name (*):
Birthday (*): mm/dd/yyyy	Sex (*): Select Sex (*)
Has health insurance: No	Health State:
Ethnic Group (*):	Faculty:
Class:	Study Status (*): Select Study Status (*)
Building	Room
mm/dd/yyyy	Select Sex (*)
Has health insurance: No	Health State:
Ethnic Group (*):	Faculty:
Class:	Study Status (*): Select Study Status (*)
Building Select Building	Room
Addresses: Commune + Add Address	Province
Emails: Emails + Add Emails	Phone Numbers: Phone Numbers + Add Phone Numbers
<b>Save</b>	

**Confirm Save**

Are you sure you want to save this student's information?

**Cancel** **Yes**

Hình 4.11: Trang thêm sinh viên vào ký túc xá



## TRƯỜNG ĐẠI HỌC BÁCH KHOA KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

The screenshots show the 'Statistics' section of the system. The left sidebar includes links for Dashboard, Students, Rooms, and Statistics (which is selected). The main content area has a title 'Statistics' with a subtitle 'View statistics about student demographics, department distribution, and more.' Below this are two sections: 'Statistics Options' and 'Disciplined Students'.

**Top Screenshot:** The 'Disciplined Students' section contains two buttons: 'Nhập tay' and 'Chọn lịch'. It has two date input fields: 'Start Date' (YYYY/MM/DD) and 'End Date' (mm/dd/yyyy), both currently showing placeholder text. A 'Get Data' button is at the bottom right.

**Middle Screenshot:** The 'Statistics Options' section shows 'Disciplined Students' and 'Total Students by Building ID' (which is selected). It also shows 'Number of Valid Dorm Cards'. The 'Total Students by Building ID' section has a text input field 'Building ID' with placeholder 'Enter building ID' and a 'Get Data' button.

**Bottom Screenshot:** The 'Statistics Options' section shows 'Disciplined Students', 'Total Students by Building ID' (selected), and 'Number of Valid Dorm Cards' (selected). The 'Number of Valid Dorm Cards' section has a text input field 'Number of valid dorm cards' and a 'Get Data' button.

Each screenshot includes a footer bar with the text 'Dormitory Management System - © 2025 by Database L03 - Team 08' and a circular icon.

Hình 4.12: Trang thống kê cho quản lý



Dormitory Management System

- Dashboard
- Students
- Rooms
- Statistics

### Statistics

View statistics about student demographics, department distribution, and more.

**Statistics Options**

Disciplined Students

Total Students by Building ID

Number of Valid Dorm Cards

**Total Students by Building ID**

Building ID

Get Data

10
TOTAL STUDENTS

■ Dormitory Management System - © 2025 by Database L03 - Team 08

 Dormitory Management System
 

- Dashboard
- Students
- Rooms
- Statistics

### Statistics

View statistics about student demographics, department distribution, and more.

**Statistics Options**

Disciplined Students

Number of Valid Dorm Cards

Total Students by Building ID

**Number of Valid Dorm Cards**

Number of valid dorm cards

Get Data

40
VALID DORMITORY CARDS

■ Dormitory Management System - © 2025 by Database L03 - Team 08
 
 Dormitory Management System
 

- Dashboard
- Students
- Rooms
- Statistics

### Statistics

View statistics about student demographics, department distribution, and more.

**Statistics Options**

Disciplined Students

Total Students by Building ID

Number of Valid Dorm Cards

**Disciplined Students**

Nhập tay Chọn lịch

Start Date

mm/dd/yyyy

End Date

mm/dd/yyyy

Get Data

3
TOTAL DISCIPLINED STUDENTS

■ Dormitory Management System - © 2025 by Database L03 - Team 08
 

Hình 4.13: Trang thống kê với số liệu