

# Hướng dẫn Buổi 2 SQL Sever

## Hướng dẫn tạo CSDL

```
CREATE DATABASE ABC
```

```
USE ABC
```

## Hướng dẫn tạo Tạo Bảng

```
CREATE TABLE table_name (  
    column1 datatype constraint,  
    column2 datatype constraint,  
    ...  
    columnN datatype constraint  
);
```

# Hướng Dẫn Tạo Ràng Buộc

- ❑ Ràng Buộc Khóa Chính(PRIMARY KEY)
- ❑ Ràng Buộc Khóa Duy Nhất(UNIQUE)
- ❑ Ràng Buộc Khóa Ngoại (FOREIGN KEY)
- ❑ Ràng Buộc NOT NULL
- ❑ Ràng Buộc CHECK

# Hướng dẫn Thêm, Sửa, Xóa

- ❑ THÊM : **INSERT INTO** Tên\_Bảng (...)
- ❑ SỬA: **UPDATE** Tên\_Bảng **SET**  
Tên\_Cột\_Cần\_Update **WHERE** Tên\_Khóa Chính
- ❑ Xóa Bảng: **DROP TABLE** Tên\_Bảng

# Câu lệnh ALTER TABLE

- Thêm cột mới vào bảng
- Xóa cột khỏi bảng
- Thay đổi kiểu dữ liệu của cột
- Thay đổi tên của cột
- Thêm ràng buộc vào bảng, chẳng hạn như ràng buộc khóa ngoại
- Xóa ràng buộc khỏi bảng
- Thay đổi tên bảng

VD: ALTER TABLE Bang\_1  
ADD CONSTRAINT CK\_KtraSDT  
CHECK (SDT >= 0)

VD: ALTER TABLE Bang\_1  
ADD CONSTRAINT CK\_KtraSDT  
FOREIGN KEY (Cột Sử dụng làm khóa  
ngoại)  
REFERENCES Bang\_2(Khóa chính)

# ❑ Các mệnh đề trong câu truy vấn:

- ❑ **SELECT:** dùng để chọn ra các cột trong bảng dữ liệu mà muốn hiển thị trong kết quả truy vấn.
  - ❑ Ví dụ: `SELECT name, age, address FROM people`
- ❑ **TOP:** chỉ định số lượng bản ghi tối đa sẽ được trả về. Ví dụ: `SELECT TOP 10 name FROM people;`
- ❑ **DISTINCT:** loại bỏ các giá trị trùng lặp trong kết quả truy vấn.
  - ❑ Ví dụ: `SELECT DISTINCT name FROM people;`
- ❑ **FROM:** chỉ định bảng hoặc các bảng mà muốn truy vấn dữ liệu.
  - ❑ Ví dụ: `SELECT name FROM people WHERE age > 18;`

❑ **WHERE:** dùng để lọc ra các bản ghi thỏa mãn điều kiện được chỉ định.

❑ Ví dụ: `SELECT name FROM people WHERE age > 18;`

❑ **GROUP BY:** nhóm các bản ghi có cùng giá trị trong một hoặc nhiều cột. Ví dụ: `SELECT gender, AVG(age) FROM people GROUP BY gender;`

❑ **HAVING:** chỉ định điều kiện để lọc các bản ghi được nhóm bởi mệnh đề `GROUP BY`.

❑ Ví dụ: `SELECT gender, AVG(age) FROM people GROUP BY gender HAVING AVG(age) > 30;`

❑ **ORDER BY:** sắp xếp các bản ghi theo thứ tự tăng hoặc giảm dần của một hoặc nhiều cột.

❑ Ví dụ: `SELECT name, age FROM people ORDER BY age DESC.`



**ĐỀ MINH HỌA:** Quản lý thông tin Sinh viên và Môn học của một trường đại học.

**Bảng 1:** Sinh viên ( Mã sinh viên (PK), Họ tên, Ngày sinh, Email, Giới tính, Địa chỉ, Số điện thoại)

**Bảng 2:** Môn học ( Mã môn học (PK) ,Tên môn học, Số tín chỉ, Giảng viên)

Mối quan hệ: Nhiều-nhiều giữa Sinh viên và Môn học thông qua bảng trung gian Đăng ký môn học.

**Bảng 3:** Đăng ký môn học (Mã sinh viên (FK), Mã môn học (FK), Điểm)

**\*Truy vấn dữ liệu:**

- 1. Hiển thị thông tin Sinh viên và Môn học có Điểm cao nhất**
- 2. Hiển thị danh sách Sinh viên chưa đăng ký bất kỳ Môn học nào**
- 3. Truy vấn số lượng Sinh viên đăng ký Môn học theo Giảng viên chủ nhiệm, sắp xếp theo thứ tự giảm dần của số lượng Sinh viên**
- 4. Thông tin Sinh viên có điểm trung bình cao nhất và Môn học tương ứng với điểm trung bình đó**
- 5. Tìm kiếm các sinh viên có điểm trung bình môn học lớn hơn 8**

**6. Lấy ra danh sách các môn học đã được đăng ký bởi sinh viên có mã số 1001, bao gồm tên môn học, số tín chỉ và giảng viên giảng dạy, sắp xếp theo tên môn học theo thứ tự tăng dần.**

- + Dùng câu lệnh SELECT để lấy ra thông tin cần hiển thị.
  - + Dùng DISTINCT để loại bỏ các bản ghi trùng lặp.
  - + Dùng các bảng SinhVien, DangKyMonHoc và MonHoc để lấy thông tin về sinh viên, môn học và đăng ký môn học.
  - + Sử dụng INNER JOIN để kết nối các bảng theo khóa chính và khóa ngoại tương ứng.
  - + Dùng WHERE để lọc ra các bản ghi có mã sinh viên là 1001.
- Sử dụng ORDER BY để sắp xếp kết quả theo tên môn học theo thứ tự tăng dần.