ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
UNIVERSITY OF SCIENCE AND TECHNOLOGY - UD

IT FACULTY

CƠ SỞ DỮ LIỆU

TS. Võ Đức Hoàng
Khoa Công nghệ Thông tin
Trường Đại học Bách khoa - Đại học Đà Nẵng

THINKING - CREATING - HUMANITY CHERISHING

TRƯỜNG ĐẠI HỌC BÁCH KHOA TỪ 1975

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
UNIVERSITY OF SCIENCE AND TECHNOLOGY - UD

IT FACULTY

CHƯƠNG 9
SQL cơ bản

THINKING - CREATING - HUMANITY CHERISHING

TRƯỜNG ĐẠI HỌC BÁCH KHOA TỪ 1975

NỘI DUNG

9.1. Khởi tạo và quản lý cơ sở dữ liệu bằng SQL

9.2. Các câu lệnh cơ bản:

- CREATE TABLE, DROP, ALTER
- INSERT, UPDATE, DELETE
- SELECT ... FROM ... WHERE ...

9.3. Thực hành: tạo bảng, thêm dữ liệu, truy vấn đơn giản

9.4. Hướng dẫn sử dụng công cụ SQL Server Management Studio (SSMS)

Mục tiêu của bài học

1. Hiểu và sử dụng được các **lệnh SQL cơ bản** trong định nghĩa và thao tác dữ liệu.
2. Thực hành **tạo bảng, cập nhật, truy vấn dữ liệu** trong SQL Server.
3. Phân biệt và áp dụng các lệnh SELECT, INSERT, UPDATE, DELETE.
4. Làm quen với công cụ **SQL Server Management Studio (SSMS).**

9.1. Khởi tạo và quản lý CSDL bằng SQL

Giới thiệu SQL – Ngôn ngữ truy vấn CSDL quan hệ

- SQL (Structured Query Language) là ngôn ngữ chuẩn để thao tác trên cơ sở dữ liệu quan hệ (RDBMS).
- Là ngôn ngữ cấp cao, người dùng không cần biết cách tổ chức lưu trữ, chỉ cần nêu rõ nội dung cần truy vấn.
- Được phát triển bởi IBM vào những năm 1970s, ban đầu mang tên SEQUEL.
- Trở thành chuẩn công nghiệp do ANSI và ISO công nhận.
- Các phiên bản quan trọng:
  - SQL-86: Chuẩn đầu tiênSQL-92:
  - Chuẩn phổ biến nền tảng cho các hệ QTCSDL hiện nay
  - SQL:1999, SQL:2003, SQL:2011, … SQL:2023

Tại sao SQL phổ biến đến hiện nay?

- Dễ học, cú pháp gần với ngôn ngữ tự nhiên
- Hỗ trợ cả truy vấn dữ liệu (SELECT) và thao tác dữ liệu (INSERT, UPDATE, DELETE)
- Được hỗ trợ bởi hầu hết các hệ QTCSDL: SQL Server, MySQL, PostgreSQL, Oracle, SQLite...
- Chuẩn hóa liên tục, phù hợp với ứng dụng hiện đại (web, cloud, AI, phân tán...)

SQL – Ngôn ngữ thao tác cơ sở dữ liệu

SQL không chỉ là ngôn ngữ truy vấn mà còn hỗ trợ toàn diện các chức năng quản lý CSDL:

- **DDL (Data Definition Language)** – Định nghĩa dữ liệu:
  → CREATE, ALTER, DROP...
- **DML (Data Manipulation Language)** – Thao tác dữ liệu:
  → SELECT, INSERT, UPDATE, DELETE
- **Khung nhìn (View)** – Định nghĩa bảng ảo
  → CREATE VIEW, DROP VIEW
- **Ràng buộc toàn vẹn (Integrity Constraints)**
  → PRIMARY KEY, FOREIGN KEY, CHECK, NOT NULL
- **Phân quyền và bảo mật (Authorization)**
  → GRANT, REVOKE
- **Điều khiển giao tác (Transaction Control)**
  → BEGIN, COMMIT, ROLLBACK

SQL và tương ứng trong mô hình quan hệ

SQL	Mô hình quan hệ
Bảng (Table)	Quan hệ (Relation)
Cột (Column)	Thuộc tính (Attribute)
Dòng (Row)	Bộ (Tuple)

- SQL là ngôn ngữ **cấp cao**, mô tả "cần gì", không yêu cầu "làm như thế nào"

Định nghĩa dữ liệu trong SQL (DDL)

- **DDL (Data Definition Language)** là nhóm lệnh trong SQL dùng để **tạo**, **sửa**, **xóa** cấu trúc dữ liệu trong CSDL.
- Được dùng để định nghĩa:
  - **Cơ sở dữ liệu**
  - **Bảng (table)**
  - **Thuộc tính (column)**
  - **Ràng buộc (constraints)**
  - **Khung nhìn (view)**
- Thường dùng trong giai đoạn **thiết kế lược đồ CSDL**.

Các kiểu dữ liệu trong SQL

SQL	Mô tả thực tế	Ghi chú
INT, INTEGER	Số nguyên	Dùng cho ID, số lượng
DECIMAL(p,s)	Số thực có độ chính xác cố định	VD: tiền (DECIMAL(10,2))
FLOAT, REAL	Số thực dấu phẩy động	Không chính xác tuyệt đối
CHAR(n)	Chuỗi ký tự cố định độ dài	VD: mã quốc gia CHAR(2)
VARCHAR(n)	Chuỗi ký tự có độ dài thay đổi	Phổ biến nhất cho tên, địa chỉ
DATE	Ngày (yyyy-mm-dd)	VD: 2003-12-25
TIME	Giờ trong ngày (hh:mm:ss)	
DATETIME	Ngày + Giờ	
BOOLEAN	TRUE / FALSE	Một số hệ CSDL dùng BIT
TEXT, NTEXT	Chuỗi dài	Dùng cho mô tả, nội dung lớn

- ***Tùy hệ quản trị, cú pháp có thể thay đổi (SQL Server, MySQL, PostgreSQL...)***

Các lệnh định nghĩa dữ liệu (DDL)

Lệnh SQL	Mục đích sử dụng
CREATE TABLE	Tạo bảng mới
CREATE DATABASE	Tạo cơ sở dữ liệu mới
ALTER TABLE	Thêm, sửa, xóa cột hoặc ràng buộc
DROP TABLE	Xóa bảng khỏi CSDL
TRUNCATE TABLE	Xóa nhanh toàn bộ dữ liệu trong bảng
CREATE VIEW	Tạo bảng ảo (khung nhìn)

9.2. Các câu lệnh cơ bản

Tạo cơ sở dữ liệu với CREATE DATABASE

- **Mục đích: Tạo một cơ sở dữ liệu mới để bắt đầu thiết kế hệ thống**
- Cú pháp: CREATE DATABASE <Tên_CSDL>;
- Ví dụ: CREATE DATABASE QL_NHANVIEN;
- Sau khi tạo xong: USE QL_NHANVIEN;

=> Chuyển sang làm việc trong CSDL vừa tạo.

Tạo bảng với CREATE TABLE

- Mục đích: Khởi tạo bảng lưu dữ liệu, định nghĩa cột và kiểu dữ liệu
- Cú pháp

```
CREATE TABLE <Tên_bảng> (
  <Tên_cột> <Kiểu_dữ_liệu> [<Ràng_buộc>],
  ...
  [CONSTRAINT <Tên_ràng_buộc> <LOẠI_RB> (danh sách cột)]
);
```

- Ví dụ:

```
CREATE TABLE NHANVIEN (
  MANV    CHAR(9),
  HONV    VARCHAR(10),
  TENLOT  VARCHAR(20),
  TENNV   VARCHAR(10),
  NGSINH  DATETIME
);
```

Thêm ràng buộc: NOT NULL, DEFAULT, CHECK

- Mục đích: Đảm bảo tính hợp lệ và đầy đủ của dữ liệu nhập vào
- Ví dụ với các ràng buộc:

```
CREATE TABLE NHANVIEN (
  MANV   CHAR(9) PRIMARY KEY,
  TENNV  VARCHAR(20) NOT NULL,
  PHAI   CHAR(3) CHECK (PHAI IN ('Nam', 'Nu')),
  LUONG  INT DEFAULT 10000
);
```

  - NOT NULL: Không được để trống
  - DEFAULT: Giá trị mặc định
  - CHECK: Kiểm tra logic

Ràng buộc UNIQUE, PRIMARY KEY

- Mục đích: Đảm bảo dữ liệu là duy nhất và định danh được từng dòng
- Ví dụ

```
CREATE TABLE PHONGBAN (
 TENPB   VARCHAR(20) UNIQUE,
 MAPHG   INT NOT NULL,
 NG_NHANCHUC DATETIME DEFAULT GETDATE()
);
```

  - UNIQUE: Không trùng lặp
  - PRIMARY KEY: Không NULL và duy nhất

Ràng buộc FOREIGN KEY (Khóa ngoại)

- Mục đích: Thiết lập mối liên kết giữa các bảng
- Ví dụ

```
CREATE TABLE PHANCONG (
 MA_NVIEN CHAR(9) REFERENCES NHANVIEN(MANV),
 SODA     INT REFERENCES DEAN(MADA),
 THOIGIAN DECIMAL(3,1)
);
```

- Đảm bảo giá trị MA_NVIEN tồn tại trong NHANVIEN, SODA trong DEAN.

Đặt tên ràng buộc với CONSTRAINT

- Mục đích: Dễ bảo trì, sửa đổi hoặc xóa các ràng buộc sau này
- Ví dụ:

```
CREATE TABLE NHANVIEN (
  MANV  CHAR(9) CONSTRAINT PK_MANV PRIMARY KEY,
  LUONG INT CONSTRAINT DF_LUONG DEFAULT 10000,
  PHAI  CHAR(3) CONSTRAINT CK_PHAI CHECK (PHAI IN ('Nam', 'Nu'))
);
```

- Giúp quản trị CSDL dễ theo dõi và thao tác.

Khóa chính ghép và khóa ngoại có tên

- Mục đích: Tạo bảng có khóa chính tổ hợp, khóa ngoại rõ tên

```
CREATE TABLE PHANCONG (
 MA_NVIEN CHAR(9),
 SODA    INT,
 THOIGIAN DECIMAL(3,1),
 CONSTRAINT PK_PC PRIMARY KEY (MA_NVIEN, SODA),
 CONSTRAINT FK_PC_MANVIEN FOREIGN KEY (MA_NVIEN) REFERENCES NHANVIEN(MANV),
 CONSTRAINT FK_PC_SODA FOREIGN KEY (SODA) REFERENCES DEAN(MADA)
);
```

- Đáp ứng thiết kế 1 nhân viên – nhiều đề án với giờ làm cụ thể

Lệnh ALTER TABLE: Thêm cột

- Mục đích: Thêm một hoặc nhiều cột mới vào bảng hiện có
- Cú pháp:

```
ALTER TABLE <Tên_Bảng>
ADD <Tên_Cột> <Kiểu_Dữ_Liệu> [<Ràng_Buộc>];
```

- Ví dụ:

```
ALTER TABLE NHANVIEN
ADD EMAIL VARCHAR(100);
```

→ Bảng NHANVIEN sẽ có thêm cột EMAIL.

Lệnh ALTER TABLE: Thêm ràng buộc

- Mục đích: Bổ sung ràng buộc toàn vẹn sau khi bảng đã được tạo
- Cú pháp:

```
ALTER TABLE <Tên_Bảng>
ADD CONSTRAINT <Tên_Ràng_Buộc> <Loại_Ràng_Buộc>;
```

- Ví dụ:

```
ALTER TABLE NHANVIEN
ADD CONSTRAINT CK_LUONG_GT0 CHECK (LUONG > 0);
```

- Lưu ý: Không phải hệ CSDL nào cũng hỗ trợ thêm NOT NULL nếu cột đã có dữ liệu.

Lệnh ALTER TABLE: Xóa cột hoặc ràng buộc

- Mục đích: Xóa cột không dùng hoặc ràng buộc không còn phù hợp
- Xóa cột:

```
ALTER TABLE NHANVIEN
DROP COLUMN EMAIL;
```

- Xóa ràng buộc:

```
ALTER TABLE NHANVIEN
DROP CONSTRAINT CK_LUONG_GT0;
```

- Phải biết tên ràng buộc khi muốn xóa (CONSTRAINT đã đặt tên).

Lệnh DROP TABLE: Xóa bảng

- Mục đích: Xóa toàn bộ bảng và dữ liệu bên trong khỏi cơ sở dữ liệu
- Cú pháp: DROP TABLE <Tên_Bảng>;
- Ví dụ: DROP TABLE PHANCONG;
- Khi xóa bảng:
  - Các khóa ngoại liên kết từ bảng khác sẽ bị ảnh hưởng
  - Không thể hoàn tác nếu không dùng trong giao dịch

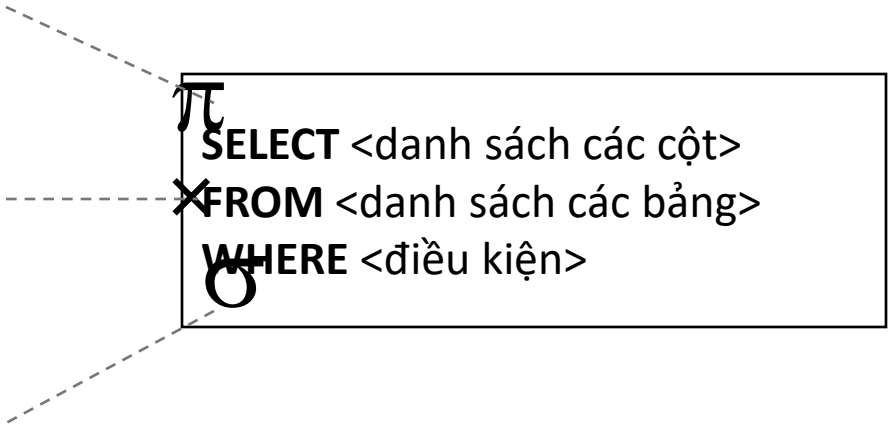
Lệnh DROP TYPE: Xóa kiểu miền người dùng

- Mục đích: Xóa các kiểu dữ liệu tùy chỉnh (user-defined types)
- Cú pháp: DROP TYPE <Tên_Kiểu>;
- Ví dụ: DROP TYPE MaNhanVien;
- Chỉ áp dụng nếu kiểu đó không còn được sử dụng trong bất kỳ bảng nào
- ***Dùng trong trường hợp đã định nghĩa kiểu qua:***

```
CREATE TYPE MaNhanVien FROM CHAR(9);
```

Câu lệnh SELECT: Truy vấn dữ liệu

- SQL và Đại số quan hệ



SELECT L
FROM R
WHERE C

$\pi_L(\sigma_C(R))$ ⟶

Câu lệnh SELECT: Truy vấn dữ liệu (tt)

- Mục đích: Truy xuất dữ liệu từ một hoặc nhiều bảng. Cho phép 1 bảng có nhiều dòng trùng nhau
- Cú pháp cơ bản:

```
SELECT <Danh_sách_cột>
FROM <Tên_bảng>
[WHERE <Điều_kiện>];
```

- Ví dụ: Lọc danh sách nhân viên phòng số 5

$$\sigma_{PHG=5}(NHANVIEN)$$

```
SELECT *
FROM NHANVIEN
WHERE PHG = 5;
```

Lấy tất cả các cột của quan hệ kết quả

Câu lệnh SELECT: với các toán tử lọc

- Mục đích: Lọc dữ liệu chi tiết hơn với điều kiện nâng cao
- Các toán tử thường dùng:
  - So sánh: =, <>, >, <, >=, <=
  - Phủ định: NOT
  - Phạm vi: BETWEEN, IN
  - Mẫu chuỗi: LIKE 'A%', LIKE '_an'
- Ví dụ:

```
SELECT *
FROM NHANVIEN
WHERE LUONG BETWEEN 8000 AND 15000;
```

Câu lệnh SELECT: với các toán tử lọc (tt)

- Ví dụ: Hiển thị thông tin MANV, HONV, TENLOT, TENNV của những nhân viên có PHG=5 và PHAI='Nam'

$$\pi_{MANV,HONV,TENLOT,TENNV}(\sigma_{PHG=5 \wedge PHAI='Nam'}(NHANVIEN))$$

```
SELECT MANV, HONV, TENLOT, TENNV
FROM NHANVIEN
WHERE PHG=5 AND PHAI='Nam'
```

MANV	HONV	TENLOT	TENNV
333445555	Nguyen	Thanh	Tung
987987987	Nguyen	Manh	Hung

Câu lệnh SELECT: với các toán tử lọc (tt)

- Tên bí danh

```
SELECT MANV, HONV AS HO, TENLOT AS 'TEN LOT', TENNV AS TEN
FROM NHANVIEN
WHERE PHG=5 AND PHAI='Nam'
```

MANV	HO	TEN LOT	TEN
333445555	Nguyen	Thanh	Tung
987987987	Nguyen	Manh	Hung

$$\rho_{MANV,HO,TEN\ LOT,TEN}(\pi_{MANV,HONV,TENLOT,TENNV}(\sigma_{PHG=5 \wedge PHAI='Nam'}(NHANVIEN)))$$

Câu lệnh SELECT: với các toán tử lọc (tt)

- Mở rộng

```
SELECT MANV, HONV + ' ' + TENLOT + ' ' + TENNV AS 'HO TEN'
FROM NHANVIEN
WHERE PHG=5 AND PHAI='Nam'
```

MANV	HO TEN
333445555	Nguyen Thanh Tung
987987987	Nguyen Manh Hung

$$\rho_{MANV,HO\ TEN}(\pi_{MANV,HONV+TENLOT+TENNV}(\sigma_{PHG=5 \wedge PHAI='Nam'}(NHANVIEN)))$$

Câu lệnh SELECT: với các toán tử lọc (tt)

- Mở rộng

```
SELECT MANV, LUONG*1.1 AS 'LUONG10%'
FROM NHANVIEN
WHERE PHG=5 AND PHAI='Nam'
```

MANV	LUONG10%
333445555	33000
987987987	27500

$$\rho_{MANV,LUONG10\%}(\pi_{MANV,LUONG*1.1}(\sigma_{PHG=5 \wedge PHAI='Nam'}(NHANVIEN)))$$

Câu lệnh SELECT: với rút gọn

- Mục đích: Loại bỏ các dòng trùng nhau

```
SELECT DISTINCT LUONG
FROM NHANVIEN
WHERE PHG=5 AND PHAI='Nam'
```

LUONG
30000
25000
28000
38000

- Tốn chi phí
- Người dùng muốn thấy

Câu lệnh SELECT: với sắp xếp

- Mục đích: Sắp xếp kết quả mệnh đề ORDER BY được sử dụng để sắp xếp kết quả theo thứ tự tăng dần hoặc giảm dần
- Cú pháp:

```
SELECT * FROM <Bảng>
ORDER BY <Cột> [ASC|DESC];
```

- *Nếu không chọn cụ thể ASC hay DESC trong mệnh đề ORDER BY thì kết quả sẽ mặc định được sắp xếp theo thứ tự tăng dần*
- Ví dụ:

```
SELECT TENNV, LUONG FROM NHANVIEN ORDER BY LUONG DESC;
```

Câu lệnh INSERT: Thêm dữ liệu

- Mục đích: Thêm bản ghi mới vào bảng
- Cú pháp:

```
INSERT INTO <Tên_Bảng> (<Cột1>, <Cột2>, ...)
VALUES (<Giá_trị1>, <Giá_trị2>, ...);
```

- Ví dụ:

```
INSERT INTO PHONGBAN (TENPB, MAPHG)
VALUES ('Kế toán', 7);
```

- *Phải đảm bảo đúng thứ tự và kiểu dữ liệu.*

Câu lệnh UPDATE: Cập nhật dữ liệu

- Mục đích: Thay đổi dữ liệu đã có trong bảng
- Cú pháp:

```
UPDATE <Tên_Bảng>
SET <Cột> = <Giá_trị_mới>, ...
WHERE <Điều_kiện>;
```

- Ví dụ:

```
UPDATE NHANVIEN
SET LUONG = LUONG + 1000
WHERE PHG = 5;
```

- ⚠ *Nếu không có WHERE, tất cả các dòng sẽ bị cập nhật!*

Câu lệnh DELETE: Xóa dữ liệu

- Mục đích: Xóa bản ghi khỏi bảng
- Cú pháp:

```
DELETE FROM <Tên_Bảng>
WHERE <Điều_kiện>;
```

- Ví dụ:

```
DELETE FROM NHANVIEN
WHERE LUONG < 5000;
```

- *Nếu thiếu WHERE, toàn bộ bảng sẽ bị xóa!*

Tổng kết DML (INSERT, UPDATE, DELETE)

Lệnh	Mục đích	Ghi chú
INSERT	Thêm bản ghi mới	Chú ý đúng kiểu và thứ tự cột
UPDATE	Cập nhật dữ liệu hiện có	Luôn dùng WHERE để tránh lỗi
DELETE	Xóa bản ghi	Có thể kết hợp điều kiện nhiều bảng