

## **CHƯƠNG 5**

# **SQL (Structured Query Language)**

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

- Ngôn ngữ ĐSQH
  - Cách thức truy vấn dữ liệu
  - Khó khăn cho người sử dụng
- SQL (Structured Query Language)
  - Ngôn ngữ cấp cao
  - Người sử dụng chỉ cần đưa ra nội dung cần truy vấn
  - Được phát triển bởi IBM (1970s)
  - Được gọi là SEQUEL
  - Được ANSI công nhận và phát triển thành chuẩn
    - SQL-86
    - SQL-92
    - SQL-99

## ■ SQL gồm

- Định nghĩa dữ liệu (DDL)
- Thao tác dữ liệu (DML)
- Định nghĩa khung nhìn
- Ràng buộc toàn vẹn
- Phân quyền và bảo mật
- Điều khiển giao tác

**Lý thuyết : Chuẩn SQL-92**

## ■ SQL sử dụng thuật ngữ

- Bảng ~ quan hệ
- Cột ~ thuộc tính
- Dòng ~ bộ

**Thương mại : SQL Server**

- Giới thiệu
- **Định nghĩa dữ liệu**
  - Kiểu dữ liệu
  - Các lệnh định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

- Là ngôn ngữ mô tả
  - Lược đồ cho mỗi quan hệ
  - Miền giá trị tương ứng của từng thuộc tính
  - Ràng buộc toàn vẹn
  - Chỉ mục trên mỗi quan hệ
- Gồm
  - CREATE TABLE (tạo bảng)
  - DROP TABLE (xóa bảng)
  - ALTER TABLE (sửa bảng)
  - CREATE DOMAIN (tạo miền giá trị)
  - CREATE DATABASE

- Số (numeric)
  - INTEGER
  - SMALLINT
  - NUMERIC, NUMERIC(p), NUMERIC(p,s)
  - DECIMAL, DECIMAL(p), DECIMAL(p,s)
  - REAL
  - DOUBLE PRECISION
  - FLOAT, FLOAT(p)

- Chuỗi ký tự (character string)
  - CHARACTER hoặc CHAR
  - CHARACTER(n) hoặc CHAR (n)
  - CHARACTER VARYING(n) hoặc VARCHAR(n)
  - NATIONAL CHARACTER (n) hoặc NCHAR(n)
- Chuỗi bit (bit string)
  - BIT, BIT(x)
  - BIT VARYING(x)
- Ngày giờ (datetime)
  - DATE gồm ngày, tháng và năm
  - TIME gồm giờ, phút và giây
  - TIMESTAMP gồm ngày và giờ
  - DATETIME: gồm DATE và TIME (Chỉ có trong SQL của SQL Server)



- Để định nghĩa một bảng
  - Tên bảng
  - Các thuộc tính
    - Tên thuộc tính
    - Kiểu dữ liệu
    - Các RBTV trên thuộc tính
- Cú pháp

```
CREATE TABLE <Tên_bảng> (  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],  
    ...  
    [<RBTV>]  
)
```

- Tạo bảng lưu trữ thông tin giáo viên với các thuộc tính và quy định sau:
  - **MaGV** : Khoá chính
  - **HoTen**: Họ tên của giáo viên không được để trống
  - **Luong**: Mặc định là 1000
  - **Phai**: Phải là Nam hoặc Nữ
  - **NgaySinh**: Ngày tháng năm sinh của giáo viên
  - **SoNha, Duong, Quan, ThanhPho** : địa chỉ của giáo viên
  - **GVQLCM**: mã giáo viên quản lý chuyên môn của giáo viên (khoá ngoại)
  - **MaBM**: mã bộ môn của giáo viên (khoá ngoại)

```
CREATE TABLE GIAOVIEN (  
    MaGV          CHAR(9),  
    HoTen         NVARCHAR(50),  
    Luong         INT,  
    Phai          CHAR(3),  
    NgaySinh      DATETIME,  
    SoNha         NVARCHAR(10),  
    Duong NVARCHAR(50),  
    Quan          NVARCHAR(50),  
    ThanhPho      NVARCHAR(50),  
    GVQLCM        CHAR(9),  
    MaBM          CHAR(9)  
)
```

- <RBTV>
  - NOT NULL
  - NULL
  - UNIQUE
  - DEFAULT
  - PRIMARY KEY
  - FOREIGN KEY / REFERENCES
  - CHECK
  
- Đặt tên cho RBTV

**CONSTRAINT** <Ten\_RBTV> <RBTV>

```
CREATE TABLE GIAOVIEN (  
    MAGV          CHAR(9) PRIMARY KEY,  
    HOTEN         NVARCHAR(50) NOT NULL,  
    LUONG INT DEFAULT (1000),  
    PHAI          CHAR(3) CHECK (PHAI IN( 'Nam',  
    'Nu' )),  
    NGAYSINH      DATETIME,  
    SONHA NVARCHAR(10),  
    DUONG NVARCHAR(50),  
    QUAN          NVARCHAR(50),  
    THANHPHO      NVARCHAR(50),  
    GVQLCM        CHAR(9),  
    MABM          CHAR(9)
```

```
CREATE TABLE CONGVIEC (  
    MADT  VARCHAR(10),  
    STT    INT,  
    TENCV NVARCHAR(50),  
    NGAYBD DATETIME,  
    NGAYKT DATETIME,  
    PRIMARY KEY(MADT, STT)  
)
```

```
CREATE TABLE DETAI (  
    MADT  VARCHAR(10) PRIMARY KEY,  
    TENDT NVARCHAR(50) UNIQUE,  
    KINHPHI INT,  
    CAPQL NVARCHAR(50),  
    NGAYBD DATETIME,  
    NGAYKT DATETIME,  
    MACD  VARCHAR(10),  
    GVCNDT CHAR(9)
```

# Ví dụ - Đặt tên cho RBTV



```
CREATE TABLE GIAOVIEN (  
    MAGV          CHAR(9) CONSTRAINT PK_GV PRIMARY KEY,  
    HOTEN         NVARCHAR(50) CONSTRAINT NN_HOTEN NOT NULL,  
    LUONG         INT CONSTRAINT DE_LUONG DEFAULT (10000),  
    PHAI          CHAR(3) CONSTRAINT CK_PHA1 CHECK (PHAI IN('Nam', 'Nu'))  
    CONSTRAINT NN_PHA1 NOT NULL,  
    NGAYSINH      DATETIME,  
    SONHA         NVARCHAR(10),  
    DUONG         NVARCHAR(50),  
    QUAN          NVARCHAR(50),  
    THANHPHO      NVARCHAR(50),  
    GVQL          CHAR(9),  
    MABM          CHAR(9)
```

# Ví dụ - Đặt tên cho RBTV



```
CREATE TABLE CONGVIEC(  
    MADT  VARCHAR(10),  
    STT    INT,  
    TENCV NVARCHAR(50),  
    NGAYBD DATETIME,  
    NGAYKT DATETIME,  
  
    CONSTRAINT PK_CV PRIMARY KEY(MADT, STT),  
    CONSTRAINT FK_CONGVIEC_DETAI  
        FOREIGN KEY MADT REFERENCES DETAI(MADT)  
)  
)
```



- Được dùng để
  - Thay đổi cấu trúc bảng
  - Thay đổi RBTV

- Thêm cột

```
ALTER TABLE <Tên_bảng> ADD  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>]
```

- Xóa cột

```
ALTER TABLE <Tên_bảng> DROP COLUMN <Tên_cột>
```

- Cập nhật cột

```
ALTER TABLE <Tên_bảng> ALTER COLUMN  
    <Tên_cột> <Kiểu_dữ_liệu_mới>
```

- Thêm RBTV

```
ALTER TABLE <Tên_bảng> ADD  
    CONSTRAINT <Ten_RBTV> <RBTV>,  
    CONSTRAINT <Ten_RBTV> <RBTV>,  
    ...
```

- Xóa RBTV

```
ALTER TABLE <Tên_bảng> DROP <Tên_RBTV>
```

# Ví dụ - Thay đổi cấu trúc bảng



```
ALTER TABLE GIAOVIEN ADD TUOI INT  
CONSTRAINT CK_TUOI CHECK (TUOI >= 23 AND TUOI <=60) NOT NULL
```

```
ALTER TABLE GIAOVIEN DROP COLUMN HOTEN
```

```
ALTER TABLE GIAOVIEN ALTER COLUMN HOTEN  
NVARCHAR(100)
```

# Ví dụ - Thay đổi RBTV



```
CREATE TABLE BOMON(
```

```
    MABM          INT NOT NULL,  
    TENBM         NVARCHAR(50),  
    PHONG         CHAR(10),  
    DIENTHOAI     CHAR(15),  
    TRUONGBM      CHAR(9),  
    MAKHOA        CHAR(4),  
    NGAYNHANCHUC DATETIME
```

*Phải xác định NOT NULL  
trước khi tạo khóa chính*

```
)
```

```
ALTER TABLE BOMON ADD
```

```
    CONSTRAINT PK_BOMON PRIMARY KEY (MABM),
```

```
    CONSTRAINT FK_TRBOMON FOREIGN KEY (TRUONGBM)  
        REFERENCES GIAOVIEN(MAGV),
```

```
    CONSTRAINT UNI_TENBM UNIQUE(TENBM),
```

```
    CONSTRAINT DF_NGAYNHANCHUC DEFAULT(GETDATE()) FOR NGAYNHANCHUC
```

*Cần có bảng  
GIAOVIEN trước*

- Được dùng để xóa cấu trúc bảng
  - Tất cả dữ liệu của bảng cũng bị xóa

- Cú pháp

**DROP TABLE** <Tên\_bảng>

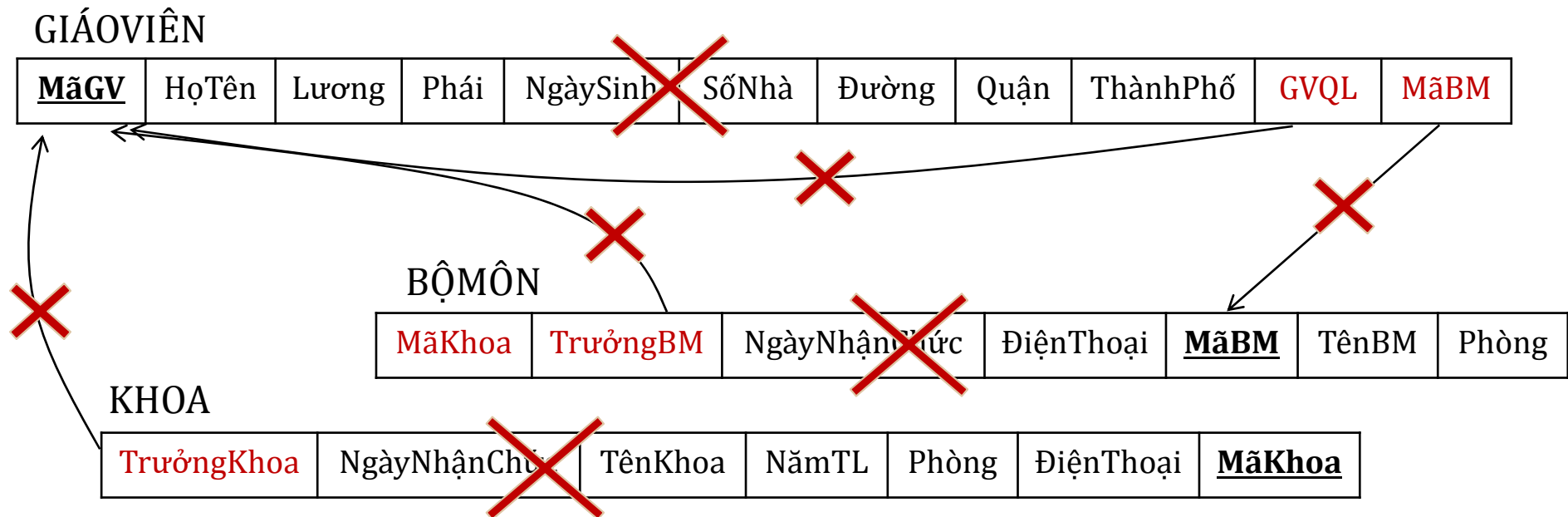
- Ví dụ

**DROP TABLE GIAOVIEN**

**DROP TABLE BOMON**

**DROP TABLE THAMGIADT**

# Lệnh xóa bảng (tt)



- Tạo ra một kiểu dữ liệu mới kế thừa những kiểu dữ liệu có sẵn
- Cú pháp

```
CREATE DOMAIN <Tên_kdl_mới> AS <Kiểu_dữ_liệu>
```

- Ví dụ

```
CREATE DOMAIN MyString30 AS VARCHAR(30)
```

- Giới thiệu
- Định nghĩa dữ liệu
- **Truy vấn dữ liệu**
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - Hàm kết hợp và gom nhóm
  - Một số kiểu truy vấn khác
- Cập nhật dữ liệu
- Khung nhìn (view)
- **Chỉ mục (index)**

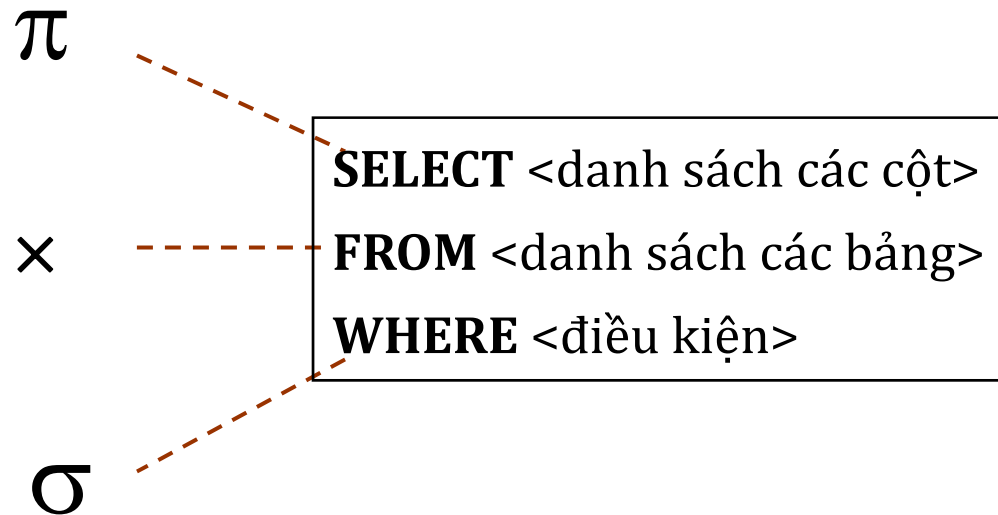


- Là ngôn ngữ rút trích dữ liệu thỏa một số điều kiện nào đó
- Gồm 3 mệnh đề

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>
```

- <danh sách các cột>
  - Tên các cột cần được hiển thị trong kết quả truy vấn
- <danh sách các bảng>
  - Tên các bảng liên quan đến câu truy vấn
- <điều kiện>
  - Biểu thức boolean xác định dòng nào sẽ được rút trích
  - Nối các biểu thức: AND, OR, và NOT
  - Phép toán: <, >, ≤, ≥, ≠, =, LIKE và BETWEEN

- SQL và ĐSQH



**SELECT L**  
**FROM R**  
**WHERE C**  $\rightarrow \pi_L (\sigma_C (R))$

# Mệnh đề SELECT



Lấy ra tất cả các cột

SELECT \*

FROM KHOA

WHERE PHONG='I53'

AND NAMTL = '1995'

Lấy tất cả các  
cột của quan hệ  
kết quả

MaKhoa	TenKhoa	Phong	NamTL	DienThoai	TruongKhoa	NgayNhanChuc
CNTT	Công nghệ thông tin	I53	1995	08313964145	GV130	01/01/2007
SH	Sinh học	B32	1975	08313123545	GV250	01/01/1990

$\sigma_{PHG='I53' \wedge NamTL='1995'}(KHOA)$

# Mệnh đề SELECT (tt)



Lấy ra một số cột

```
SELECT MAKHOA, TENKHOA, PHONG  
FROM KHOA  
WHERE PHONG='I53'  
AND NAMTL = '1995'
```

MaKhoa	TenKhoa	Phong
CNTT	Công nghệ thông tin	I53

$$\pi_{\text{MAKHOA, TENKHOA, PHONG}}(\sigma_{\text{PHG}='I53' \wedge \text{NamTL}='1995'}(\text{KHOA}))$$

# Mệnh đề SELECT (tt)



## Tên bí danh

```
SELECT MAKHOA AS 'Mã khoa', TENKHOA AS 'Tên khoa', PHONG AS 'Mã phòng'  
FROM KHOA  
WHERE PHONG='I53' AND NAMTL = '1995'
```

Mã khoa	Tên khoa	Mã phòng
CNTT	Công nghệ thông tin	I53

$\rho_{\text{Mã khoa, Tên khoa, Mã phòng}}(\pi_{\text{MANV, HONV, TENLOT, TENNV}}(\sigma_{\text{PHG='I53' \wedge NamTL='1995'}}(\text{KHOA})))$

# Mệnh đề SELECT (tt)



## Mở rộng

```
SELECT MAGV, HOTEN, SONHA + ',' + DUONG + ',' + ',' + QUAN + ',' +  
THANHPHO AS 'DIA CHI'  
FROM GIAOVIEN  
WHERE PHAI='Nam'
```

MAGV	HOTEN	DIA CHI
GV001	Nguyễn Văn A	123 Phan Đăng Lưu, Q.Phú Nhuận, TP.Hồ Chí Minh

$\rho_{\text{MAGV,HOTEN,DIA CHI}}(\pi_{\text{MAGV,HOTEN,SONHA+DUONG+QUAN+THANHPHO}}(\sigma_{\text{PHAI='Nam'}}(\text{GIAOVIEN})))$

# Mệnh đề SELECT (tt)



## Mở rộng

```
SELECT MAGV, LUONG*1.1 AS 'LUONG10%'  
FROM GIAOVIEN  
WHERE PHAI='Nam'
```

MAGV	LUONG10%
GV001	550000

$$\rho_{\text{MAGV,LUONG10\%}}(\pi_{\text{MAGV,LUONG*1.1}}(\sigma_{\text{PHAI='Nam'}}(\text{GIAOVIEN})))$$

# Mệnh đề SELECT (tt)



Loại bỏ các dòng trùng nhau

```
SELECT LUONG  
FROM GIAOVIEN  
WHERE PHAI= 'Nam'
```

<u>LUONG</u>
30000
25000
25000
38000



```
SELECT DISTINCT LUONG  
FROM GIAOVIEN  
WHERE PHAI= 'Nam'
```

<u>LUONG</u>
30000
25000
38000

- Tổn chi phí
- Người dùng muốn thấy



- Cho biết MAGV và TENGV làm việc ở bộ môn ‘Hệ thống thông tin’

$R1 \leftarrow \text{GIAOVIEN} \bowtie_{\text{GIAOVIEN.MABM}=\text{BOMON.MABM}} \text{BOMON}$

$KQ \leftarrow \pi_{\text{MAGV, HOTEN}} (\sigma_{\text{TENBM}=\text{'Hệ thống thông tin'}} (R1))$

```
SELECT  MAGV, HOTEN
FROM    GIAOVIEN, BOMON
WHERE   TENBM= N'Hệ thống thông tin'
        AND  GIAOVIEN.MABM=BOMON.MABM
```

# Mệnh đề WHERE



- Sử dụng các phép toán AND, OR để nối các biểu thức điều kiện lại với nhau

**Biểu thức luận lý**

```
SELECT MAGV, HOTEN
FROM GIAOVIEN, BOMON
WHERE TENBM = N'Hệ thống thông tin'
AND GIAOVIEN.MABM = BOMON.MABM
```

TRUE  
AND  
TRUE

# Mệnh đề WHERE (tt)



## Độ ưu tiên

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN, BOMON  
WHERE (TENBM = N'Hệ thống thông tin' OR TENBM = N'Mạng máy tính')  
AND GIAOVIEN.MABM = BOMON.MABM
```

Độ ưu tiên mặc định của các phép toán: từ trái qua phải.

**Nên** sử dụng dấu ngoặc thể hiện tường minh sự ưu tiên của các phép toán.

# Mệnh đề WHERE (tt)



## BETWEEN

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE LUONG >= 20000 AND LUONG <= 30000
```

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE LUONG BETWEEN 20000 AND 30000
```

# Mệnh đề WHERE (tt)



NOT BETWEEN

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE LUONG NOT BETWEEN 20000 AND  
30000
```



```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE LUONG < 20000 OR LUONG > 30000
```

# Mệnh đề WHERE (tt)



## LIKE

SELECT MAGV, HOTEN

FROM GIAOVIEN

WHERE HOTEN LIKE 'Nguyen \_ \_ \_ \_'



HOTEN LIKE 'nguyen \_ \_ \_ \_'

4 ký tự bất kỳ

SELECT MAGV, HOTEN

FROM GIAOVIEN

WHERE HOTEN LIKE 'Nguyen %'

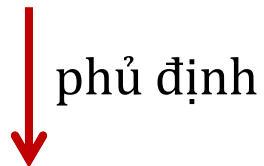
Chuỗi bất kỳ

# Mệnh đề WHERE (tt)



NOT LIKE

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE HOTEN LIKE 'Nguyen'
```



```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE HOTEN NOT LIKE 'Nguyen'
```

# Mệnh đề WHERE (tt)



**charlist**

MAGV	HOTEN
GV001	Nguyễn Văn A
GV002	Hoàng Thị C

SELECT \*  
FROM GIAOVIEN GV  
WHERE GV.HOTEN LIKE N'[n]%'



MAGV	HOTEN
GV001	Nguyễn Văn A

SELECT \*  
FROM GIAOVIEN GV  
WHERE GV.HOTEN LIKE N'[nh]%'



MAGV	HOTEN
GV001	Nguyễn Văn A
GV002	Hoàng Thị C



# Mệnh đề WHERE (tt)



**charlist**

MAGV	HOTEN
GV001	Nguyễn Văn A
GV002	Hoàng Thị C

SELECT \*  
FROM GIAOVIEN GV  
WHERE GV.HOTEN LIKE N'^h]%'



MAGV	HOTEN
GV001	Nguyễn Văn A

SELECT \*  
FROM GIAOVIEN GV  
WHERE GV.HOTEN LIKE N'^nh]%'



MAGV	HOTEN

# Mệnh đề WHERE (tt)



Ngày giờ

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE NGAYSINH BETWEEN '1955-12-08' AND '1966-07-19'
```

'1955-12-08'	YYYY-MM-DD	' 17:30:00'	HH:MI:SS
' 12/08/1955'	MM/DD/YYYY	' 05:30 PM'	
'December 8, 1955'			

'1955-12-08 17:30:00'

# Mệnh đề WHERE (tt)



## NULL

- Sử dụng trong trường hợp:
  - Không biết (value unknown)
  - Không thể áp dụng (value inapplicable)
  - Không tồn tại (value withheld)
- Những biểu thức tính toán có liên quan đến giá trị NULL sẽ cho ra kết quả là NULL
  - $x$  có giá trị là NULL
  - $x + 3$  cho ra kết quả là NULL
- Những biểu thức so sánh có liên quan đến giá trị NULL sẽ cho ra kết quả là UNKNOWN
  - $x = 3$  cho ra kết quả là UNKNOWN
  - $x = 3$  là một so sánh không hợp lệ trong SQL

# Mệnh đề WHERE (tt)



NULL

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE GVQL IS NULL
```

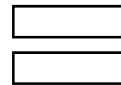
```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE GVQL IS NOT NULL
```

# Mệnh đề FROM



Mệnh đề FROM tương đương với phép  $\times$  trong đại số quan hệ

GIAOVIEN  $\times$  BOMON



SELECT \*

FROM GIAOVIEN, BOMON

SELECT \*

FROM GIAOVIEN, BOMON

WHERE TRUE

MAGV	MAB		MABM	TENBM
001	MHTTT		HTTT	Hệ thống thông tin
001	HTTT		MTT	Mạng máy tính
001	HTTT		CNPM	Công nghệ PM
002	MMT		HTTT	Hệ thống thông tin
002	MMT		MTT	Mạng máy tính
002	MMT		CNPM	Công nghệ PM
...	...		...	...



Sử dụng thêm điều kiện ở WHERE để biểu diễn phép kết

GIAOVIEN  $\bowtie$  BOMON

**C**

SELECT \*

FROM GIAOVIEN, BOMON

WHERE **C**

# Mệnh đề FROM (tt)



Tên bí danh

Nhập nhằng (Ambiguous) ???

MAGV	MAB	MABM	TENBM
001	MHTTT	HTTT	Hệ thống thông tin
001	HTTT	MTT	Mạng máy tính
001	HTTT	CNPM	Công nghệ PM
002	MMT	HTTT	Hệ thống thông tin
002	MMT	MTT	Mạng máy tính
002	MMT	CNPM	Công nghệ PM
...	...	...	...

MAGV	MABM	TENBM
001	HTTT	Hệ thống thông tin
002	MTT	Mạng máy tính

```
SELECT MAGV, MABM, TENBM
FROM GIAOVIEN, BOMON
WHERE MABM = MABM
```

```
SELECT G.MAGV, G.MABM, B.TENBM
FROM GIAOVIEN G, BOMON AS B
WHERE G.MABM = B.MABM
```

Đặt bí danh cho bảng sử dụng khoảng trắng hoặc từ khoá AS

Điều kiện kết để tìm ra bộ môn của giáo viên

# Ví dụ 1



- Cho biết tên của bộ môn và tên của trưởng bộ môn của những bộ môn thuộc khoa CNTT (mã khoa)

```
SELECT BM.TENBM, GV.HOTEN AS TEN_TRUONGBM  
FROM BOMON BM, GIAOVIEN GV  
WHERE BM.TRUONGBM = GV.MAGV AND  
       BM.MAKHOA= 'CNTT'
```

Điều kiện kết để tìm ra giáo viên làm trưởng bộ môn

## Ví dụ 2



- Với những đề tài thuộc cấp quản lý ‘Thành phố’, cho biết mã đề tài, đề tài thuộc về chủ đề nào, họ tên người chủ nghiệm đề tài cùng với ngày sinh và địa chỉ của người ấy

```
SELECT D.MADT, C.TENCD, G.MAGV, G.HOTEN, G.DIACHI  
FROM DETAI D, CHUDE C, GIAOVIEN G  
WHERE D.CapQL = 'Thanh Pho' AND D.MACD = C.MACD AND  
D.GVCNDT = G.MAGV
```



- Tìm họ tên của giáo viên viên thuộc bộ môn “HTTT” có tham gia vào đề tài “ Mobile Database” với số tiền phụ cấp cho mỗi công việc trên 10 triệu.

```
SELECT GV.HOTEN  
FROM GIAOVIEN GV, THAMGIADT TG, DETAI DT  
WHERE GV.MAGV = TG.MAGV AND  
      TG.MADT = DT.MADT AND  
      GV.MABM='HTTT' AND  
      DT.TENDT='Mobile Database' AND TG.PHUCAP>10
```

## Ví dụ 4



- Tìm họ tên của từng giáo viên và người phụ trách chuyên môn trực tiếp của nhân viên đó.

## Ví dụ 5



- Tìm họ tên của những giáo viên được “Trần Trà Hương” phụ trách quản lý chuyên môn.

# Mệnh đề ORDER BY



- Dùng để hiển thị kết quả câu truy vấn theo một thứ tự nào đó
- Cú pháp

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>  
ORDER BY <danh sách các cột>
```

- ASC: tăng (mặc định)
- DESC: giảm

# Mệnh đề ORDER BY (tt)



## ■ Ví dụ

```
SELECT *  
FROM THAMGIADT  
ORDER BY MAGV DESC, MADT, STT DESC
```

MAGV	MADT	STT
GV01	DT01	1
GV01	DT01	2
GV01	DT02	1
GV02	DT01	2
GV02	DT01	3
GV02	DT03	1
GV02	DT03	4



MAGV	MADT	STT
GV02	DT01	3
GV02	DT01	2
GV02	DT03	4
GV02	DT03	1
GV01	DT01	2
GV01	DT01	1
GV01	DT02	1

- Giới thiệu
- Định nghĩa dữ liệu
- **Truy vấn dữ liệu**
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - Hàm kết hợp và gom nhóm
  - Một số dạng truy vấn khác
- Cập nhật dữ liệu
- Khung nhìn (view)
- **Chỉ mục (index)**

- SQL có cài đặt các phép toán
  - Hội (UNION)
  - Giao (INTERSECT)
  - Trừ (EXCEPT)
  
- Kết quả trả về là tập hợp
  - Loại bỏ các bộ trùng nhau
  - Để giữ lại các bộ trùng nhau
    - UNION ALL
    - INTERSECT ALL
    - EXCEPT ALL

# Phép toán tập hợp trong SQL (tt)



## ■ Cú pháp

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

**UNION [ALL]**

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

**INTERSECT [ALL]**

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

**EXCEPT [ALL]**

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```



- Cho biết mã của các giáo viên có họ tên bắt đầu là 'Nguyễn' và lương trên 200000 **hoặc**, giáo viên là trưởng bộ môn nhận chức sau năm 1995

```
SELECT MAGV
FROM GIAOVIEN
WHERE HOTEN LIKE N'Nguyễn%'
AND LUONG > 200000
UNION
SELECT TRUONGBM
FROM BOMON
WHERE YEAR(NGAYNHANCHUC) >= 1995
```

- Tìm những giáo viên vừa là trưởng bộ môn vừa chủ nhiệm đề tài

```
SELECT TRUONGBM  
FROM BOMON  
INTERSECT  
SELECT GVCNDT  
FROM DETAI
```

```
SELECT BM.TRUONGBM  
FROM BOMON BM, DETAI DT  
WHERE BM.TRUONGBM = DT.GVCNDT
```

- Tìm những giáo viên không tham gia bất kỳ đề tài nào

```
SELECT MAGV  
FROM GIAOVIEN  
EXCEPT  
SELECT MAGV  
FROM THAMGIADT
```

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN, BOMON  
WHERE TENBM = N'Hệ thống thông tin'  
AND GIAOVIEN.MABM = BOMON.MABM
```

Câu truy vấn  
cha (Outer  
query)

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <so sánh tập hợp> (
```

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>)
```

Câu truy vấn  
con (Subquery)

- Các câu lệnh SELECT có thể lồng nhau ở nhiều mức
- Câu truy vấn con thường trả về một tập các giá trị
- Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic
- Mệnh đề WHERE của câu truy vấn cha
  - <biểu thức> <so sánh tập hợp> <truy vấn con>
  - So sánh tập hợp thường đi cùng với một số toán tử
    - IN, NOT IN
    - ALL
    - ANY hoặc SOME
  - Kiểm tra sự tồn tại
    - EXISTS
    - NOT EXISTS

## ■ Có 2 loại truy vấn lồng

### - Lồng phân cấp

- Mệnh đề WHERE của truy vấn con không tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
- Khi thực hiện, câu truy vấn con sẽ được thực hiện trước

### - Lồng tương quan

- Mệnh đề WHERE của truy vấn con tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha
- Khi thực hiện, câu truy vấn con sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của truy vấn cha

- Tìm những giáo viên là trưởng bộ môn

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN, BOMON
```

```
WHERE GIAOVIEN.MAGV = BOMON.TRUONGBM
```

*Sử dụng phép kết*

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN
```

```
WHERE MAGV IN (SELECT TRUONGBM  
FROM BOMON)
```

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN  
WHERE MABM IN ('001',  
'002', '004', '005', '007')
```

*Sử dụng truy vấn lồng với IN*

Mệnh đề WHERE của truy vấn con **không** tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha

## Ví dụ 9



```
SELECT HOTEN
FROM GIAOVIEN
WHERE MAGV IN (SELECT MAGV
                FROM GIAOVIEN
                WHERE HOTEN LIKE N'Nguyễn%'
                AND LUONG > 200000)
OR MAGV IN (SELECT TRUONGBM
            FROM BOMON
            WHERE YEAR(NGAYNHANCHUC)>=1995)
```



## Ví dụ 10



- Tìm những giáo viên không tham gia đề tài nào

```
SELECT *  
FROM GIAOVIEN  
WHERE MAGV NOT IN(SELECT MAGV  
                   FROM THAMGIADT)
```

```
SELECT *  
FROM GIAOVIEN  
WHERE MAGV <> ALL(SELECT MAGV  
                  FROM THAMGIADT)
```

## Ví dụ 11



- Tìm những giáo viên có lương lớn hơn lương của ít nhất một giáo viên bộ môn ‘**Công nghệ phần mềm**’

```
SELECT *  
FROM GIAOVIEN  
WHERE LUONG > ANY (SELECT GV.LUONG  
                    FROM GIAOVIEN GV, BOMON BM  
                    WHERE GV.MABM = BM.MABM  
                    AND BM.TENBM = N'Công nghệ phần mềm')
```

```
SELECT GV1.*  
FROM GIAOVIEN GV1, GIAOVIEN GV2, BOMON BM  
WHERE GV2.MABM = BM.MABM  
AND BM.TENBM = N'Công nghệ phần mềm' AND GV1.LUONG > GV2.LUONG
```

## Ví dụ 12



- Tìm những giáo viên có lương lớn hơn lương của tất cả giáo viên thuộc bộ môn ‘**Hệ thống thông tin**’

```
SELECT *  
FROM GIAOVIEN  
WHERE LUONG > ALL (SELECT LUONG  
                    FROM GIAOVIEN GV, BOMON BM  
                    WHERE GV.MABM = BM.MABM  
                    AND BM.TENBM = N'Hệ thống thông tin')
```

- Tìm những trưởng bộ môn tham gia tối thiểu 1 đề tài

```
SELECT *  
FROM GIAOVIEN  
WHERE MAGV IN (SELECT TRUONGBM  
                FROM BOMON)  
AND MAGV IN (SELECT MAGV  
              FROM THAMGIADT)
```

# Ví dụ - Lồng tương quan



- Tìm những giáo viên là trưởng bộ môn

```
SELECT MAGV, HOTEN
```

```
FROM GIAOVIEN
```

*Sử dụng phép truy vấn lồng với IN*

```
WHERE MAGV IN (SELECT TRUONGBM FROM BOMON)
```

```
SELECT MAGV, HOTEN
```

```
FROM GIAOVIEN GV
```

```
WHERE EXISTS (SELECT *
```

```
FROM BOMON BM
```

*Sử dụng truy vấn lồng với EXISTS*

```
WHERE BM.TRUONGBM = GV.MAGV)
```

Giáo viên là trưởng bộ môn khi **tồn tại** một bộ môn có TRUONGBM = MAGV của giáo viên đó

Mệnh đề WHERE của truy vấn con tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha

# Ví dụ - Lồng tương quan



- Tìm những giáo viên có lương lớn nhất

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN GV  
WHERE NOT EXISTS (SELECT *  
                   FROM GIAOVIEN GV2  
                   WHERE GV2.LUONG > GV.LUONG)
```

Giáo viên là có lương lớn nhất khi **không tồn tại** một giáo viên nào mà có lương lớn hơn giáo viên đó

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN GV  
WHERE LUONG >= ALL (SELECT LUONG  
                    FROM GIAOVIEN GV2)
```

Giáo viên là có lương lớn nhất khi lương của giáo viên **lớn hơn hoặc bằng tất cả** lương của các giáo viên (lồng phân cấp)

## Ví dụ 14



- Tìm giáo viên trùng tên và cùng giới tính với giáo viên khác trong cùng bộ môn

```
SELECT *  
FROM GIAOVIEN GV1  
WHERE EXISTS (SELECT *  
              FROM GIAOVIEN GV2  
              WHERE GV1.HOTEN LIKE GV2.HOTEN  
              AND GV1.PHAI = GV2.PHAI  
              AND GV1.MABM = GV2.MABM  
              AND GV1.MAGV <> GV2.MAGV)
```

## Ví dụ 15



- Tìm những giáo viên không tham gia đề tài nào

```
SELECT *  
FROM GIAOVIEN GV  
WHERE NOT EXISTS (SELECT *  
                   FROM THAMGIADT PC  
                   WHERE PC.MAGV = GV.MAGV)
```

Giáo viên **GV** không tham gia đề tài khi **không tồn tại** một dòng nào trong THAMGIADT mà có  $MAGV = GV.MAGV$



## Ví dụ 16



- Tìm những giáo viên có lương lớn hơn lương của ít nhất một giáo viên bộ môn 'Công nghệ phần mềm'

```
SELECT *  
FROM GIAOVIEN GV1  
WHERE EXISTS (SELECT *  
               FROM GIAOVIEN GV2, BOMON BM  
               WHERE GV2.MABM = BM.MABM  
               AND BM.TENBM = N'Công nghệ phần  
mềm'  
               AND GV1.LUONG > GV2.LUONG)
```

## Ví dụ 17



- Tìm những trưởng bộ môn tham gia tối thiểu 1 đề tài

```
SELECT *  
FROM GIAOVIEN GV  
WHERE EXISTS (SELECT *  
               FROM BOMON BM  
               WHERE GV.MAGV = BM.TRUONGBM)  
AND EXISTS (SELECT * FROM THAMGIADT PC  
            WHERE PC.MAGV = GV.MAGV)
```

## ■ IN

- <tên cột> IN <câu truy vấn con>
- Thuộc tính ở mệnh đề SELECT của truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của truy vấn cha

## ■ EXISTS

- Không cần có thuộc tính, hằng số hay biểu thức nào khác đứng trước
- Không nhất thiết liệt kê tên thuộc tính ở mệnh đề SELECT của truy vấn con
- Những câu truy vấn có = ANY hay IN đều có thể chuyển thành câu truy vấn có EXISTS

## ■ EXISTS:

- Sử dụng để kiểm tra tồn tại
- Cú pháp sử dụng
- ... EXISTS (Select ... From ... Where...)

*Sub query*

Nếu sub query có dữ liệu ( $\geq 1$  dòng)  
→ tồn tại  $\leftrightarrow$  mệnh đề EXISTS (...) = TRUE

Nếu sub query không có dữ liệu (0 dòng hoặc rỗng)  
→ Không tồn tại  $\leftrightarrow$  mệnh đề EXISTS (...) = FALSE

## Ví dụ 18



- Tìm những giáo viên có tham gia đề tài

```
SELECT *  
FROM GIAOVIEN GV  
WHERE EXISTS (SELECT *  
                FROM THAMGIADT PC  
                WHERE PC.MAGV = GV.MAGV)
```

Nếu giáo viên **GV** có tham gia đề tài →  
câu truy vấn bên trong sẽ có dữ liệu  
( $\geq 1$  dòng) → mệnh EXISTS (S .. F...  
W) có giá trị **TRUE**

## Ví dụ 19



- Tìm những giáo viên không tham gia đề tài

```
SELECT *  
FROM GIAOVIEN GV  
WHERE NOT EXISTS (SELECT *  
                    FROM THAMGIADT PC  
                    WHERE PC.MAGV = GV.MAGV)
```

Nếu giáo viên **GV** không tham gia đề tài → câu truy vấn bên trong sẽ rỗng (0 dòng) → NOT EXISTS (S...F..W) có giá trị TRUE

# Phép chia trong SQL



R	A	B	C	D	E
	$\alpha$	a	$\alpha$	a	1
	$\alpha$	a	$\gamma$	a	1
	$\alpha$	a	$\gamma$	b	1
	$\beta$	a	$\gamma$	a	1
	$\beta$	a	$\gamma$	b	3
	$\gamma$	a	$\gamma$	a	1
	$\gamma$	a	$\gamma$	b	1
	$\gamma$	a	$\beta$	b	1

S	D	E
$b_i$	a	1
	b	1

$R \div S$	A	B	C
$a_i$	$\alpha$	a	$\gamma$
	$\gamma$	a	$\gamma$

- $R \div S$  là tập các giá trị  $a_i$  trong R sao cho không có giá trị  $b_i$  nào trong S làm cho bộ  $(a_i, b_i)$  không tồn tại trong R

- Sử dụng EXCEPT để biểu diễn

```
SELECT R1.A, R1.B, R1.C
FROM R R1
WHERE NOT EXISTS (
    ( SELECT S.D, S.E FROM S)
    EXCEPT
    ( SELECT R2.D, R2.E
      FROM R R2
      WHERE R1.A=R2.A AND R1.B=R2.B
        AND R1.C=R2.C )
)
```



- Sử dụng NOT EXISTS để biểu diễn

```
SELECT R1.A, R1.B, R1.C
FROM R R1
WHERE NOT EXISTS (
    SELECT *
    FROM S
    WHERE NOT EXISTS (
        SELECT *
        FROM R R2
        WHERE R2.D=S.D AND R2.E=S.E
        AND R1.A=R2.A AND R1.B=R2.B AND R1.C=R2.C ))
```

- Tìm tên các giáo viên được phân công làm tất cả các đề tài
  - Tìm tên các **nhân viên** mà không có **đề án** nào là không được **phân công làm**
  - Tập bị chia: THAMGIADT(MAGV, MADT)
  - Tập chia: DETAI(MADT)
  - Tập kết quả: KQ(MAGV)
  - Kết KQ với GIAOVIEN để lấy ra TENGV

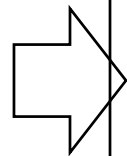
## Ví dụ 20 – Except



```
SELECT DISTINCT GV.MAGV, GV.HOTEN  
FROM GIAOVIEN GV, THAMGIADT PC1  
WHERE GV.MAGV = PC1.MAGV
```

AND NOT EXISTS

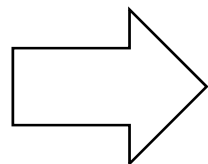
Những đề tài  
mà giáo viên  
PC1.MAGV  
không tham  
gia



```
( (SELECT DT.MADT FROM DETAI DT)  
EXCEPT  
(SELECT PC2.MADT  
FROM THAMGIADT PC2  
WHERE PC2.MAGV = PC1.MAGV) )
```

*Toàn bộ Ác tài*

*Những Ác tài mà  
PC1.MAGV Đã tham gia*



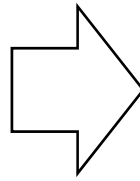
Nếu danh sách đề tài chưa tham gia =  
rỗng → Giáo viên đã tham gia tất cả các đề  
tài

## Ví dụ 20 – Not Exists



```
SELECT DISTINCT GV.MAGV, GV.HOTEN  
FROM GIAOVIEN GV, THAMGIADT PC1  
WHERE GV.MAGV = PC1.MAGV  
AND NOT EXISTS (
```

Những đề tài  
mà giáo viên  
PC1.MAGV  
không tham gia



```
SELECT *  
FROM DETAI DT  
WHERE NOT EXISTS (SELECT *  
FROM THAMGIADT PC2  
WHERE PC2.MAGV = PC1.MAGV  
AND DT.MADT = PC2.MADT))
```

Tìm những giáo viên mà danh sách đề tài không tham gia = rỗng

## Ví dụ 20 – Not Exists



```
SELECT DISTINCT GV.MAGV, GV.HOTEN
FROM GIAOVIEN GV, THAMGIADT PC1
WHERE GV.MAGV = PC1.MAGV
AND 0 = (SELECT COUNT(*)
        FROM DETAI DT
        WHERE NOT EXISTS (SELECT *
                          FROM THAMGIADT PC2
                          WHERE PC2.MAGV = PC1.MAGV
                          AND DT.MADT = PC2.MADT))
```

Tìm những giáo viên mà số lượng đề tài chưa tham gia bằng 0

- Cho  $R(\underline{A}, \underline{B})$ ,  $S(\underline{B})$ , thực hiện  $R \div S$

```
SELECT R.A
FROM R
[WHERE R.B IN (SELECT S.B FROM S [WHERE <ĐK>])]
GROUP BY R.A
HAVING COUNT(DISTINCT R.B) = ( SELECT COUNT(S.B)
                               FROM S
                               [WHERE <ĐK>])
```

## Ví dụ 20 – Count



```
SELECT DISTINCT GV.MAGV, GV.HOTEN  
FROM GIAOVIEN GV, THAMGIADT PC1  
WHERE GV.MAGV = PC1.MAGV  
GROUP BY GV.MAGV, GV.HOTEN  
HAVING COUNT(DISTINCT PC1.MADT) = (SELECT COUNT (MADT)  
                                   FROM DETAI DT)
```

## Ví dụ 21 – Count



- Tìm tên các giáo viên được phân công làm tất cả các đề tài có kinh phí trên 100 triệu?

```
SELECT DISTINCT GV.MAGV, GV.HOTEN
FROM GIAOVIEN GV, THAMGIADT PC1
WHERE GV.MAGV = PC1.MAGV
AND PC1.MADT IN (SELECT DT.MADT FROM DETAI WHERE KINHPhi > 100)
GROUP BY GV.MAGV, GV.HOTEN
HAVING COUNT(DISTINCT PC1.MADT) = (SELECT COUNT (MADT)
FROM DETAI DT
WHERE KINHPhi >100)
```



1. Cho biết mã số, họ tên, ngày sinh của giáo viên tham gia tất cả các công việc của đề tài ‘Ứng dụng hóa học xanh’.
2. Cho biết mã số, họ tên, tên bộ môn và tên người quản lý chuyên môn của giáo viên tham gia tất cả các đề tài thuộc chủ đề ‘Nghiên cứu phát triển’.
3. Cho biết họ tên, ngày sinh, tên khoa, tên trưởng khoa của giáo viên tham gia tất cả các đề tài có giáo viên ‘Nguyễn Hoài An’ tham gia.
4. Cho biết họ tên giáo viên khoa ‘Công nghệ thông tin’ tham gia tất cả các công việc của đề tài có trưởng bộ môn của bộ môn đồng nhất khoa ‘Công nghệ thông tin’ làm chủ nhiệm.

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - **Hàm kết hợp và gom nhóm**
  - Một số dạng truy vấn khác
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

## ■ COUNT

- COUNT(\*) đếm số dòng
- COUNT(<tên thuộc tính>) đếm số giá trị khác NULL của thuộc tính
- COUNT(DISTINCT <tên thuộc tính>) đếm số giá trị khác nhau và khác NULL của thuộc tính

## ■ MIN

## ■ MAX

## ■ SUM

## ■ AVG

## ■ Các hàm kết hợp được đặt ở mệnh đề SELECT hoặc HAVING

## Ví dụ 22



- Tìm tổng lương, lương cao nhất, lương thấp nhất và lương trung bình của các giáo viên

```
SELECT SUM(LUONG), MAX(LUONG), MIN(LUONG), AVG(LUONG)  
FROM GIAOVIEN
```

## Ví dụ 23



- Cho biết số lượng giáo viên của bộ môn ‘Mạng máy tính’

```
SELECT COUNT(*) AS SL_GV  
FROM GIAOVIEN GV, BOMON BM  
WHERE GV.MABM = BM.MABM  
AND TENBM=N'Mạng máy tính'
```

## Ví dụ 24



- Tìm những giáo viên có lương thuộc 3 mức lương cao nhất

```
SELECT *  
FROM GIAOVIEN GV1  
WHERE 2 >= (SELECT COUNT(*)  
            FROM GIAOVIEN GV2  
            WHERE GV2.LUONG > GV1.LUONG)
```

## Ví dụ 25



- Cho biết số lượng giáo viên của từng bộ môn

Bộ môn	Số lượng
HTTT	2
CNPM	1
MMT	1

MANV	HOTEN	...	MABM
GV001	Nguyễn Văn A	...	HTTT
GV002	Trần Văn B	...	HTTT
GV003	Trần Thị C	...	CNPM
GV004	Đặng Thị D	...	MMT

## ■ Cú pháp

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>  
GROUP BY <danh sách các cột gom nhóm>
```

## ■ Sau khi gom nhóm

- Mỗi nhóm các bộ sẽ có cùng giá trị tại các thuộc tính gom nhóm



- Cho biết số lượng giáo viên của từng bộ môn

```
SELECT MABM, COUNT(*) 'Số lượng giáo viên'  
FROM GIAOVIEN  
GROUP BY MABM
```

```
SELECT GV.MABM, COUNT(*) 'Số lượng giáo viên'  
FROM GIAOVIEN GV, BOMON BM  
WHERE GV.MABM = BM.MABM  
GROUP BY GV.MABM
```

## Ví dụ 27



- Với mỗi giáo viên cho biết mã số, mã đề tài và số công việc mà họ tham gia ứng với mỗi đề tài

MAGV	MADT	STT
GV001	DT001	1
GV001	DT001	2
GV001	DT002	1
GV002	DT002	2
GV003	DT001	3
GV003	DT002	3

```
SELECT PC.MAGV, PC.MADT, COUNT(*) AS 'Số lượng công việc'  
FROM THAMGIADT PC  
GROUP BY PC.MAGV, PC.MADT
```

## Ví dụ 28



- Cho biết những giáo viên tham gia từ 2 công việc trở lên cho mỗi đề tài?

MAGV	MADT	STT
GV001	DT001	1
GV001	DT001	2
GV001	DT002	1
GV002	DT002	2
GV003	DT001	3
GV003	DT002	3

## ■ Cú pháp

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <điều kiện>  
GROUP BY <danh sách các cột gom nhóm>  
HAVING <điều kiện trên nhóm>
```

## Ví dụ 29



- Cho biết những giáo viên tham gia từ 2 công việc trở lên cho mỗi đề tài?

```
SELECT PC.MAGV, PC.MADT, COUNT(*) AS 'Số lượng công việc'  
FROM THAMGIADT PC  
GROUP BY PC.MAGV, PC.MADT  
HAVING COUNT(*) >= 2
```

## Ví dụ 30



- Cho biết những giáo viên tham gia từ 2 đề tài trở lên

MAGV	MADT	STT
GV001	DT001	1
GV001	DT001	2
GV001	DT002	1
GV002	DT002	2
GV003	DT001	3
GV003	DT002	3

```
SELECT PC.MAGV, COUNT(DISTINCT MADT) AS 'Số lượng đề tài'  
FROM THAMGIADT PC  
GROUP BY PC.MAGV  
HAVING COUNT(DISTINCT MADT) >= 2
```

## Ví dụ 31



- Cho biết những bộ môn (TENBM) có lương trung bình của các giáo viên lớn hơn 20000

```
SELECT GV.MABM, AVG(GV.LUONG) AS 'Lương trung  
bình'
```

```
FROM GIAOVIEN GV
```

```
GROUP BY GV.MABM
```

```
HAVING AVG(GV.LUONG)>20000
```

```
SELECT BM.TENBM, AVG(GV.LUONG) AS 'Lương trung bình'
```

```
FROM GIAOVIEN GV, BOMON BM
```

```
WHERE GV.MABM = BM.MABM
```

```
GROUP BY BM.MABM, BM.TENBM
```

```
HAVING AVG(GV.LUONG)>=20000
```

## ■ Mệnh đề GROUP BY

- Các thuộc tính trong mệnh đề SELECT (trừ những thuộc tính trong các hàm kết hợp) phải xuất hiện trong mệnh đề GROUP BY

## ■ Mệnh đề HAVING

- Sử dụng các hàm kết hợp trong mệnh đề SELECT để kiểm tra một số điều kiện nào đó
- Chỉ kiểm tra điều kiện trên nhóm, không là điều kiện lọc trên từng bộ
- Sau khi gom nhóm điều kiện trên nhóm mới được thực hiện



- Thứ tự thực hiện câu truy vấn có mệnh đề GROUP BY và HAVING
  - (1) Chọn ra những dòng thỏa điều kiện trong mệnh đề WHERE
  - (2) Những dòng này sẽ được gom thành nhiều nhóm tương ứng với mệnh đề GROUP BY
  - (3) Áp dụng các hàm kết hợp cho mỗi nhóm
  - (4) Bỏ qua những nhóm không thỏa điều kiện trong mệnh đề HAVING
  - (5) Rút trích các giá trị của các cột và hàm kết hợp trong mệnh đề SELECT

## Ví dụ 32



- Tìm những phòng ban có lương trung bình cao nhất

```
SELECT GV.MABM, AVG(GV.LUONG) AS 'Lương trung bình'
FROM GIAOVIEN GV
GROUP BY GV.MABM
HAVING AVG(GV.LUONG)) = (SELECT MAX(AVG(GV.LUONG))
FROM GIAOVIEN GV
GROUP BY GV.MABM)
```

- Tìm những phòng ban có lương trung bình cao nhất

```
SELECT GV.MABM, AVG(GV.LUONG) AS 'Lương trung bình'
FROM GIAOVIEN GV
GROUP BY GV.MABM
HAVING AVG(GV.LUONG) >= ALL(SELECT AVG(GV.LUONG)
                             FROM GIAOVIEN GV
                             GROUP BY GV.MABM)
```

## Ví dụ 34



- Tìm tên các giáo viên được phân công làm tất cả các đề tài

```
SELECT PC.MAGV, COUNT(DISTINCT PC.MADT) AS 'Số lượng đề tài'  
FROM THAMGIADT PC  
GROUP BY PC.MAGV  
HAVING COUNT(DISTINCT PC.MADT) = (SELECT COUNT(MADT)  
                                  FROM DETAI)
```

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
  - Truy vấn cơ bản
  - Tập hợp, so sánh tập hợp và truy vấn lồng
  - Hàm kết hợp và gom nhóm
  - **Một số dạng truy vấn khác**
- Cập nhật dữ liệu
- Khung nhìn (view)
- Chỉ mục (index)

# Một số dạng truy vấn khác



- Truy vấn con ở mệnh đề FROM
- Điều kiện kết ở mệnh đề FROM
  - Phép kết tự nhiên
  - Phép kết ngoài
- Cấu trúc CASE

# Truy vấn con ở mệnh đề FROM



- Kết quả trả về của một câu truy vấn phụ là một bảng
  - Bảng trung gian trong quá trình truy vấn
  - Không có lưu trữ thật sự
- Cú pháp

```
SELECT <danh sách các cột>  
FROM R1, R2, (<truy vấn con>) AS tên_bảng  
WHERE <điều kiện>
```

## Ví dụ 35



- Cho biết những bộ môn (TENBM) có lương trung bình của các giáo viên lớn hơn 20000

```
SELECT BM.TENBM, AVG(GV.LUONG) AS LUONG_TB  
FROM GIAOVIEN GV, BOMON BM  
WHERE GV.MABM = BM.MABM  
GROUP BY BM.MABM, BM.TENBM  
HAVING AVG(GV.LUONG) >= 20000
```



- Cho biết những bộ môn (TENBM) có lương trung bình của các giáo viên lớn hơn 20000

```
SELECT BM.TENBM, LUONG_GV.LUONG_TB
FROM BOMON BM, (SELECT MABM, AVG(LUONG) LUONG_TB
                FROM GIAOVIEN
                GROUP BY MABM) AS LUONG_GV
WHERE BM.MABM = LUONG_GV.MABM
```

# Điều kiện kết ở mệnh đề FROM



## ■ Kết bằng

```
SELECT <danh sách các cột>  
FROM R1 [INNER] JOIN R2 ON <biểu thức>  
WHERE <điều kiện>
```

## ■ Kết ngoài

```
SELECT <danh sách các cột>  
FROM R1 LEFT|RIGHT [OUTER] JOIN R2 ON <biểu thức>  
WHERE <điều kiện>
```

## Ví dụ 36



- Tìm mã và tên các giáo viên làm việc tại bộ môn ‘Hệ thống thông tin’

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN, BOMON  
WHERE TENBM = N'Hệ thống thông tin'  
AND GIAOVIEN.MABM = BOMON.MABM
```

```
SELECT MAGV, HOTEN  
FROM GIAOVIEN GV INNER JOIN BOMON BM ON GV.MABM = BM.MABM  
WHERE TENBM = N'Hệ thống thông tin'
```

## Ví dụ 37



- Tìm họ tên các giáo viên và tên các đề tài giáo viên tham gia nếu có

```
SELECT DISTINCT GV.*  
FROM GIAOVIEN GV LEFT JOIN (THAMGIADT PC JOIN DETAI DT ON  
PC.MADT = DT.MADT) ON GV.MAGV = PC.MAGV
```

GIAOVIEN

GIAOVIEN JOIN THAMGIADT

GV.MAGV = PC.MAGV

mở rộng



- Cho phép kiểm tra điều kiện và xuất thông tin theo từng trường hợp
- Cú pháp

```
CASE <tên cột>  
    WHEN <giá trị> THEN <biểu thức>  
    WHEN <giá trị> THEN <biểu thức>  
    ...  
    [ELSE <biểu thức>]  
END
```

## Ví dụ 38



- Cho biết họ tên các giáo viên đã đến tuổi về hưu (nam 60 tuổi, nữ 55 tuổi)

```
SELECT HOTEN
FROM GIAOVIEN
WHERE YEAR(GETDATE()) - YEAR(NGAYSINH) >= ( CASE PHAI
                                              WHEN 'Nam' THEN 60
                                              WHEN 'Nu' THEN 55
                                              END)
```

## Ví dụ 39



- Cho biết họ tên các giáo viên và năm về hưu

```
SELECT GV.HOTEN, YEAR(GV.NGAYSINH) + ( CASE PHAI
                                         WHEN 'Nam' THEN 60
                                         WHEN 'Nu' THEN 55
                                         END) AS NAMVEHUU
FROM GIAOVIEN GV
```

```
SELECT GV.HOTEN, ( CASE PHAI
                   WHEN 'Nam' THEN YEAR(NGAYSINH) + 60
                   WHEN 'Nu' THEN YEAR(NGAYSINH) + 55
                   END) AS NAMVEHUU
FROM GIAOVIEN GV
```

**SELECT** <danh sách các cột>  
**FROM** <danh sách các bảng>  
**[WHERE** <điều kiện>  
**[GROUP BY** <các thuộc tính gom nhóm>  
**[HAVING** <điều kiện trên nhóm>  
**[ORDER BY** <các thuộc tính sắp thứ tự>



- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
- **Cập nhật dữ liệu**
  - Thêm (insert)
  - Xóa (delete)
  - Sửa (update)
- Khung nhìn (view)
- Chỉ mục (index)

- Dùng để thêm 1 hay nhiều dòng vào bảng
  
- Để thêm dữ liệu
  - Tên quan hệ
  - Danh sách các thuộc tính cần thêm dữ liệu
  - Danh sách các giá trị tương ứng

# Lệnh INSERT (tt)



- Cú pháp (thêm 1 dòng)

```
INSERT INTO <tên bảng>(<danh sách các thuộc tính>)  
VALUES (<danh sách các giá trị>)
```

## Ví dụ 40



```
INSERT INTO THAMGIADT(MAGV, MADT, STT, PHUCAP, KETQUA)  
VALUES('002', '001', 1, 1.0, N'ĐẠT')
```

```
INSERT INTO THAMGIADT(MAGV, MADT, STT, PHUCAP, KETQUA)  
VALUES('002', '001', 2, 1.2, NULL)
```

## ■ Nhận xét

- Thứ tự các giá trị phải trùng với thứ tự các cột
- Có thể thêm giá trị NULL ở những thuộc tính không là khóa chính và NOT NULL
- Câu lệnh INSERT sẽ gặp lỗi nếu vi phạm RBTV
  - Khóa chính
  - Tham chiếu
  - NOT NULL - các thuộc tính có ràng buộc NOT NULL bắt buộc phải có giá trị

- Cú pháp (thêm nhiều dòng)

```
INSERT INTO <tên bảng>(<danh sách các thuộc tính>)  
          <câu truy vấn con>
```

## Ví dụ 41



```
CREATE TABLE THONGKE_BM (  
    TENBM NVARCHAR(50),  
    SL_GV INT,  
    LUONG_TC INT  
    PRIMARY KEY(TENBM)  
)
```

```
INSERT INTO THONGKE_BM  
SELECT BM.TENBM, COUNT(GV.MAGV), SUM(GV.LUONG)  
FROM GIAOVIEN GV, BOMON BM  
WHERE GV.MABM = BM.MABM  
GROUP BY BM.MABM, BM.TENBM
```

# Lệnh DELETE



- Dùng để xóa các dòng của bảng
- Cú pháp

```
DELETE FROM <tên bảng>  
[WHERE <điều kiện>]
```



```
DELETE FROM GIAOVIEN  
WHERE HOTEN LIKE N'Trần%
```

```
DELETE FROM GIAOVIEN  
WHERE MAGV = 'GV001'
```

```
DELETE FROM GIAOVIEN
```

- Xóa đi những giáo viên ở bộ môn 'Hệ thống thông tin'

```
DELETE FROM GIAOVIEN
WHERE MABM IN (SELECT MABM
                FROM BOMON
                WHERE TENBM = N'Hệ thống thông tin')
```

```
DELETE FROM GIAOVIEN
FROM BOMON BM
WHERE GIAOVIEN.MABM = BM.MABM
AND BM.TENBM = N'Hệ thống thông tin'
```

## ■ Nhận xét

- Số lượng số dòng bị xóa phụ thuộc vào điều kiện ở mệnh đề WHERE
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị xóa
- Lệnh DELETE có thể gây ra vi phạm RB tham chiếu
  - Không cho xóa
  - Xóa luôn những dòng có giá trị đang tham chiếu đến
    - CASCADE
  - Đặt NULL cho những giá trị tham chiếu

# Lệnh DELETE (tt)



MAGV	HOTEN	...	MABM
GV001	Nguyễn Văn A	...	HTTT
GV002	Trần Văn B	...	HTTT
GV003	Trần Thị C	...	CNPM
GV004	Đặng Thị D	...	MMT

MAGV	MADT	STT	PHUCAP	KETQUA
GV001	001	1	...	...
GV001	001	3	...	...
GV003	002	1	...	
GV004	003	1	...	...
...	...	...	...	...

# Lệnh DELETE (tt)



MABM	TENBM
<del>HTTT</del>	<del>Hệ thống thông tin</del>
CNPM	Công nghệ phần mềm
MMT	Mạng máy tính
KHMT	Khoa học máy tính

MANV	HOTEN	...	MABM
GV001	Nguyễn Văn A	...	<b>NULL</b>
GV002	Trần Văn B	...	<b>NULL</b>
GV003	Trần Thị C	...	CNPM
GV004	Đặng Thị D	...	MMT

- Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng
- Cú pháp

```
UPDATE <tên bảng>  
SET <tên thuộc tính>=<giá trị mới>,  
      <tên thuộc tính>=<giá trị mới>,  
...  
[WHERE <điều kiện>]
```

```
UPDATE GIAOVIEN  
SET NGAYSINH='08/12/1965'  
WHERE MAGV='GV001'
```

```
UPDATE GIAOVIEN  
SET LUONG=LUONG*1.1
```

## Ví dụ 45



- Với mỗi giáo viên của bộ môn ‘Hệ thống thông tin’ , nâng lương của các giáo viên gấp 1.5 lần và gán giáo viên quản lý (GVQL) thành null

```
UPDATE GIAOVIEN
SET LUONG = LUONG*1.5, GVQL = NULL
WHERE MABM = (SELECT MABM
              FROM BOMON
              WHERE TENBM = N'Hệ thống thông tin')
```



## Ví dụ 46



- Tăng 10% lương cho giáo viên có tham gia đề tài

```
UPDATE GIAOVIEN  
SET LUONG = LUONG * 1.1  
FROM THAMGIADT TG  
WHERE TG.MAGV = GIAOVIEN.MAGV
```

```
UPDATE GIAOVIEN  
SET LUONG = LUONG * 1.1  
WHERE EXISTS (  
SELECT *  
FROM THAMGIADT TG  
WHERE TG.MAGV = GIAOVIEN.MAGV )
```

```
UPDATE GIAOVIEN  
SET LUONG = LUONG * 1.1  
WHERE MAGV IN (  
SELECT TG.MAGV  
FROM THAMGIADT TG  
WHERE TG.MAGV = GIAOVIEN.MAGV )
```

## Ví dụ 47



- Tăng lương 10% cho tất cả các giáo viên làm cho đề tài 'DT001' nhiều hơn 3 công việc.

```
UPDATE GIAOVIEN
SET LUONG = LUONG * 1.1
WHERE MAGV IN (SELECT PC.MAGV
               FROM THAMGIADT TG
               WHERE TG.MADT = 'DT001'
               GROUP BY TG.MAGV, TG.MADT
               HAVING COUNT(*) >= 3)
```

## ■ Nhận xét

- Những dòng thỏa điều kiện tại mệnh đề WHERE sẽ được cập nhật giá trị mới
- Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị cập nhật
- Lệnh UPDATE có thể gây ra vi phạm RB tham chiếu
  - Không cho sửa
  - Sửa luôn những dòng có giá trị đang tham chiếu đến
    - CASCADE

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- **Khung nhìn**
  - Định nghĩa
  - Truy vấn
  - Cập nhật
- Chỉ mục

- Bảng là một quan hệ được tổ chức lưu trữ vật lý trong CSDL
- Khung nhìn cũng là một quan hệ
  - Không được lưu trữ vật lý (bảng ảo)
  - Không chứa dữ liệu
  - Được định nghĩa từ những bảng khác
  - Có thể truy vấn hay cập nhật thông qua khung nhìn

- Tại sao phải sử dụng khung nhìn?
  - Che dấu tính phức tạp của dữ liệu
  - Đơn giản hóa các câu truy vấn
  - Hiển thị dữ liệu dưới dạng tiện dụng nhất
  - An toàn dữ liệu

## ■ Cú pháp

```
CREATE VIEW <tên khung nhìn> AS  
    <câu truy vấn>
```

```
DROP VIEW <tên khung nhìn>
```

## ■ Bảng ảo này có

- Danh sách thuộc tính trùng với các thuộc tính trong mệnh đề **SELECT**
- Số dòng phụ thuộc vào điều kiện ở mệnh đề **WHERE**
- Dữ liệu được lấy từ các bảng ở mệnh đề **FROM**

```
CREATE VIEW GV_HTTT AS  
    SELECT GV.*  
    FROM GIAOVIEN GV  
    WHERE BM.MABM = 'HTTT'
```

```
CREATE VIEW THONGKE_BM AS  
    SELECT BM.TENBM, COUNT(GV.MAGV) SL_GV,  
           SUM(GV.LUONG) TONG_LUONG  
    FROM GIAOVIEN GV, BOMON BM  
    WHERE GV.MABM = BM.MABM  
    GROUP BY BM.MABM, BM.TENBM
```



- Tuy không chứa dữ liệu nhưng có thể thực hiện các câu truy vấn trên khung nhìn

```
SELECT GV.HOTEN  
FROM GV_HTTT GV  
WHERE GV.MAGV = 'GV003'
```

$$GV\_HTTT \leftarrow \sigma_{MABM='HTTT'}(GIAOVIEN)$$
$$\pi_{HOTEN}(\sigma_{MAGV='GV003'}(GV\_HTTT))$$

- Có thể viết câu truy vấn dữ liệu từ khung nhìn và bảng

```
SELECT DISTINCT GV.*  
FROM GV_HTTT GV, THAMGIADT PC  
WHERE GV.MAGV = PC.MAGV
```

$GV\_HTTT \leftarrow \sigma_{MABM='HTTT'}(GIAOVIEN)$

$KQ \leftarrow GV\_HTTT \bowtie_{GV\_HTTT.MAGV=THAMGIADT.MAGV} THAMGIADT$

- Có thể dùng các câu lệnh INSERT, DELETE và UPDATE cho các khung nhìn đơn giản
  - Khung nhìn được xây dựng trên 1 bảng và có khóa chính của bảng
  
- Không thể cập nhật dữ liệu nếu
  - Khung nhìn có dùng từ khóa DISTINCT
  - Khung nhìn có sử dụng các hàm kết hợp
  - Khung nhìn có mệnh đề SELECT mở rộng
  - Khung nhìn được xây dựng từ bảng có RB trên cột
  - Khung nhìn được xây dựng từ nhiều bảng

- Sửa lại lương cho giáo viên mã 'GV003' ở bộ môn 'Hệ thống thông tin' tăng lên 10%

```
UPDATE GV_HTTT  
SET LUONG = LUONG * 1.1  
WHERE MAGV = 'GV003'
```

- Giới thiệu
- Định nghĩa dữ liệu
- Truy vấn dữ liệu
- Cập nhật dữ liệu
- Khung nhìn
- **Chỉ mục**

- Chỉ mục trên thuộc tính A là một cấu trúc dữ liệu làm cho việc tìm kiếm mẫu tin có chứa A hiệu quả hơn

```
SELECT *  
FROM GIAOVIEN  
WHERE MABM='HTTT' AND PHAI= 'Nu'
```

Đọc 10.000 bộ

Đọc 200 bộ

Bảng GIAOVIEN có 10.000 bộ  
Có 200 giáo viên làm việc cho bộ môn 'HTTT'

Đọc 70 bộ

- Cú pháp

```
CREATE INDEX <tên chỉ mục> ON <tên bảng>(<tên cột>)
```

```
DROP INDEX <tên chỉ mục>
```

- Ví dụ

```
CREATE INDEX MABM_IND ON GIAOVIEN(MABM)
```

```
CREATE INDEX MABM_PHAI_IND ON GIAOVIEN(MABM, PHAI)
```

- Nhận xét
  - Tìm kiếm nhanh trong trường hợp so sánh với hằng số và phép kết
  - Làm chậm đi các thao tác thêm, xóa và sửa
  - Tốn chi phí
    - Lưu trữ chỉ mục
    - Truy xuất đĩa nhiều
- Chọn lựa cài đặt chỉ mục hợp lý?  $\Rightarrow$  sẽ được tìm hiểu kỹ trong các môn học tiếp Hệ QTCSDL, CSDL NC.



