

Bài 06:  
**NGÔN NGỮ SQL**



# Nội dung

---

1. Tổng quan
2. Định nghĩa dữ liệu
3. Thao tác cập nhật dữ liệu
4. Truy vấn dữ liệu
5. Khung nhìn

# 1. Tổng quan

---

SQL (Structured Query Language) là gì ?

- Là một công cụ quản lý dữ liệu được sử dụng phổ biến ở nhiều lĩnh vực.
- Hầu hết các ngôn ngữ bậc cao đều có trình hỗ trợ SQL
- Là ngôn ngữ chuẩn để truy vấn và thao tác trên CSDL quan hệ

# 1. Tổng quan

---

## Đặc điểm của SQL

- ✓ Là ngôn ngữ tựa Tiếng Anh
- ✓ Là ngôn ngữ phi thủ tục
- ✓ Người sử dụng chỉ cần đưa ra nội dung cần truy vấn
- ✓ Cung cấp tập lệnh phong phú cho các công việc hỏi đáp dữ liệu
- ✓ Yêu cầu duy nhất để sử dụng cho các hỏi đáp là phải nắm vững được các cấu trúc CSDL của mình.

# SQL và Đại số quan hệ

---



- ❖ Đại số quan hệ – Relational algebra
  - ✓ Cách thức truy vấn dữ liệu (Mô hình hóa, toán học)
  - ✓ Khó khăn cho sử dụng
- ❖ SQL – Structured Query Language
  - ✓ Ngôn ngữ lập trình cấp cao (CSDL)
  - ✓ Thuận lợi cho việc truy vấn thông tin thông qua câu lệnh có cấu trúc

# SQL và các mục tiêu chính



## ✓ Các mục tiêu chính của SQL bao gồm:

- Định nghĩa dữ liệu (DDL – Data Definition Language)
- Thao tác dữ liệu (DML – Data Manipulation language)
- Định nghĩa khung nhìn (View Definition)
- Ràng buộc toàn vẹn (Integrity Constraints)
- Phân quyền và bảo mật (Security and Privileges)
- Điều khiển giao tác (TCL – Transaction control language)

## ✓ Thuật ngữ

- Bảng (Table) → Quan hệ (Relation)
- Cột (Column, Field) → Thuộc tính (Attribute)
- Dòng (Row, Record) → Bộ (Tuple)

# Định nghĩa dữ liệu (DDL – Data Definition Language)

# Định nghĩa dữ liệu



- ❖ Bao gồm tập các lệnh phục vụ cho việc
  - ✓ Mô tả lược đồ cho mỗi quan hệ
  - ✓ Xác định miền giá trị tương ứng của từng thuộc tính
  - ✓ Tạo ra các ràng buộc toàn vẹn
  - ✓ Quy định chỉ mục trên mỗi quan hệ
- ❖ Các lệnh bao gồm
  - ✓ Tạo mới: **Create**[Database, Table, Column, ...]
  - ✓ Chỉnh sửa: **Alter**[Table, Column, Constraints...]
  - ✓ Loại bỏ: **Drop**[Database, Table, Column, ...]



# Kiểu dữ liệu

Kiểu dữ liệu	Loại	Bytes	Diễn giải
Bit	Integer	1	giá trị 0 hoặc 1
Bigint	Integer	8	$-2^{63}$ đến $2^{63} - 1$
Int	Integer	4	-2.147.483.648 đến 2.147.483.647
SmallInt	Integer	2	-32.768 đến 32.767
TinyInt	Integer	1	0 đến 255
Numeric	Numeric	Varies	$-10^{38} - 1$ đến $10^{38} - 1$
Money	Money	8	$-2^{63}$ đến $2^{63}$ thêm 4 số lẻ

# Kiểu dữ liệu

Kiểu dữ liệu	Loại	Bytes	Diễn giải
SmallMoney	Money	4	-214.748,3468 đến 214.748,3468
Float	Numeric	Varies	-1.79E+308 đến 1.79E+308
DateTime	Date/Time	8	từ 01/01/1900 đến 31/12/9999. Cho phép có giá trị Null
Small datetime	Date/Time	4	từ 01/01/1900 đến 31/12/2079. Cho phép có giá trị Null
Char	Character	Varies	Kiểu ký tự Non-Unicode với chiều dài cố định, cho phép 8000 ký tự
Varchar	Character	Varies	Kiểu ký tự Non-Unicode với chiều dài biến đổi, cho phép 8000 ký tự

# Kiểu dữ liệu

Kiểu dữ liệu	Loại	Bytes	Diễn giải
Nchar	Unicode	Varies	Kiểu dữ liệu Unicode với chiều dài cố định, chiều dài cho phép 4000 ký tự
Nvarchar	Unicode	Varies	Kiểu dữ liệu Unicode với chiều dài biến đổi, chiều dài cho phép 4000 ký tự
Text	Character	Varies	Kiểu dữ liệu ký tự Non-Unicode, cho phép 2.147.483.647 ký tự
Ntext	Unicode	Varies	Kiểu dữ liệu ký tự Non-Unicode, cho phép 1.073.741.823 ký tự

# Tạo – Hủy cơ sở dữ liệu

**CREATE DATABASE** <Tên\_cSDL>;

Vd: Tạo CSDL QLDA

Create database QLDA;

Tạo CSDL QLSV

Create database QLSV;

**DROP DATABASE** <Tên\_cSDL>;

Vd: Xóa CSDL QLDA

Drop database QLDA;

# Tạo bảng - Create table

## ✓ 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>],
    ...
    [<khai báo khóa chính, khóa ngoại, ràng buộc>]
)
```

# Tạo bảng - Create table

Cho lược đồ quan hệ NHANVIEN với các thuộc tính cần lưu trữ dữ liệu như yêu cầu sau:

✓MANV	Số nguyên
✓HONV	15 ký tự
✓TENDEM	20 ký tự
✓TENNV	10 ký tự
✓NGAYSINH	Ngày tháng
✓DIACHI	60 ký tự
✓GIOITINH	0 1
✓LUONG	Số nguyên
✓MANQL	Số nguyên
✓MAPB	Số nguyên

```
CREATE TABLE NHANVIEN (  
    MANV          INT          Not Null,  
    HONV          nVarChar(15)  Not Null,  
    TENDEM        nvarchar(20)  null,  
    TENNV         nVARCHAR(10)  NOT NULL,  
    NGAYSINH      DATETIME,  
    DIACHI        NVARCHAR(60),  
    GIOITINH      bit,  
    LUONG         int,  
    MANQL         INT,  
    MAPB          INT          NOT NULL  
);
```

# Tạo bảng - Create table

- ❖ **Null** Cho phép chứa giá trị null
- ❖ **Not null** Không được chứa giá trị null
- ❖ **Default <val>** Gán giá trị mặc định cho thuộc tính tương ứng
- ❖ **Primary key** Thuộc tính được chỉ định làm khóa chính  
(Không chấp nhận giá trị null)
- ❖ **Foreign key** Thuộc tính là khóa ngoại, tham chiếu (references) tới khóa chính của một table khác
- ❖ **Unique** Giá trị của thuộc tính là duy nhất, có thể chứa giá trị null
- ❖ **Check <dk>** Điều kiện kiểm tra dữ liệu đối với thuộc tính tương ứng

# Tạo bảng với RBTV



Cho lược đồ quan hệ **NHANVIEN** với các cột cần lưu trữ thông tin như sau:

⇒ <u>MANV</u>	Số nguyên	Khóa chính
⇒ HONV	15 ký tự	Bắt buộc
⇒ TENDEM	20 ký tự	Không bắt buộc
⇒ TENNV	10 ký tự	Bắt buộc
⇒ NGAYSINH	Ngày tháng	
⇒ DIACHI	60 ký tự	Không bắt buộc
⇒ GIOITINH	0   1	Mặc nhiên là 1
⇒ LUONG	Số nguyên	Phải lớn hơn 0
⇒ MANQL	Số nguyên	
⇒ MAPB	Số nguyên	Khóa ngoại [table: PHONGBAN(MAPB)]



# Tạo bảng với RBTV

```
CREATE TABLE NHANVIEN (  
    MANV          INT          Not Null,  
    HONV          nVarChar(15) Not Null,  
    TENDEM        nvarchar(20) null,  
    TENNV         nVARCHAR(10) NOT NULL,  
    NGAYSINH       DATETIME,  
    DIACHI        NVARCHAR(60),  
    GIOITINH       bit,  
    LUONG          int,  
    MANQL         INT,  
    MAPB          INT          NOT NULL  
);
```

Trong SQL Server, mỗi RBTV có thể được đặt tên bằng cách sử dụng:

CONSTRAINT <Tên ràng buộc> <Kiểu ràng buộc>.

Lưu ý: Tên của ràng buộc phải là duy nhất trong một lược đồ CSDL.

# Tạo bảng với RBTV

```
CREATE TABLE NHANVIEN (  
    MANV      INT      Constraint NV_MANV Primary Key Not Null,  
    HONV      NVARCHAR(10) NOT NULL,  
    TENDEM    NVARCHAR(20) NULL,  
    TENNV     NVARCHAR(10) NOT NULL,  
    NGAYSINH  DATETIME,  
    DIACHI    NVARCHAR(50),  
    GIOITINH  BIT      DEFAULT 1,  
    LUONG     INT      Constraint KT_LUONG Check (LUONG>0),  
    MANQL     INT,  
    MAPB      INT      Constraint NV_MAPB Foreign Key References PHONGBAN(MAPB)  
);
```

Lưu ý: thuộc tính Cascade đối với RBTV thuộc loại References:: On delete cascade, On update cascade  
SQL-Server:

Constraint NV\_MAPB Foreign Key References PHONGBAN(MAPB) ON DELETE CASCADE

# Thay đổi cấu trúc bảng

---



## ✓ Thay đổi cấu trúc bảng

- Thêm cột
- Xóa cột
- Mở rộng cột

## ✓ Thay đổi ràng buộc toàn vẹn (RBTV)

- Thêm RBTV
- Xóa RBTV

# Thêm cột

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

Ví dụ: Thêm cột NGHENGHIEP vào bảng NHANVIEN

```
ALTER TABLE NHANVIEN ADD NGHENGHIEP CHAR(20)
```

# Xóa cột

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

Ví dụ: Xóa cột NGHENGHIEP vào bảng NHANVIEN

```
ALTER TABLE NHANVIEN DROP COLUMN NGHENGHIEP
```

# Sửa kiểu dữ liệu của thuộc tính

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

Ví dụ: Đổi kiểu dữ liệu của NGHENGHIEP trong bảng NHANVIEN

```
ALTER TABLE NHANVIEN ALTER COLUMN NGHENGHIEP CHAR(50)
```

*Lưu ý: Không phải sửa bất kỳ kiểu dữ liệu nào cũng được.*

# Sửa bảng :: Alter table

**Chú ý:** Thêm ràng buộc toàn vẹn

Mục đích của sửa bảng

- Thay đổi cấu trúc bảng
- Thay đổi RBTV

**ALTER TABLE** <tên\_bảng>  
**ADD CONSTRAINT**  
<tên\_ràng\_buộc>

**UNIQUE** tên\_cột

**PRIMARY KEY** (tên\_cột)

**FOREIGN KEY** (tên\_cột)  
**REFERENCES** tên\_bảng  
(cột\_là\_khóa\_chính)

**CHECK** (tên\_cột điều\_kiện)

Ví dụ: Thêm các RBTV vào bảng PHONGBAN

**ALTER TABLE** PHONGBAN **ADD**

**CONSTRAINT** PB\_MAPHG\_PK **PRIMARY KEY** (MAPHG),

**CONSTRAINT** PB\_TRPHG **FOREIGN KEY** (TRPHG) **REFERENCES** NHANVIEN(MANV),

**CONSTRAINT** PB\_NGNHANCHUC\_DF **DEFAULT** (GETDATE()) **FOR** (NG\_NHANCHUC),

**CONSTRAINT** PB\_TENPB\_UNI **UNIQUE** (TENPB)

# Thêm ràng buộc toàn vẹn

**Lưu ý:** Không phải sửa bất kỳ kiểu dữ liệu nào cũng được

Ví dụ:

- ✓ Nếu sửa kiểu dữ liệu của cột Ghi\_chu thành varchar(50), mà trước đó đã nhập giá trị cho cột Ghi\_chu có độ dài hơn 50 ký tự thì không được phép.
- ✓ Hoặc sửa từ kiểu chuỗi ký tự sang kiểu số, ...



# Xóa ràng buộc toàn vẹn

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

Ví dụ: Xóa các RBTV vào bảng PHONGBAN

```
ALTER TABLE PHONGBAN DROP  
    CONSTRAINT PB_TENPB_UNI
```

# Xóa bảng

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

**DROP TABLE** <Tên\_bảng>

Ví dụ: Xóa bảng DEAN

**DROP TABLE** DEAN

**Lưu ý:** khi muốn xóa một bảng phải xóa tất cả những khóa ngoại tham chiếu tới bảng đó trước.

**NHANVIEN**

HONV	TENLOT	TENNV	<u>MANV</u>	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
------	--------	-------	-------------	--------	------	------	-------	--------	-----

**PHONGBAN**

TENPHG	<u>MAPHG</u>	TRPHG	NG_NHANCHUC
--------	--------------	-------	-------------

# Lệnh tạo miền giá trị

✓ 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 Kieu_Ten AS VARCHAR(30)
```

# Thao tác dữ liệu (DML – Data Manipulation Language)

# Thêm dữ liệu



- ✓ Mục đích: dùng để thêm 1 hay nhiều dòng vào bảng
- ✓ Nguyên tắc thêm dữ liệu
  - Tên quan hệ
  - Danh sách các thuộc tính cần thêm dữ liệu
  - Danh sách các giá trị tương ứng
- ✓ Cú pháp

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

# Thêm dữ liệu

✓ Hãy tiến hành nhập dữ liệu cho table NHANVIEN với các thông tin:

- Công ty mới tuyển một nhân viên nữ, phòng kế toán (mã phòng: 4), họ tên: Đỗ Thị Hải Yến, sinh ngày: 25/01/1986, mã số NV: 127, Lương: 5000000

```
INSERT INTO NHANVIEN(MANV, HONV, TENDEM, TENNV, NGAYSINH, GIOITINH, MAPB, LUONG)
VALUES (127, N'Đỗ', N'Thị Hải', N'Yến', '1986/01/25', 0, 4, 5000000);
```

- Công ty mới tuyển một nhân viên nam, phòng kỹ thuật (mã phòng: 3), họ tên: Trần Mạnh Hà, sinh ngày: 13/06/1992, Mã số: 129, Lương: 8000000

```
INSERT INTO NHANVIEN(MANV, HONV, TENDEM, TENNV, NGAYSINH, DCHI, MAPB, LUONG)
VALUES (129, N'Trần', N'Mạnh', N'Hà', '1992/06/13', NULL, 3, 8000000)
```

- Nhập thông tin cho chị “Nguyễn Đỗ Minh Loan”, sinh ngày 28/10/1980, địa chỉ: 41 Ngô Quyền, Q5, TP.HCM, mức lương 12 triệu, phòng 3, mã người Q. 00

```
INSERT INTO NHANVIEN VALUES (112, N'Nguyễn', N'Đỗ Minh', N'Loan', '1980/10/28',
N'41 Ngô Quyền, Q.5, TP.HCM', 0, 12000000, 98, 3)
```

# Thêm dữ liệu



- ✓ 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

# Thêm nhiều dòng dữ liệu vào bảng

- ✓ Sử dụng để thêm 1 hay nhiều dòng vào bảng

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

```
CREATE TABLE THONGKE_PB
```

```
(  
    TENPHG VARCHAR(20),  
    SL_NV INT,  
    LUONG_TC INT  
)
```

```
INSERT INTO THONGKE_PB(TENPHG, SL_NV, LUONG_TC)  
    SELECT TENPB, COUNT(MANV), SUM(LUONG)  
    FROM NHANVIEN, PHONGBAN  
    WHERE PHG=MAPHG  
    GROUP BY TENPB
```



# Cập nhật dữ liệu trong bảng

**UPDATE** <tên bảng>

**SET** <tên thuộc tính>=<giá trị mới> ,

<tên thuộc tính>=<giá trị mới> ,

...

[**WHERE** <điều kiện>]

# Cập nhật dữ liệu trong bảng

- ✓ Cập nhật lương cơ bản của tất cả nhân viên tăng 15%

```
UPDATE NHANVIEN  
SET LUONG=LUONG*1.15
```

- ✓ Cập nhật lại địa điểm của dự án có mã 10 thành '*Cần thơ*'

```
UPDATE DUAN  
SET DIADIEM_DA=N'Cần thơ'  
WHERE MADA=10
```

- ✓ Cập nhật ngày sinh của nhân viên có mã 101 thành '*28/10/2001*' và địa chỉ mới là '*95 Thành Thái , P.14, Q.10, TP.HCM*'

```
UPDATE NHANVIEN  
SET NGAYSINH='2001/10/28', DIACHI=N'95 Thành thái, Q.10, TP.HCM'  
WHERE MANV='101'
```

# Cập nhật dữ liệu trong bảng



## Lưu ý

- ✓ 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 **CASCADE**
  - Không cho sửa
  - Sửa luôn những dòng có giá trị đang tham chiếu đến

# Xóa dữ liệu trong bảng

✓ Sử dụng để xóa 1 hay nhiều dòng trong bảng

**DELETE** <tên bảng>

**WHERE** <điều kiện>

Ví dụ:

```
DELETE FROM NHANVIEN
```

```
DELETE FROM NHANVIEN
```

```
WHERE HONV= 'Tran'
```

```
DELETE FROM NHANVIEN
```

```
WHERE MANV= '345345345'
```

# Xóa dữ liệu trong bảng

Ví dụ: Xóa các nhân viên thuộc phòng nghiên cứu

```
DELETE FROM NHANVIEN  
WHERE PHG IN (  
    SELECT MAPHG  
    FROM PHONGBAN  
    WHERE TENPB='Nghien cuu')
```

# Xóa dữ liệu trong bảng

- ✓ 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

## 5. Ngôn ngữ truy vấn dữ liệu có cấu trúc

# Truy vấn dữ liệu



- ✓ Được dùng để đọc, truy xuất và trả về thông tin dựa trên các tiêu chí cụ thể. VD: Ở đâu (**From**)?, Thông tin gì (**Select**)?, Điều kiện chọn (**Where**)?, ...
- ✓ Hợp thành bởi các mệnh đề(**Clauses**) dựa trên các từ khóa :  
**From, Where, Select, Oder by, Group by, ....**
- ✓ Nguyên lý hoạt động dựa trên dự đoán(*Predicates*) của điều kiện dựa trên 3 giá trị logic: true|false|unknown
- ✓ • Cú pháp  
**Clause + Clause + Clause + ...;**



# Câu truy vấn tổng quát

```
SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
[WHERE <điều_kiện>]  
[GROUP BY <tên_cột>]  
[HAVING <điều_kiện>]  
[ORDER BY <tên_cột> ASC | DESC]
```

Trong đó:

<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**

# Truy vấn cơ bản



## Select

- ✓ tương đương phép chiếu trong ĐSQH( $\pi$ )
- ✓ liệt kê các thuộc tính cần hiển thị trong kết quả.

## From

- ✓ liệt kê các quan hệ cần thiết, các phép kết

## Where

- ✓ tương đương với phép chọn trong ĐSQH( $\sigma$ )
- ✓ điều kiện liên quan tới thuộc tính, sử dụng các phép nối luận lý AND, OR, NOT, các phép toán so sánh, BETWEEN

$\pi_L (\sigma_C(R))$

SELECT L

FROM R

WHERE C

# Truy vấn cơ bản

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	GTINH	LUONG	PHG
132458	Nguyen	Thanh	Tung	12/8/1989	123 NVC Q5	Nam	15000	DH
123678	Tran	Ngoc	Lan	23/4/1987	242 NTMK Q3	Nu	17000	QL
345653	Pham	Tri	Hung	12/2/1990	333 PNL Q1	Nam	18000	DH
435635	Nguyen	Minh	Trang	25/7/1991	Dong Nai	Nu	16500	NC
223524	Le	Nhu	Ngoc	30/5/1988	BRVT	Nu	20000	NC

Lấy tất cả các cột của quan hệ NHANVIEN của phòng DH

$\sigma_{PHG='DH'}(NHANVIEN)$

SELECT \*

FROM NHANVIEN

WHERE PHG= 'DH'

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	GTINH	LUONG	PHG
132458	Nguyen	Thanh	Tung	12/8/1989	123 NVC Q5	Nam	15000	DH
345653	Pham	Tri	Hung	12/2/1990	333 PNL Q1	Nam	18000	DH

# Truy vấn cơ bản



Cho biết Họ, Tên, Ngày sinh, Địa chỉ của các nhân viên nam trong công ty

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	GTINH	LUONG	PHG
132458	Nguyen	Thanh	Tung	12/8/1989	123 NVC Q5	Nam	15000	DH
123678	Tran	Ngoc	Lan	23/4/1987	242 NTMK Q3	Nu	17000	QL
345653	Pham	Tri	Hung	12/2/1990	333 PNL Q1	Nam	18000	DH
435635	Nguyen	Minh	Trang	25/7/1991	Dong Nai	Nu	16500	NC
223524	Le	Nhu	Ngoc	30/5/1988	BRVT	Nu	20000	NC

# Truy vấn cơ bản

Cho biết Họ, Tên, Địa, Ngày sinh, Địa chỉ của các nhân viên nam trong công ty

$\pi_{\text{HONV, TENNV, NGSINH, DCHI}} \sigma_{\text{GTINH} = \text{'Nam'}}(\text{NHANVIEN})$

```
SELECT HONV, TENNV, NGSINH, DCHI  
FROM NHANVIEN  
WHERE GTINH = 'Nam'
```

HONV	TENNV	NGSINH	DCHI
Nguyen	Tung	12/8/1989	123 NVC Q5
Pham	Hung	12/2/1990	333 PNL Q1

# Truy vấn cơ bản



Cho biết Mã NV, Họ, Tên, Địa chỉ, Ngày sinh của các nhân viên nữ thuộc phòng Nghiên cứu trong công ty

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	GTINH	LUONG	PHG
132458	Nguyen	Thanh	Tung	12/8/1989	123 NVC Q5	Nam	15000	DH
123678	Tran	Ngoc	Lan	23/4/1987	242 NTMK Q3	Nu	17000	QL
345653	Pham	Tri	Hung	12/2/1990	333 PNL Q1	Nam	18000	DH
435635	Nguyen	Minh	Trang	25/7/1991	Dong Nai	Nu	16500	NC
223524	Le	Nhu	Ngoc	30/5/1988	BRVT	Nu	20000	NC

# Truy vấn cơ bản

Cho biết Mã NV Họ, Tên, Địa, Ngày sinh, Địa chỉ của các nhân viên nữ trong công ty

$\rho_{\text{MANV,HỌ,TÊN}}(\pi_{\text{MANV,HONV,TENNV}}(\sigma_{\text{GTINH}=\text{'Nu'} \wedge \text{PHG}=\text{'NC'}}(\text{NHANVIEN})))$

```
SELECT MANV, HONV AS [HỌ], TENNV AS [TÊN]  
FROM NHANVIEN  
WHERE PHAI= 'Nu' AND PHG = 'NC'
```

MaNV	HỌ	TÊN
435635	Nguyen	Trang
223524	Le	Ngoc

# Truy vấn cơ bản



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

MaNV	LUONG 10%
132458	16500
345653	19800



# Truy vấn cơ bản



Loại bỏ các giá trị trùng nhau

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

MaNV	LUONG
132458	15000
345653	18000

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	GTINH	LUONG	PHG
132458	Nguyen	Thanh	Tung	12/8/1989	123 NVC Q5	Nam	15000	DH
123678	Tran	Ngoc	Lan	23/4/1987	242 NTMK Q3	Nu	17000	QL
345653	Pham	Tri	Hung	12/2/1990	333 PNL Q1	Nam	18000	DH
435635	Nguyen	Minh	Trang	25/7/1991	Dong Nai	Nu	16500	NC
223524	Le	Nhu	Ngoc	30/5/1988	BRVT	Nu	20000	NC
232332	Tran	Thanh	Minh	3/4/1993	222 NTT q4	Nam	15000	DH

# Truy vấn cơ bản

Sử dụng các phép toán AND, OR

VD: Cho biết MaNV và Tên nhân viên làm ở phòng Nghiên cứu

$R1 \leftarrow \text{NHANVIEN} \bowtie_{\text{PHG}=\text{MAPHG}} \text{PHONGBAN}$

$KQ \leftarrow \pi_{\text{MANV}, \text{TENNV}} (\sigma_{\text{TENPB}=\text{'Nghiên cứu'}}(R1))$

Biểu thức  
luận lý

SELECT MANV, TENNV

FROM NHANVIEN, PHONGBAN

WHERE TENPB=N'Nghiên cứu'

AND PHG=MAPHG

TRUE

AND

TRUE

# Truy vấn cơ bản



```
SELECT MANV, TENNV  
FROM NHANVIEN, PHONGBAN  
WHERE (TENPB=N'Nghien cứu' OR TENPB=N'Quản lý') AND PHG=MAPHG
```

- ✓ 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.

# Truy vấn cơ bản – Between

```
SELECT MANV, TENNV, LUONG  
FROM NHANVIEN  
WHERE LUONG >= 20000 AND  
LUONG <= 30000
```



```
SELECT MANV, TENNV, LUONG  
FROM NHANVIEN  
WHERE LUONG BETWEEN 20000 AND 30000
```

# Truy vấn cơ bản – Not Between

```
SELECT MANV, TENNV, LUONG  
FROM NHANVIEN  
WHERE LUONG<20000 AND  
LUONG>30000
```



```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE LUONG NOT BETWEEN 20000 AND 30000
```

# Truy vấn cơ bản – Like

---



## ✓ Toán tử LIKE

- ✓ So sánh chuỗi tương đối
- ✓ Cú pháp: s LIKE p, p có thể chứa % hoặc \_
- ✓ % : thay thế một chuỗi ký tự bất kỳ
- ✓ \_ : thay thế một ký tự bất kỳ



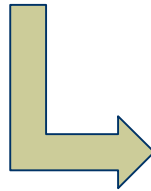
# Truy vấn cơ bản – Like



✓ VD: Hiển thị sản phẩm có masp bắt đầu là B, kết thúc là 1  
SANPHAM

masp	tensp	dvt	nuocsx	gia
BB21	But bi	hop	Thai Lan	100000
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	2500
ST04	So tay	quyen	Thai Lan	55000
ST01	So tay mong	quyen	Thai Lan	20000

Select masp, tensp  
From SANPHAM  
Where masp like 'B%1'



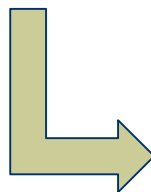
masp	tensp
BB21	But bi
BC01	But chi

# Truy vấn cơ bản – Like

✓ VD: Hiển thị sản phẩm có masp bắt đầu là B, kết thúc là 1  
SANPHAM

masp	tensp	dvt	nuocsx	gia
BB21	But bi	hop	Thai Lan	100000
BC01	But chi	cay	Singapore	3000
BC02	But chi	cay	Singapore	2500
ST04	So tay	quyen	Thai Lan	55000
ST01	So tay mong	quyen	Thai Lan	20000

Select masp, tensp  
From SANPHAM  
Where masp like 'B\_01'



masp	tensp
BC01	But chi



# Truy vấn cơ bản – Like

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HoTen LIKE 'Nguyen _____'
```

4 ký tự bất kỳ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HoTen LIKE 'Nguyen%'
```

Chuỗi bất kỳ

# Truy vấn cơ bản – Like – Not like

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HoTen LIKE 'Nguyen _____'
```

4 ký tự bất kỳ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HoTen LIKE 'Nguyen%'
```

Chuỗi bất kỳ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HoTen NOT LIKE 'Nguyen'
```

# Truy vấn cơ bản – Escape

MANV	TENNV
GV001	Nguyễn Văn A
<b>GV002</b>	<b>Nguyễn_Văn_A</b>

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE TENNV LIKE N'Nguyễn_%'
```

MANV	TENNV
GV001	Nguyễn Văn A
GV002	Nguyễn_Văn_A

```
SELECT MANV, TENNV  
FROM NHANVIEN
```

```
WHERE DCHI LIKE N'Nguyễn![_]%'  
ESCAPE '!'
```

MANV	TENNV
<del>GV001</del>	<del>Nguyễn Văn A</del>
GV002	Nguyễn_Văn_A

# Truy vấn cơ bản – Escape



Symbol	Meaning
LIKE '5[%]'	5%
LIKE '[_]n'	_n
LIKE '[a-cdf]'	a, b, c, d, or f
LIKE '[-acdf]'	-, a, c, d, or f
LIKE '[ [ ]'	[
LIKE ']'	]
LIKE 'abc[_]d%'	abc_d and abc_de
LIKE 'abc[def]'	abcd, abce, and abcf

# Truy vấn cơ bản – Ngày giờ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE NGSINH = '1987-04-23'
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE NGSINH BETWEEN '1987-04-23' AND '1992-05-30'
```

'1987-04-23' YYYY-MM-DD

'23/04/1987' DD/MM/YYY

'1987-04-23' YYYY-MM-DD

'April 23, 1987'

'1987-04-23'

'17:30:00'

'05:30 PM'

# Truy vấn cơ bản – 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
  - $x + 3$  là một biểu thức không hợp lệ trong SQL
- ✓ 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

# Truy vấn cơ bản – Null



```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE MA_NQL IS NULL
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE MA_NQL IS NOT NULL
```

# Truy vấn cơ bản – Tên bí danh



```
SELECT TENPHG, DIADIEM  
FROM PHONGBAN, DDIEM_PHG  
WHERE MAPHG=MAPHG
```



```
SELECT TENNV, NGSINH, TENTN,  
NGSINH  
FROM NHANVIEN, THANNHAN  
WHERE MANV=MA_NVIEEN
```

```
SELECT TENPHG, DIADIEM  
FROM PHONGBAN PB, DDIEM_PHG DD  
WHERE PB.MAPHG=DD.MAPHG
```

```
SELECT TENNV, NV.NGSINH, TENTN,  
TN.NGSINH  
FROM NHANVIEN NV, THANNHAN TN  
WHERE MANV=MA_NVIEEN
```



# Truy vấn cơ bản – **order by**



✓ Dùng để hiển thị kết quả câu truy vấn theo một thứ tự nào đó

**SELECT** <danh sách các cột>

**FROM** <danh sách các bảng>

**WHERE** <điều kiện>

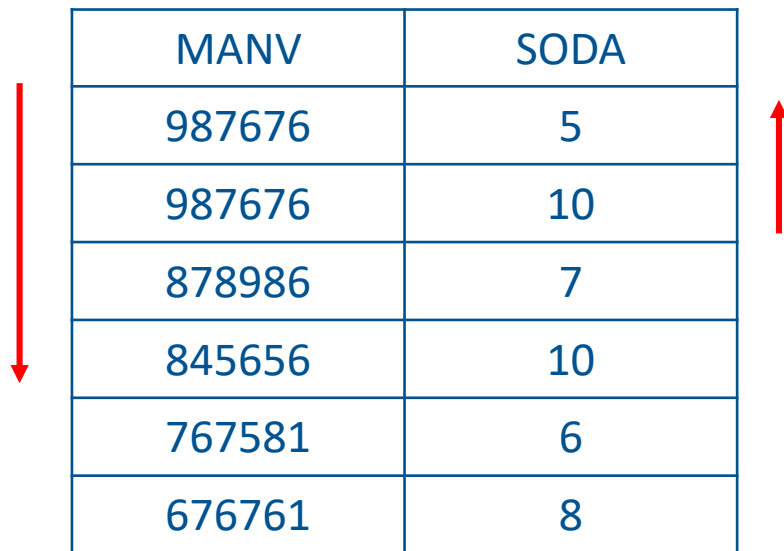
**ORDER BY** <danh sách các cột>

- **ASC**: tăng (mặc định)
- **DESC**: giảm

# Truy vấn cơ bản – order by



```
SELECT MANV, SODA  
FROM PHANCONG  
ORDER BY MANV DESC, SODA
```



MANV	SODA
987676	5
987676	10
878986	7
845656	10
767581	6
676761	8

# Bài tập

## Bài 02:

**GIÁO VIÊN** (MÃ GV, HỌ TÊN, LƯƠNG, PHÁI, NGÀY SINH, SỐ NHÀ, ĐƯỜNG, QUẬN, THÀNH PHỐ, MÃ BM, MÃ QL)

**KHOA** (MÃ KHOA, TÊN KHOA, NĂM TL, ĐIỆN THOẠI, PHÒNG, MÃ GV, NGÀY NHẬN CHỨC)

**BỘ MÔN** (MÃ BM, TÊN BM, PHÒNG, ĐIỆN THOẠI, MÃ GV, NGÀY NHẬN CHỨC, MÃ KHOA)

**ĐỀ TÀI** (MÃ ĐT, TÊN ĐT, NGÀY BĐ, NGÀY KT, KINH PHÍ, CẤP QL, MÃ GV, MÃ CD)

**CHỦ ĐỀ** (MÃ CD, TÊN CD)

**THAM GIA** (SỐ TT, MÃ ĐỀ TÀI, MÃ GV, PHỤ CẤP, KẾT QUẢ)

**CÔNG VIỆC** (SỐ TT, MÃ ĐT, TÊN, NGÀY KT, NGÀY BĐ)

**ĐIỆN THOẠI** (MÃ GV, ĐIỆN THOẠI)

1. Cho biết tên của bộ môn và tên của trưởng bộ môn của những bộ môn thuộc khoa CNTT (mã khoa)
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.
3. 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.

# Bài tập

---



1. Với những đề án ở ‘Ha Noi’, cho biết mã đề án, mã phòng ban chủ trì đề án, họ tên trưởng phòng cùng với ngày sinh và địa chỉ của người ấy
2. Tìm họ tên của nhân viên phòng số 5 có tham gia vào đề án “Sản phẩm X” với số giờ làm việc trên 10 giờ
3. Tìm họ tên của từng nhân viên và người phụ trách trực tiếp nhân viên đó
4. Tìm họ tên của những nhân viên được “Nguyen Thanh Tung” phụ trách trực tiếp

# Bài tập

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

```
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
```

```
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
```

# Phép toán tập hợp

---

- ✓ 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 – Phép hội

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

**UNION [ALL]**

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

Ví dụ: Cho biết các mã đề án có Nhân viên với họ là ‘Nguyễn’ tham gia đề án hoặc Trưởng phòng chủ trì đề án với họ là ‘Nguyễn’

```
SELECT MADA
FROM NHANVIEN, PHANCONG
WHERE MANV=MA_NVIENT AND HONV=N'Nguyễn'
UNION
SELECT MADA
FROM NHANVIEN, PHONGBAN, DEAN
WHERE MANV=TRPHG AND MAPHG=PHONG AND HONV=N'Nguyễn'
```

# Phép toán tập hợp – Phép giao



```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
INTERSECT [ALL]  
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

Ví dụ: Tìm nhân viên có người thân cùng tên và cùng giới tính

```
SELECT TENNV, PHAI, MANV FROM NHANVIEN  
INTERSECT  
SELECT TENTN, PHAI, MA_NVIENTH FROM THANNHAN
```

```
SELECT NV.*  
FROM NHANVIEN NV, THANNHAN TN  
WHERE NV.MANV=TN.MA_NVIENTH  
AND NV.TENNV=TN.TENTN AND NV.PHAI=TN.PHAI
```



# Phép toán tập hợp – Phép trừ

```
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>  
EXCEPT [ALL]  
SELECT <ds cột> FROM <ds bảng> WHERE <điều kiện>
```

Ví dụ: Tìm những nhân viên không có thân nhân nào

```
SELECT MANV FROM NHANVIEN  
EXCEPT  
SELECT MA_NVIENT AS MANV FROM THANNHAN
```

```
SELECT MANV  
FROM NHANVIEN  
WHERE MANV NOT IN  
    (SELECT MANV FROM THANNHAN)
```

# Bài tập

---

1. 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
2. Tìm những giáo viên vừa là trưởng bộ môn vừa chủ nhiệm đề tài

# Bài tập

1. SELECT MAGV FROM GIAOVIEN

WHERE HOTEN LIKE N'Nguyễn%' AND LUONG > 200000

UNION

SELECT TRUONGBM FROM BOMON

WHERE YEAR(NGAYNHANCHUC) >= 1995

2.

SELECT TRUONGBM FROM BOMON

INTERSECT

SELECT GVCNDT FROM DETAI

SELECT BM.TRUONGBM FROM BOMON BM, DETAI DT

WHERE BM.TRUONGBM = DT.GVCNDT

# Truy vấn lồng



- ✓ Các lệnh **SELECT** có thể lồng nhau ở nhiều mức
- ✓ Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng **phép nối logic**
- ✓ Câu truy vấn con thường **trả về một tập** các giá trị
- ✓ 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**

# Truy vấn lồng

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)

# Truy vấn lồng

Ví dụ: Tìm nhân viên có lương cao nhất

```
SELECT *  
FROM NHANVIEN  
WHERE LUONG >= ALL  
      (SELECT LUONG FROM NHANVIEN)
```

B1: Chọn ra những lương không phải là lớn nhất

**$R1 \leftarrow (\pi_{LUONG}(NHANVIEN))$**

**$R2 \leftarrow \sigma_{NHAN\_VIEN.LUONG < R1.LUONG}(NHANVIEN \times R1)$**

**$R3 \leftarrow \pi_{NHAN\_VIEN.LUONG}(R2)$**

B2: Lấy tập hợp lương trừ đi lương trong R3

**$KQ \leftarrow \pi_{LUONG}(NHANVIEN) - R3$**

# Truy vấn lồng

Có 2 loại truy vấn lồng:

## 1. 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

## 2. 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

# Truy vấn lồng– Truy vấn lồng phân cấp



VD: Hãy tìm Đề án mà nhân viên tham gia đề án có họ Nguyễn HOẶC Đề án có trưởng phòng chủ trì đề án có họ Nguyễn

```
SELECT MADA
FROM NHANVIEN NV, PHANCONG PC
WHERE NV.MANV=PC.MA_NVIENT AND NV.HONV=N'Nguyễn'
UNION
SELECT MADA
FROM NHANVIEN NV, PHONGBAN PB, DEAN DA
WHERE NV.MANV=PB.TRPHG AND PB.MAPHG=DA.PHONG
AND NV.HONV=N'Nguyễn'
```



# Truy vấn lồng – Truy vấn lồng phân cấp



VD: Hãy tìm Đề án mà nhân viên tham gia đề án có họ Nguyễn HOẶC Đề án có trưởng phòng chủ trì đề án có họ Nguyễn

```
SELECT DISTINCT TENDA
```

```
FROM DEAN
```

```
WHERE MADA IN (
```

```
    SELECT MADA  
    FROM NHANVIEN, PHANCONG  
    WHERE MANV=MA_NVIENT AND HONV='Nguyễn' )
```

```
OR MADA IN (
```

```
    SELECT MADA  
    FROM NHANVIEN, PHONGBAN PB, DEAN DA  
    WHERE MANV=TRPHG AND PB.MAPHG=DA.PHONG  
    AND HONV='Nguyễn' )
```

# Truy vấn lồng – Truy vấn lồng phân cấp



VD: Hãy tìm những nhân viên không có thân nhân

```
SELECT *  
FROM NHANVIEN  
WHERE MANV <> ALL  
(  
    SELECT MA_NVIEN  
    FROM THANNHAN  
)
```

```
SELECT *  
FROM NHANVIEN  
WHERE MANV NOT IN  
(  
    SELECT MA_NVIEN  
    FROM THANNHAN  
)
```

# Truy vấn lồng – Truy vấn lồng phân cấp



VD: Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên phòng QL

```
SELECT *  
FROM NHANVIEN  
WHERE LUONG > ANY  
(  
    SELECT LUONG  
    FROM NHANVIEN  
    WHERE PHG= 'QL'  
)
```

```
SELECT NV1.*  
FROM NHANVIEN NV1, NHANVIEN NV2  
WHERE NV1.LUONG > NV2.LUONG AND  
NV2.PHG= 'QL'
```

# Truy vấn lồng – Truy vấn lồng phân cấp



VD: Tìm những nhân viên có lương lớn hơn lương của tất cả nhân viên phòng QL

```
SELECT *  
FROM NHANVIEN  
WHERE LUONG > ALL  
(  
    SELECT LUONG  
    FROM NHANVIEN  
    WHERE PHG= 'QL'  
)
```

# Truy vấn lồng – Truy vấn lồng phân cấp



VD: Tìm những trưởng phòng có tối thiểu một thân nhân

# Truy vấn lồng – Truy vấn lồng phân cấp



VD: Tìm những trưởng phòng có tối thiểu một thân nhân

```
SELECT *  
FROM NHANVIEN  
WHERE MANV IN (SELECT MA_NVIEN FROM THANNHAN)  
AND MANV IN (SELECT TRPHG FROM PHONGBAN)
```

# Phép toán tập hợp – Tương quan



## ✓ 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

# Truy vấn lồng – Truy vấn lồng tương quan



VD: Tìm nhân viên thuộc phòng Nghiên cứu

```
SELECT MANV, TENNV  
FROM NHANVIEN, PHONGBAN  
WHERE TENPHG=N'Nghiên cứu' AND PHONG=MAPHG
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE EXISTS  
(  
    SELECT *  
    FROM PHONGBAN  
    WHERE TENPHG=N'Nghiên cứu')
```



# Truy vấn lồng – Truy vấn lồng tương quan



VD: Tìm nhân viên có thân nhân cùng tên, cùng phái

```
SELECT *  
FROM NHANVIEN NV  
WHERE EXISTS  
(  
    SELECT *  
    FROM THANNHAN TN  
    WHERE NV.MANV=TN.MA_NVIAN  
    AND NV.TENNV=TN.TENTN  
    AND NV.PHAI=TN.PHAI  
)
```

# Phép toán tập hợp – Sử dụng NOT EXISTS



```
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
    )
)
```

# Phép toán tập hợp – Sử dụng EXCEPT



```
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
  )
)
```

# Phép toán tập hợp – Sử dụng EXCEPT



```
SELECT R.A,R.B,R.C
FROM R
WHERE R.A+R.B+R.C NOT IN
(SELECT Q2.A+Q2.B+Q2.C
FROM (SELECT R.A,R.B,R.C,S.* FROM R,S) Q2
WHERE Q2.A+Q2.B+Q2.C+Q2.D+Q2.E NOT IN
(SELECT R.A+R.B+R.C+R.D+R.E FROM R)
) -- Q3 LÀ A,B,C KHÔNG THAM GIA ĐẦY ĐỦ VÀO S
```

# Phép toán tập hợp – Phép chia



Ví dụ: Tìm tên các nhân viên được phân công làm tất cả các đề án

- Tìm tên các nhân viên mà không có đề án nào là không được phân công làm
- Tập bị chia: PHANCONG(MA\_NVIEN, MADA)
- Tập chia: DEAN(MADA)
- Tập kết quả: KQ(MA\_NVIEN)
- Kết KQ với NHANVIEN để lấy ra TENNV

# Phép toán tập hợp – Phép chia



```
SELECT NV.TENNV
FROM NHANVIEN NV, PHANCONG PC1
WHERE NV.MANV=PC1.MANV
AND NOT EXISTS
(
  SELECT *
  FROM DEAN DA
  WHERE NOT EXISTS
  (
    SELECT *
    FROM PHANCONG PC2
    WHERE PC2.MADA=DA.MADA
    AND PC1.MANV=PC2.MANV
  )
)
```

# Phép toán tập hợp – Hàm kết hợp



## ✓ COUNT

1. COUNT(\*) đếm số dòng
2. COUNT(<tên thuộc tính>) đếm số giá trị khác NULL của thuộc tính
3. 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

# Phép toán tập hợp – Hàm kết hợp



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

```
SELECT COUNT(*) AS SL_NV
FROM NHANVIEN, PHONGBAN
WHERE PHG=MAPHG AND TENPHG=N'Nguyễn cứu'
```



# Phép toán tập hợp – Gom nhóm

**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

# Phép toán tập hợp – Gom nhóm

```
SELECT PHG, COUNT(*) AS SL_NV  
FROM NHANVIEN  
GROUP BY PHG
```

```
SELECT TENPHG, COUNT(*) AS SL_NV  
FROM NHANVIEN, PHONGBAN  
WHERE PHG=MAPHG  
GROUP BY TENPHG
```

*Ghi chú:* SELECT cột nào GROUP BY cột đó (ngoại trừ hàm kết hợp)

# Phép toán tập hợp – Gom nhóm

**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>

# Phép toán tập hợp – Gom nhóm



VD: Cho biết những phòng ban (TENPHG) có lương trung bình của các nhân viên lớn hơn 20000

```
SELECT PHONG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN  
GROUP BY PHONG  
HAVING AVG(LUONG) > 20000
```

```
SELECT TENPHG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN, PHONGBAN  
WHERE PHG=MAPHG  
GROUP BY TENPHG  
HAVING AVG(LUONG) > 20000
```

# Phép toán tập hợp – Gom nhóm



## ✓ 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

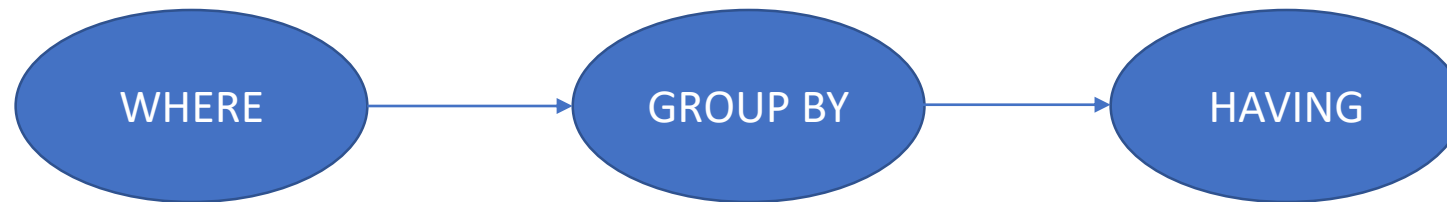
# Phép toán tập hợp – Gom nhóm



- ✓ Chọn ra những dòng thỏa điều kiện trong mệnh đề WHERE
- ✓ Những dòng này sẽ được gom thành nhiều nhóm tương ứng với mệnh đề GROUP BY
- ✓ Áp dụng các hàm kết hợp cho mỗi nhóm
- ✓ Bỏ qua những nhóm không thỏa điều kiện trong mệnh đề HAVING
- ✓ Rút trích các giá trị của các cột và hàm kết hợp trong mệnh đề SELECT

# Phép toán tập hợp – Gom nhóm

Thứ tự câu lệnh



# Một số dạng truy vấn khác

---

- ✓ Truy vấn con ở mệnh đề FROM
- ✓ Điều kiện kết hợp ở mệnh đề FROM
  - Phép kết tự nhiên
  - Phép kết ngoài



# Một số dạng truy vấn khác

- ✓ 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>

# Một số dạng truy vấn khác

## ✓ Kết bằng

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

## ✓ Kết ngoài – Full join

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

# Cấu trúc CASE

- ✓ 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
```

# Cấu trúc CASE

```
SELECT HONV, TENNV
FROM NHANVIEN
WHERE YEAR(GETDATE()) - YEAR(NGSINH) >= (
    CASE PHAI
        WHEN 'Nam' THEN 65
        WHEN 'Nu' THEN 60
    END )
```

VD: Tính năm về hưu theo giới tính (nam trên 65, nữ trên 60 tuổi)

```
SELECT HONV, TENNV,
( CASE PHAI
    WHEN 'Nam' THEN YEAR(NGSINH) + 65
    WHEN 'Nu' THEN YEAR(NGSINH) + 60
    END ) AS NAMVEHUU
FROM NHANVIEN
```

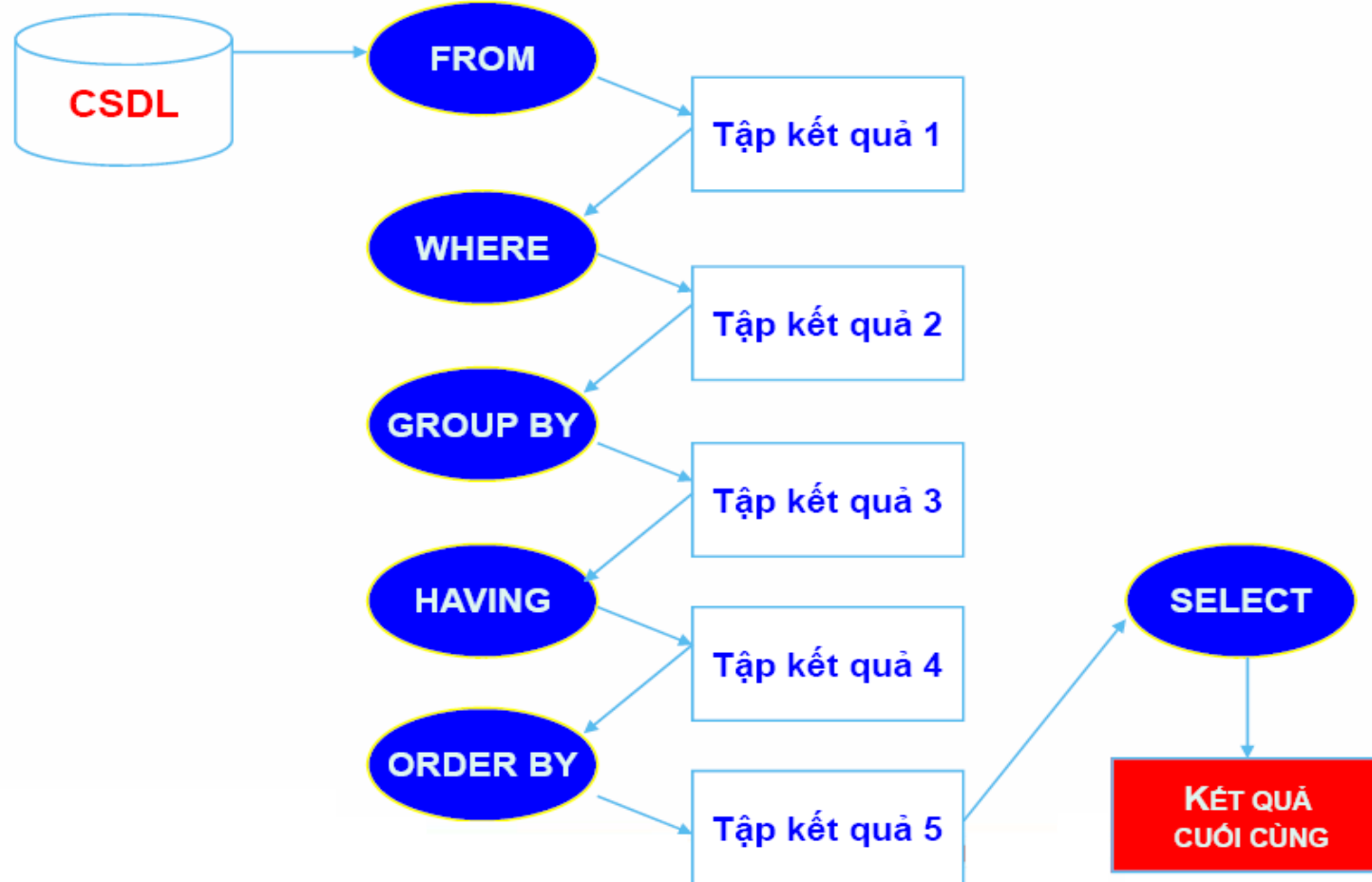
# Cấu trúc CASE

```
SELECT HONV, TENNV
FROM NHANVIEN
WHERE YEAR(GETDATE()) - YEAR(NGSINH) >= (
    CASE PHAI
        WHEN 'Nam' THEN 65
        WHEN 'Nu' THEN 60
    END )
```

```
SELECT HONV, TENNV,
( CASE PHAI
    WHEN 'Nam' THEN YEAR(NGSINH) + 65
    WHEN 'Nu' THEN YEAR(NGSINH) + 60
    END ) AS NAMVEHUU
FROM NHANVIEN
```

# Quy trình xử lý câu truy vấn

<b>SELECT</b>	Col1, Col2,...ColN	(1)
<b>FROM</b>	TableA, Table B,...	(2)
<b>WHERE</b>	Dk1, Dk2, Dk3,...	(3)
<b>GROUP BY</b>	ColA, CLoB,...	(4)
<b>HAVING</b>	DKG1, DKG2,...	(5)
<b>ORDER BY</b>	ColX, ColY,...	(6)



Một số vấn đề  
khác

# Khung nhìn - View

---



- ✓ 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





# Khung nhìn - View

---



Tác dụng của 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



# Khung nhìn - View

---



```
CREATE VIEW NV_NC AS
SELECT MANV, HONV, TENLOT, TENVN
FROM NHANVIEN
WHERE PHG= 'NC'
```

---

```
CREATE VIEW TONGLNG_SLVN_PB AS
SELECT MAPHG, TENPHG, COUNT(*) AS SLNV, SUM(LUONG) AS TONGLNG
FROM NHANVIEN, PHONGBAN
WHERE PHG=MAPHG
GROUP BY TENPHG, MAPHG
```

# Truy vấn trên khung nhìn



- ✓ Tuy không chứa dữ liệu nhưng có thể thực hiện các câu truy vấn trên khung nhìn
- ✓ Có thể viết câu truy vấn dữ liệu từ khung nhìn và bảng

```
SELECT TENVN  
FROM NV_NC  
WHERE HONV LIKE 'NGUYEN'
```

```
SELECT HONV, TENVN, TENDA, THOIGIAN  
FROM NV_NC, PHANCONH, DEAN  
WHERE MANV=MA_NVIENT AND SODA=MADA
```

# Cập nhật trên khung nhìn



- ✓ 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

# Cập nhật trên khung nhìn

✓ VD: Sửa lại họ cho nhân viên mã '123456789' ở phòng NC là 'Pham'

```
UPDATE NV_NC  
SET HONV= 'PHAM'  
WHERE MANV= '123456789'
```

# Chỉ mục - Index

- ✓ 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 NHANVIEN  
WHERE PHG= 'NC' AND PHAI= 'Nu'
```

# Chỉ mục - Index

## ✓ Cú pháp

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

```
DROP <tên chỉ mục>
```

## ✓ Ví dụ

```
CREATE INDEX PHG_IND ON NHANVIEN(PHG)
```

```
CREATE INDEX PHG_PHAII_IND ON NHANVIEN(PHG, PHAI)
```

# Chỉ mục - Index

---

## ✓ 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???





Hết