



# CƠ SỞ DỮ LIỆU

TRAM DOAN

**DATABASE**



# Phần 6

## NGÔN NGỮ TRUY VẤN - SQL

### Week 7 + 8

- Giới thiệu
- Các lệnh định nghĩa dữ liệu
- Các lệnh truy vấn đơn giản
- Các lệnh truy vấn trên nhiều bảng
- Các truy vấn lồng
- Gom nhóm và kết hợp
- Các lệnh cập nhật dữ liệu
- Các vấn đề khác



## Giới thiệu

- SQL ( Structured Query Language) hay gọi SEQUEL ( **S**tructured English **QU**ery **L**anguage)
- Được phát triển bởi IBM ( 1970s)
- Có nhiều phiên bản được tạo bởi tổ chức ANSI ( the American National Standards Institute) và ISO ( the International Standards Organization), gồm các phiên bản:
  - SQL-86
  - SQL-92
  - SQL-99

# Giới thiệu

- **SQL:**
  - Định nghĩa dữ liệu ( DDL - Data Definition Language)
  - Thao tác dữ liệu ( DML - Data Manipulation Language)
  - Định nghĩa khung nhìn
  - Ràng buộc toàn vẹn
  - Phân quyền và bảo mật
  - Điều khiển giao tác
- **Thuật ngữ trong SQL:**
  - Bảng ~ Quan hệ
  - Cột ~ thuộc tính
  - Dòng ~ Bộ

## Định nghĩa dữ liệu và kiểu dữ liệu

- **Định nghĩa dữ liệu:**

- Câu lệnh định nghĩa dữ liệu trong SQL là **CREATE**, được sử dụng để tạo:
  - Lược đồ ( schema)
  - Bảng ( Table)
  - Khung nhìn ( View)
  - Ràng buộc ( assertion, trigger)

## Định nghĩa dữ liệu và kiểu dữ liệu

- Định nghĩa dữ liệu:

- Gồm:

- CREATE DATABASE (tạo cơ sở dữ liệu)
    - CREATE TABLE(tạo bảng)
    - DROP TABLE(xoá bảng)
    - ALTER TABLE( sửa bảng)
    - CREATE DOMAIN( tạo miền giá trị)

# Kiểu dữ liệu

Kiểu chuỗi	Kiểu số	Ngày và thời gian
CHAR( <i>kich_thuoc</i> ) NCHAR( <i>kich_thuoc</i> )	BIT	DATE "YYYY-MM-DD"
VARCHAR( <i>kich_thuoc</i> ) NVARCHAR( <i>kich_thuoc</i> )	TINYINT SMALLINT INT BIGINT	DATETIME "YYYY-MM-DD hh:mm:ss[.mmm]"
TEXT NTEXT	DECIMAL( <i>m,d</i> ) <i>m</i> : tổng số lượng các số <i>d</i> : số lượng các số nằm sau dấu phẩy	DATETIME2(chính xác tới số thập phân của giây) "YYYY-MM-DD hh:mm:ss[số giây thập phân]"
BINARY VARBINARY	FLOAT( <i>n</i> )	SMALLDATETIME "YYYY-MM hh:mm:ss"



# TRUY VẤN CƠ BẢN

- SELECT FROM WHERE



# Truy vấn cơ bản

**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, BETWEEN

# Truy vấn cơ bản – ĐSQH & SQL

$\pi$

**SELECT** <danhsách\_các\_cột>

$\times$

**FROM** <danhsách\_các\_bảng>

$\sigma$

**WHERE** <điều\_kiện>

**SELECT** L  
**FROM** R  
**WHERE** C



$\pi_L(\sigma_C(R))$

# Truy vấn cơ bản

- Lấy ra tất cả các cột

```
SELECT *  
FROM PHONGBAN  
WHERE NamTL >= 1985
```

	MaPB	TenPB	TrPhong	NgNhanChuc	NamTL
1	4	Hanh chinh	987654321	1995-01-01 00:00:00.000	1990
2	5	Nghien cuu	333445555	1988-06-22 00:00:00.000	1985

$\pi(\sigma_{\text{NamTL} \geq 1985}(\text{PHONGBAN}))$

# Truy vấn cơ bản

- Lấy ra một số cột

```
SELECT MaPB, TenPB  
FROM PHONGBAN  
WHERE NamTL >= 1985
```

	MaPB	TenPB
1	4	Hanh chinh
2	5	Nghien cuu

$\pi_{\text{MaPB, TenPB}}(\sigma_{\text{NamTL} \geq 1985}(\text{PHONGBAN}))$

# Truy vấn cơ bản

- Tên bí danh

```
SELECT  
    MaPB as 'Mã phòng ban',  
    TenPB as 'Tên phòng ban'  
FROM PHONGBAN  
WHERE NamTL >= 1985
```

	MaPB	TenPB
1	4	Hanh chinh
2	5	Nghien cuu

	Mã phòng ban	Tên phòng ban
1	4	Hanh chinh
2	5	Nghien cuu

$\rho_{\text{Mã phòng ban, Tên phòng ban}} \pi_{\text{MaPB, TenPB}} (\sigma_{\text{NamTL} \geq 1985} (\text{PHONGBAN}))$

# Truy vấn cơ bản

● Mở rộng

```
SELECT  
    MaNV,  
    Ho + ' ' + TenDem + ' ' + Ten as HoTen  
FROM NHANVIEN
```

	MaNV	HoTen
1	123456789	Nguyen Bao Hung
2	333445555	Phan Van Nghia
3	453453453	Hoang Kim Yen
4	666884444	Tran Van Nam
5	888665555	Le Van Bo
6	987654321	Du Thi Hau
7	987987987	Nguyen Van Giap
8	999887777	Au Thi Vuong

# Truy vấn cơ bản

● Mở rộng

```
SELECT  
  MaNV,  
  Ten,  
  Luong,  
  Luong*1.1 as LuongMoi  
FROM NHANVIEN
```

	MaNV	Ten	Luong	LuongMoi
1	123456789	Hung	30000	33000.0
2	333445555	Nghia	40000	44000.0
3	453453453	Yen	25000	27500.0
4	666884444	Nam	38000	41800.0
5	888665555	Bo	55000	60500.0
6	987654321	Hau	43000	47300.0
7	987987987	Giap	25000	27500.0
8	999887777	Vuong	25000	27500.0

# Truy vấn cơ bản

- Loại bỏ các dòng trùng nhau

```
SELECT  
  Luong  
FROM NHANVIEN
```

	Luong ▼
1	30000
2	40000
3	25000
4	38000
5	55000
6	43000
7	25000
8	25000

```
SELECT Distinct  
  Luong  
FROM NHANVIEN
```

	Luong ▼
1	25000
2	30000
3	38000
4	40000
5	43000
6	55000



# Truy vấn cơ bản – Mệnh đề WHERE

- Phép toán:  $>$ ,  $<$ ,  $\geq$ ,  $\leq$ ,  $=$ ,  $\neq$ , LIKE, BETWEEN
- Nối các biểu thức: AND, OR và NOT

$>$

```
SELECT
*
FROM NHANVIEN
WHERE Luong > 25000
```

$<$

```
SELECT
*
FROM NHANVIEN
WHERE Luong < 25000
```

$=$

```
SELECT
*
FROM NHANVIEN
WHERE Luong = 25000
```

$\geq$

```
SELECT
*
FROM NHANVIEN
WHERE Luong >= 25000
```

$\leq$

```
SELECT
*
FROM NHANVIEN
WHERE Luong <= 25000
```

$\neq$ ,  $<>$

```
SELECT
*
FROM NHANVIEN
WHERE Luong != 25000
```

# Truy vấn cơ bản – Mệnh đề WHERE

- Phép toán:  $>$ ,  $<$ ,  $\geq$ ,  $\leq$ ,  $=$ ,  $\neq$ , LIKE, BETWEEN
- Nối các biểu thức: AND, OR và NOT

## AND

SELECT

\*

FROM NHANVIEN

WHERE Luong  $\geq$  30000 AND Luong  $\leq$  50000

## BETWEEN

SELECT

\*

FROM NHANVIEN

WHERE Luong BETWEEN 30000 AND 50000

## OR

SELECT

\*

FROM NHANVIEN

WHERE Luong  $<$  30000 OR Luong  $>$  50000

## NOT BETWEEN

SELECT

\*

FROM NHANVIEN

WHERE Luong NOT BETWEEN 30000 AND 50000

# Truy vấn cơ bản – Mệnh đề WHERE

- Phép toán: >, <, ≥, ≤, =, ≠, LIKE, BETWEEN
- Nối các biểu thức: AND, OR và NOT

## LIKE

SELECT

\*

FROM NHANVIEN

WHERE Ten LIKE 'H\_'

2 ký tự bất kỳ

	MaNV	Ho	TenDem	Ten
1	987654321	Du	Thi	Hau

SELECT

\*

FROM NHANVIEN

WHERE Ten LIKE 'H\_ '

3 ký tự bất kỳ

	MaNV	Ho	TenDem	Ten
1	123456789	Nguyen	Bao	Hung

SELECT

\*

FROM NHANVIEN

WHERE Ten LIKE 'H%'

Chuỗi bất kỳ

	MaNV	Ho	TenDem	Ten
1	123456789	Nguyen	Bao	Hung
2	987654321	Du	Thi	Hau

# Truy vấn cơ bản – Mệnh đề WHERE

- Phép toán: >, <, ≥, ≤, =, ≠, LIKE, BETWEEN
- Nối các biểu thức: AND, OR và NOT

## NOT LIKE

SELECT

\*

FROM NHANVIEN

WHERE Ten NOT LIKE 'H\_'

2 ký tự bất kỳ

SELECT

\*

FROM NHANVIEN

WHERE Ten NOT LIKE 'H\_ '

3 ký tự bất kỳ

SELECT

\*

FROM NHANVIEN

WHERE Ten NOT LIKE 'H%'

Chuỗi bất kỳ

# Truy vấn cơ bản – Mệnh đề WHERE

## NGÀY GIỜ

```
SELECT  
*  
FROM NHANVIEN  
WHERE NgSinh >= '1965-01-01'
```

```
SELECT  
*  
FROM NHANVIEN  
WHERE NgSinh BETWEEN '1965-01-01' AND '1970-01-01'
```



# Truy vấn cơ bản – Mệnh đề WHERE

## NULL

- Sử dụng trong trường hợp:
  - Không biết
  - Không thể áp dụng
  - Không tồn tại
- 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



## Truy vấn cơ bản – Mệnh đề WHERE

### NULL

SELECT

\*

FROM NHANVIEN

WHERE MaGSat IS NULL

SELECT

\*

FROM NHANVIEN

WHERE MaGSat IS NOT NULL

# Truy vấn cơ bản – Truy vấn từ nhiều bảng

SELECT

\*

FROM NHANVIEN, PHONGBAN

NHANVIEN x PHONGBAN

	MaNV	Ho	Ten...	Ten	NgSinh	DiaChi	GTinh	Luong	MaGSat	Phong	MaPB	TenPB	T
1	123456789	Nguyen	Bao	Hung	1965-01-09...	73 Phan D...	Nam	30000	333445555	5	1	Giam doc	
2	123456789	Nguyen	Bao	Hung	1965-01-09...	73 Phan D...	Nam	30000	333445555	5	4	Hanh chinh	
3	123456789	Nguyen	Bao	Hung	1965-01-09...	73 Phan D...	Nam	30000	333445555	5	5	Nghien cuu	
4	333445555	Phan	Van	Nghia	1955-12-08...	63 Tran H...	Nam	40000	888665555	5	1	Giam doc	
5	333445555	Phan	Van	Nghia	1955-12-08...	63 Tran H...	Nam	40000	888665555	5	4	Hanh chinh	
6	333445555	Phan	Van	Nghia	1955-12-08...	63 Tran H...	Nam	40000	888665555	5	5	Nghien cuu	
7	453453453	Hoang	Kim	Yen	1972-07-31...	56 Thich ...	Nu	25000	333445555	5	1	Giam doc	
8	453453453	Hoang	Kim	Yen	1972-07-31...	56 Thich ...	Nu	25000	333445555	5	4	Hanh chinh	
9	453453453	Hoang	Kim	Yen	1972-07-31...	56 Thich ...	Nu	25000	333445555	5	5	Nghien cuu	
10	666884444	Tran	Van	Nam	1962-09-15...	97 Dien B...	Nam	38000	333445555	5	1	Giam doc	
11	666884444	Tran	Van	Nam	1962-09-15...	97 Dien B...	Nam	38000	333445555	5	4	Hanh chinh	



# Truy vấn cơ bản – Truy vấn từ nhiều bảng

SELECT

\*

FROM NHANVIEN, PHONGBAN

WHERE Phong = MaPB

NHANVIEN ⋈ PHONGBAN

MaNV	Ho	TenDem	Ten	NgSinh	DiaChi	GTinh	Luong	MaGSat	Phong	MaPB	TenPB
123456789	Nguyen	Bao	Hung	1965-01-...	73 Phan Dan...	Nam	30000	333445555	5	5	Nghien cuu
333445555	Phan	Van	Nghia	1955-12-...	63 Tran Huy...	Nam	40000	888665555	5	5	Nghien cuu
453453453	Hoang	Kim	Yen	1972-07-...	56 Thich Qu...	Nu	25000	333445555	5	5	Nghien cuu
666884444	Tran	Van	Nam	1962-09-...	97 Dien Bie...	Nam	38000	333445555	5	5	Nghien cuu
888665555	Le	Van	Bo	1937-11-...	45 Ho Van H...	Nam	55000	NULL	1	1	Giam doc
987654321	Du	Thi	Hau	1941-06-...	29 Bach Dan...	Nu	43000	888665555	4	4	Hanh chinh
987987987	Nguyen	Van	Giap	1969-03-...	98 Huynh Va...	Nam	25000	987654321	4	4	Hanh chinh
999887777	Au	Thi	Vuong	1968-01-...	32 Cao Ba N...	Nu	25000	987654321	4	4	Hanh chinh

# Truy vấn cơ bản – Truy vấn từ nhiều bảng

SELECT

\*

FROM NHANVIEN, THANNHAN

WHERE MaNV = MaNV

Nhập nhằng (Ambiguous)???

SELECT

NV.MaNV,

TN.TenTN,

TN.QuanHe

FROM NHANVIEN NV, THANNHAN AS TN

WHERE NV.MaNV = TN.MaNV

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



## Truy vấn cơ bản – Ví dụ

Ví dụ 1: Cho biết danh sách họ tên nhân viên nam có lương trên 25000 mỗi tháng



## Truy vấn cơ bản – Ví dụ

Ví dụ 2: Cho biết tên các phòng ban và họ tên của trưởng phòng ban đó cùng ngày nhận chức của họ.



## Truy vấn cơ bản – Ví dụ

Ví dụ 3: Với những dự án thuộc phòng ‘Nghien cuu’, cho biết mã dự án, tên dự án, địa điểm thực hiện, các nhân viên thực hiện và tổng số giờ thực hiện của mỗi nhân viên

## Truy vấn cơ bản – 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/DESC>
```

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

## Truy vấn cơ bản – Mệnh đề Order by

SELECT

NV.MaNV,

NV.Ho + ' ' + NV.TenDem + ' ' + NV.Ten as HoTen,

NV.NgSinh

FROM NHANVIEN NV

ORDER BY NgSinh

	MaNV	HoTen	NgSinh
1	888665555	Le Van Bo	1937-11-10
2	987654321	Du Thi Hau	1941-06-20
3	333445555	Phan Van Nghia	1955-12-08
4	666884444	Tran Van Nam	1962-09-15
5	123456789	Nguyen Bao Hung	1965-01-09
6	999887777	Au Thi Vuong	1968-01-19
7	987987987	Nguyen Van Giap	1969-03-29
8	453453453	Hoang Kim Yen	1972-07-31

# Phép toán tập hợp

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



## Phép toán tập hợp

```
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>
```

# Phép toán tập hợp

**Ví dụ:** Cho biết mã và họ tên nhân viên có họ Nguyễn và lương trên 25000 hoặc nhân viên là trưởng phòng ban nhận chức sau năm 1995.

SELECT

NV.MaNV,

NV.Ho + ' ' + NV.TenDem + ' ' + NV.Ten AS HoTen

FROM NHANVIEN NV

WHERE Ho = 'Nguyễn' AND Luong > 25000

UNION

SELECT

NV.MaNV,

NV.Ho + ' ' + NV.TenDem + ' ' + NV.Ten AS HoTen

FROM NHANVIEN NV, PHONGBAN PB

WHERE NV.MaNV = PB.TrPhong

AND YEAR(PB.NgNhanChuc) >= 1995

# Phép toán tập hợp

● Ví dụ: Tìm các nhân viên vừa là trưởng phòng ban vừa thực hiện đề án.

```
SELECT TrPhong as NV  
FROM PHONGBAN
```

INTERSECT

```
SELECT  
MaNV  
FROM PHANCONG
```

```
SELECT DISTINCT  
TrPhong AS NV  
FROM PHONGBAN, PHANCONG  
WHERE TrPhong = MaNV
```

# Phép toán tập hợp

- Ví dụ: Tìm các nhân viên không có bất kỳ thân nhân nào

```
SELECT MaNV  
FROM NHANVIEN
```

```
EXCEPT  
SELECT DISTINCT  
MaNV  
FROM THANNHAN
```

# Truy vấn lồng

- Sử dụng khối lệnh SELECT – FROM – WHERE trong mệnh đề WHERE của một câu truy vấn khác.

```
SELECT <danh_sách_các_cột>  
FROM <danh_sách_các_bảng>  
WHERE <so sánh tập hợp > (
```

Truy vấn cha

```
SELECT <danh_sách_các_cột>  
FROM <danh_sách_các_bảng>  
WHERE <điều_kiện > )
```

Truy vấn con

# Truy vấn lồng

- 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 một số toán tử:
    - IN , NOT IN
    - ALL
    - ANY hoặc SOME
  - Kiểm tra sự tồn tại:
    - EXISTS
    - NOT EXISTS

# Truy vấn lồng

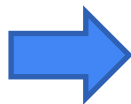
- Có 2 loại truy vấn lồng:
  - Lồng phân cấp:
    - Mệnh đề WHERE của truy vấn con không tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha.
    - Khi thực hiện, câu truy vấn con sẽ được thực hiện trước.
  - Lồng tương quan
    - Mệnh đề WHERE của truy vấn con tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở câu truy vấn cha.
    - Khi thực hiện, câu truy vấn con sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của câu truy vấn cha.

# Truy vấn lồng – Truy vấn lồng phân cấp

- Ví dụ: Cho biết mã nhân viên và tên nhân viên làm trưởng phòng ban nào đó.

```
SELECT  
NV.MaNV,  
NV.Ten  
FROM NHANVIEN NV, PHONGBAN PB  
WHERE NV.MaNV = PB.TrPhong
```

```
SELECT  
MaNV,  
Ten  
FROM NHANVIEN  
WHERE MaNV IN ( SELECT  
TrPhong  
FROM PHONGBAN  
)
```



```
SELECT  
MaNV,  
Ten  
FROM NHANVIEN  
WHERE MaNV IN ('888665555',  
'987654321', '333445555')
```

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



# Truy vấn lồng – Truy vấn lồng phân cấp

- Ví dụ: Tìm các nhân viên họ 'Nguyen' có lương > 25000 hoặc làm trưởng một phòng ban nào đó và có năm nhận chức từ năm 1995.

SELECT

MaNV,

Ten

FROM NHANVIEN

WHERE

MaNV IN (SELECT  
MaNV  
FROM NHANVIEN  
WHERE Ho = 'Nguyen' AND Luong > 25000 )

OR

MaNV IN (SELECT  
TrPhong  
FROM PHONGBAN  
WHERE YEAR(NgNhanChuc) >= 1995)

## Truy vấn lồng – Truy vấn lồng phân cấp

- **Ví dụ:** Tìm các nhân viên không có bất kỳ thân nhân nào

```
SELECT
    MaNV,
    Ho + ' ' + TenDem + ' ' + Ten AS HoTen
FROM NHANVIEN
WHERE MaNV NOT IN (
    SELECT
        MaNV
    FROM THANNHAN
)
```

```
SELECT
    MaNV,
    Ho + ' ' + TenDem + ' ' + Ten AS HoTen
FROM NHANVIEN
WHERE MaNV <> ALL (
    SELECT
        MaNV
    FROM THANNHAN
)
```

## Truy vấn lồng – Truy vấn lồng phân cấp

- Ví dụ: Tìm các nhân viên có lương lớn hơn lương của ít nhất một nhân viên thuộc phòng 'Hanh chinh'.

```
SELECT
```

```
    *
```

```
FROM NHANVIEN
```

```
WHERE Luong > ANY ( SELECT
                      LUONG
                      FROM NHANVIEN NV, PHONGBAN PB
                      WHERE NV.Phong = PB.MaPB
                      AND PB.TenPB = 'Hanh chinh' )
```

## Truy vấn lồng – Truy vấn lồng phân cấp

- Ví dụ: Tìm các nhân viên có lương lớn hơn lương của tất cả các nhân viên thuộc phòng 'Hanh chinh'.

```
SELECT
```

```
    *
```

```
FROM NHANVIEN
```

```
WHERE Luong > ALL ( SELECT
                      LUONG
                      FROM NHANVIEN NV, PHONGBAN PB
                      WHERE NV.Phong = PB.MaPB
                      AND PB.TenPB = 'Hanh chinh' )
```

# Truy vấn lồng – Truy vấn lồng tương quan

- Ví dụ: Tìm các nhân viên là trưởng phòng ban nào đó.

```
SELECT
*
FROM NHANVIEN
WHERE MaNV IN ( SELECT TrPhong FROM PHONGBAN )
```

Truy vấn lồng phân cấp

```
SELECT
*
FROM NHANVIEN NV
WHERE EXISTS ( SELECT
                *
                FROM PHONGBAN PB
                WHERE PB.TrPhong = NV.MaNV )
```

Truy vấn lồng tương quan

Nhân viên là trưởng phòng ban khi tồn tại một phòng ban có TrPhong = MaNV của nhân viên đó.

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

# Truy vấn lồng – Truy vấn lồng tương quan

- **Ví dụ:** Tìm các nhân viên không có bất kỳ thân nhân nào

```
SELECT
*
FROM NHANVIEN NV
WHERE NOT EXISTS (SELECT
*
FROM THANNHAN TN
WHERE NV.MaNV = TN.MaNV)
```

Nhân viên **NV** không có thân nhân nào khi không tồn tạo một dòng nào trong bảng THANNHAN mà có MaNV = **GV.MaNV**

# Truy vấn lồng – Nhận xét

## ● In:

- <tên cột> IN < câu truy vấn con>
- Thuộc tính ở mệnh đề SELECT trong câu truy vấn con phải có cùng kiểu dữ liệu với thuộc tính ở mệnh đề WHERE của câu 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 câu truy vấn con
- Những câu truy vấn có = ANY hay IN đều có thể chuyển thành câu truy vấn có EXISTS
- Cú pháp: ... EXISTS (SELECT ... FROM ... WHERE...)

Subquery

# Hàm tính tổng hợp trong SQL

- 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 tính tổng hợp được đặt ở mệnh đề SELECT hoặc HAVING



# Hàm tính tổng hợp trong SQL

- Ví dụ: Tìm tổng lương, lương cao nhất, lương thấp nhất, và lương trung bình của các nhân viên.

SELECT

```
SUM(LUONG) AS TONGLUONG,  
MAX(LUONG) AS MAX_LUONG,  
MIN(LUONG) AS MIN_LUONG,  
AVG(LUONG) AS AVG_LUONG
```

FROM NHANVIEN

TONGLUONG	MAX_LUONG	MIN_LUONG	AVG_LUONG
281000	55000	25000	35125

# Hàm tính tổng hợp trong SQL

- Ví dụ: Cho biết số lượng giáo viên của phòng 'Nghien cuu'


SELECT

count(\*) AS SL\_NV

FROM NHANVIEN NV, PHONGBAN PB

WHERE NV.Phong = PB.MaPB

AND PB.TenPB = 'Nghien cuu'



Cho biết số lượng nhân viên của từng phòng ban?

## Mệnh đề GROUP BY

Cú pháp:

```
SELECT <danh_sách_các_cột>  
FROM <danh_sách_các_bảng>  
WHERE <điều_kiện>  
GROUP BY <danh_sách_các_cột_gom_nhóm>
```

## Mệnh đề GROUP BY

- Ví dụ: Cho biết số lượng nhân viên của từng phòng ban

```
SELECT  
    PB.TenPB,  
    count(*) AS SL_NV  
FROM NHANVIEN NV, PHONGBAN PB  
WHERE NV.Phong = PB.MaPB  
GROUP BY PB.TenPB
```

	TenPB	SL_NV
1	Giam doc	1
2	Hanh chinh	3
3	Nghien cuu	4

## Mệnh đề GROUP BY

Ví dụ: Với mỗi nhân viên, cho biết mã nhân viên, số lượng dự án và tổng thời số giờ tham gia dự án.

```
SELECT
    NV.MaNV,
    NV.Ten,
    Count(PC.MaDA) AS SL_DA,
    SUM(PC.SoGio) AS TONG_SOGIO
FROM PHANCONG PC, NHANVIEN NV
WHERE PC.MaNV = NV.MaNV
GROUP BY NV.MaNV, NV.Ten
```

	MaNV	Ten	SL_DA	TONG_SOGIO
1	123456789	Hung	1	7.5
2	333445555	Nghia	4	40.0
3	453453453	Yen	2	40.0
4	666884444	Nam	1	40.0
5	888665555	Bo	1	NULL
6	987654321	Hau	2	35.0
7	987987987	Giap	2	40.0
8	999887777	Vuong	2	40.0

Lấy ra danh sách nhân viên tham gia > 2 dự án?

# Điều kiện trên NHÓM - HAVING

- 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>
```

## Điều kiện trên NHÓM - HAVING

- Ví dụ: Lấy ra danh sách nhân viên tham gia ít nhất 2 dự án.

```
SELECT
    NV.MaNV,
    NV.Ten,
    Count(PC.MaDA) AS SL_DA,
    SUM(PC.SoGio) AS TONG_SOGIO
FROM PHANCONG PC, NHANVIEN NV
WHERE PC.MaNV = NV.MaNV
GROUP BY NV.MaNV, NV.Ten
HAVING Count(PC.MaDA) >= 2
```

	MaNV	Ten	SL_DA	TONG_SOGIO
1	333445555	Nghia	4	40.0
2	453453453	Yen	2	40.0
3	987654321	Hau	2	35.0
4	987987987	Giap	2	40.0
5	999887777	Vuong	2	40.0

## Điều kiện trên NHÓM - HAVING

- Ví dụ: Cho biết các phòng ban có lương trung bình của các nhân viên lớn hơn 33000.

```
SELECT
    PB.TenPB,
    AVG(LUONG) AS AVG_LUONG
FROM NHANVIEN NV, PHONGBAN PB
WHERE NV.Phong = PB.MaPB
GROUP BY PB.TenPB
HAVING AVG(LUONG) > 33000
```

	TenPB	AVG_LUONG
1	Giám đốc	55000
2	Nghiên cứu	33250



# Mệnh đề **GROUP BY & HAVING**

- **Mệnh đề GROUP BY**

- Các thuộc tính trong mệnh đề SELECT (trừ những thuộc tính trong các hàm tổng 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 dòng.
- Sau khi gom nhóm điều kiện trên nhóm mới được thực hiện

## Mệnh đề **GROUP BY & HAVING**

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

# Truy vấn sử dụng phép kết - JOIN

Cú pháp:

Kết bằng:

```
SELECT <danh_sách_các_cột>  
FROM R1 [INNER] JOIN R2 ON <biểu_thức>  
WHERE <điều_kiện>
```

Kết trái/ Kết phải:

```
SELECT <danh_sách_các_cột>  
FROM R1 LEFT | RIGHT [OUTER] JOIN R2 ON <biểu_thức>  
WHERE <điều_kiện>
```

# Truy vấn sử dụng phép kết - JOIN

- Ví dụ: Tìm danh sách nhân viên làm việc ở phòng 'Nghien cuu'

```
SELECT
  NV.*
FROM NHANVIEN NV, PHONGBAN PB
WHERE NV.Phong = PB.MaPB
AND PB.TenPB = 'Nghien cuu';
```

```
SELECT
  NV.*
FROM NHANVIEN NV INNER JOIN PHONGBAN PB
ON NV.Phong = PB.MaPB
WHERE PB.TenPB = 'Nghien cuu';
```

# Truy vấn sử dụng phép kết - JOIN

- Ví dụ: Tìm danh sách nhân viên không có bất kỳ thân nhân nào

```
SELECT
NV.*
FROM NHANVIEN NV
WHERE NOT EXISTS
(
    SELECT
    *
    FROM THANNHAN TN
    WHERE NV.MaNV = TN.MaNV
)
```

```
SELECT
    NV.*
FROM NHANVIEN NV
LEFT JOIN THANNHAN TN
ON NV.MaNV = TN.MaNV
WHERE TN.MaNV IS NULL
```

# Cấu trúc CASE WHEN

- 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 WHEN

- Ví dụ: Cho biết họ tên nhân viên đã đến tuổi về hưu (nam 60 tuổi, nữ 55 tuổi)

SELECT

```
NV.Ho + ' ' + NV.TenDem + ' ' + NV.Ten AS HoTen,  
NV.NgSinh,  
NV.GTinh,  
YEAR(GETDATE()) - YEAR(NV.NgSinh) AS AGE
```

FROM NHANVIEN NV

```
WHERE YEAR(GETDATE()) - YEAR(NV.NgSinh) >= (  
    WHEN 'Nam' THEN 60  
    WHEN 'Nu' THEN 55  
    END  
)
```

# KẾT LUẬN

**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>]