

**Chương 4-1:**

## **Transact-SQL và truy vấn dữ liệu**

**Phạm Thị Thu Thúy**

Khoa CNTT, Đại học Nha Trang

[thuthuy@ntu.edu.vn](mailto:thuthuy@ntu.edu.vn)

# Nội dung chương 4

---

- **Transact SQL**
  - Câu lệnh thao tác dữ liệu
    - Truy vấn dữ liệu cơ bản
    - Truy vấn lồng
    - Hàm kết hợp và gom nhóm
    - Một số dạng truy vấn khác
  - Khung nhìn
-

# Transact-SQL Language -1

- Structured Query Language (SQL) là một ngôn ngữ rất phổ dụng trong lĩnh vực CSDL.
- Microsoft xây dựng Transact-SQL dựa trên ngôn ngữ vấn tin có cấu trúc chuẩn.
- Nó cung cấp một ngôn ngữ bao hàm toàn diện để định nghĩa bảng, chèn, xóa, thay đổi và truy cập dữ liệu trong bảng.
- Transact-SQL là một ngôn ngữ mạnh, nó hỗ trợ các tính năng khác như: kiểu dữ liệu, đối tượng tạm thời, thủ tục lưu trữ và thủ tục hệ thống.
- Nó còn cho phép chúng ta định nghĩa đối tượng con trỏ, khai báo biến, cấu trúc rẽ nhánh, vòng lặp, bắt lỗi và Transaction.

# Transact-SQL Language -2

- Example for Transact-SQL statement:

```
SELECT * FROM Employee
```

- The statement retrieves all records from Employee table.

# Variables in Transact-SQL - 1

- Biến là vùng nhớ trong bộ nhớ được đặt tên để chứa giá trị dữ liệu. Dữ liệu có thể truyền cho câu lệnh SQL bằng biến cục bộ. Tên biến cục bộ phải khai báo bắt đầu bằng ký hiệu @.
- Biến có thể phân thành 2 loại:
  - Local Variables (Biến cục bộ)
  - Global Variables (Biến toàn cục)

# Variables in Transact-SQL - 2

- Local Variables

- Trong Transact-SQL, biến cục bộ được khai báo và sử dụng tạm thời khi thực thi câu lệnh SQL.

Cú pháp:

```
DECLARE
{
@local_variable [AS] data_type
}
```

where,

@local\_variable specifies the name of a variable.

Variable names must begin with the '@'.

data\_type is a system or user-defined data type.

# Variables in Transact-SQL - 3

- Câu lệnh SET hoặc SELECT dùng để gán giá trị cho biến.

Cú pháp:

```
SET @local_variable = value  
Hoặc  
SELECT @local_variable = value
```

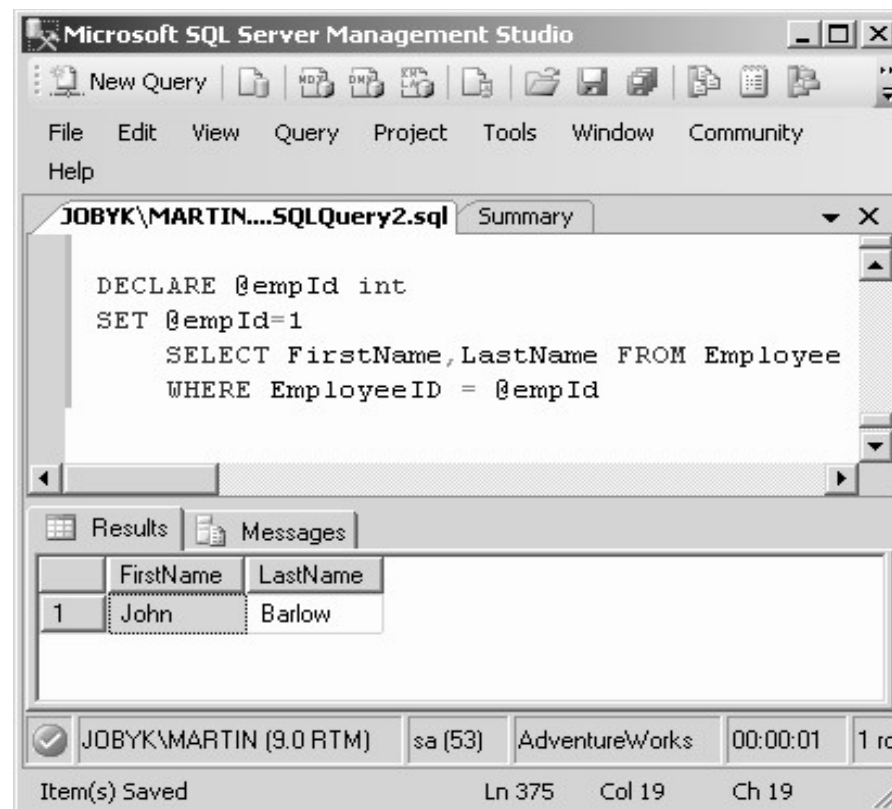
# Variables in Transact-SQL - 4

Ví dụ: DECLARE @empID int

SET @empID = 1

SELECT FirstName, LastName FROM Employee

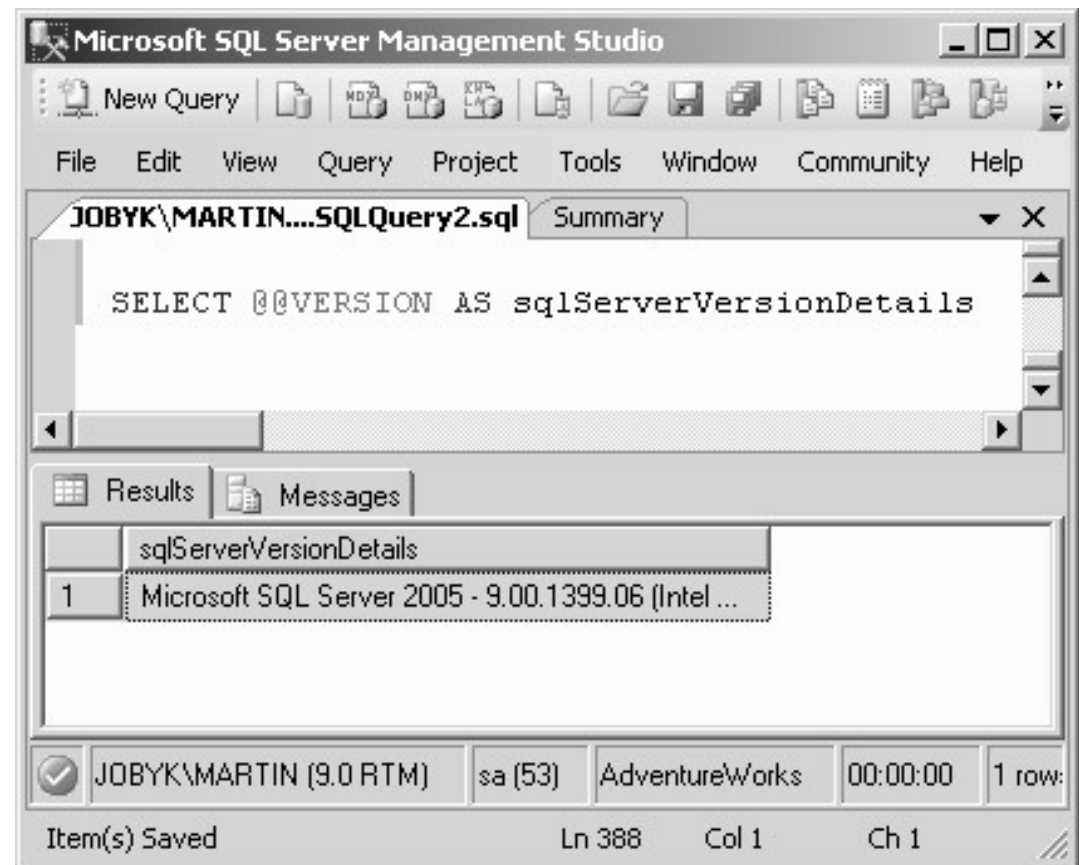
WHERE EmployeeID = @empID





# Variables in Transact-SQL - 5

- Global Variables
  - Biến toàn cục là biến có sẵn và hệ thống quản lý.
  - Biến toàn cục trong SQL Server được đặt tên bắt đầu bởi hai ký hiệu @.
- Ví dụ:
  - `SELECT @@VERSION AS 'SQL Server version'`



# Data Types in Transact-SQL

- Kiểu dữ liệu là thuộc tính định nghĩa loại dữ liệu mà đối tượng có thể chứa.
- Transact-SQL bao gồm nhiều kiểu dữ liệu chẳng hạn như: varchar, text, int,....
- Các đối tượng dưới đây sử dụng kiểu dữ liệu:
  - **Cột trong bảng**
  - **Biến**
  - Tham số trong thủ tục lưu trữ
  - Hàm trả về giá trị
  - Thủ tục lưu trữ trả về giá trị

# Comments in Transact-SQL - 1

- Microsoft SQL Server hỗ trợ hai loại chú thích sau:
  - -- chú thích một dòng
  - /\* . . . \*/ chú thích nhiều dòng

# Data Definition Language

- Data Definition Language (DDL), được dùng để định nghĩa (xây dựng), thay đổi hoặc xóa cấu trúc của các đối tượng CSDL, chẳng hạn: bảng, view, trigger, thủ tục lưu trữ, ...
- Câu lệnh DDL:
  - CREATE object\_name
  - ALTER object\_name
  - DROP object\_name

Với object\_name: TABLE, VIEW, ...

# Data Manipulation Language

- Data manipulation language is used to select, insert, update, and delete data in the objects defined with DDL.
- There are three types of Data Manipulation Languages:
  - SELECT statement
  - INSERT statement
  - UPDATE statement
  - DELETE statement

# Nội dung chương 4

---

- Transact SQL
  - **Câu lệnh thao tác dữ liệu**
    - Truy vấn dữ liệu cơ bản
    - Truy vấn lồng
    - Hàm kết hợp và gom nhóm
    - Một số dạng truy vấn khác
  - Khung nhìn
-

## Cơ sở dữ liệu ví dụ

---

- Trong suốt phần sau của chương này, ta sử dụng CSDL sau làm ví dụ minh họa:
    - NhanVien(manv, honv, tenlot, tennv, ngaysinh, diachi, phai, luong, manql, phong)
    - PhongBan(maphong, tenphong, trphong, ngnhanchuc)
    - DDPhong(maphong, diadiem)
    - DeAn(tenda, mada, ddiemda, phong)
    - PhanCong(manvien, soda, thoigian)
    - ThanNhan(manvien, tentn, phai, ngaysinh, quanhe)
-

# Truy vấn dữ liệu

---

- Là ngôn ngữ rút trích dữ liệu
  - Thường đi kèm với một số điều kiện nào đó
- Dựa trên

$$\boxed{\text{Phép toán ĐSQH}} + \boxed{\text{Một số bổ sung}}$$

  - Cho phép kết quả trả về của bảng có nhiều dòng trùng nhau



# Truy vấn cơ bản

- Gồm 3 mệnh đề

**SELECT** [~~**DISTINCT**~~] <danh sách các cột>  
**FROM** <danh sách các bảng>  
**[WHERE]** <điều kiện>

Từ khóa **DISTINCT** để loại bỏ các dòng trùng nhau trong bảng kết quả

- <danh sách các cột>:

- Tên các cột cần được hiển thị trong kết quả truy vấn

- <danh sách các bảng>

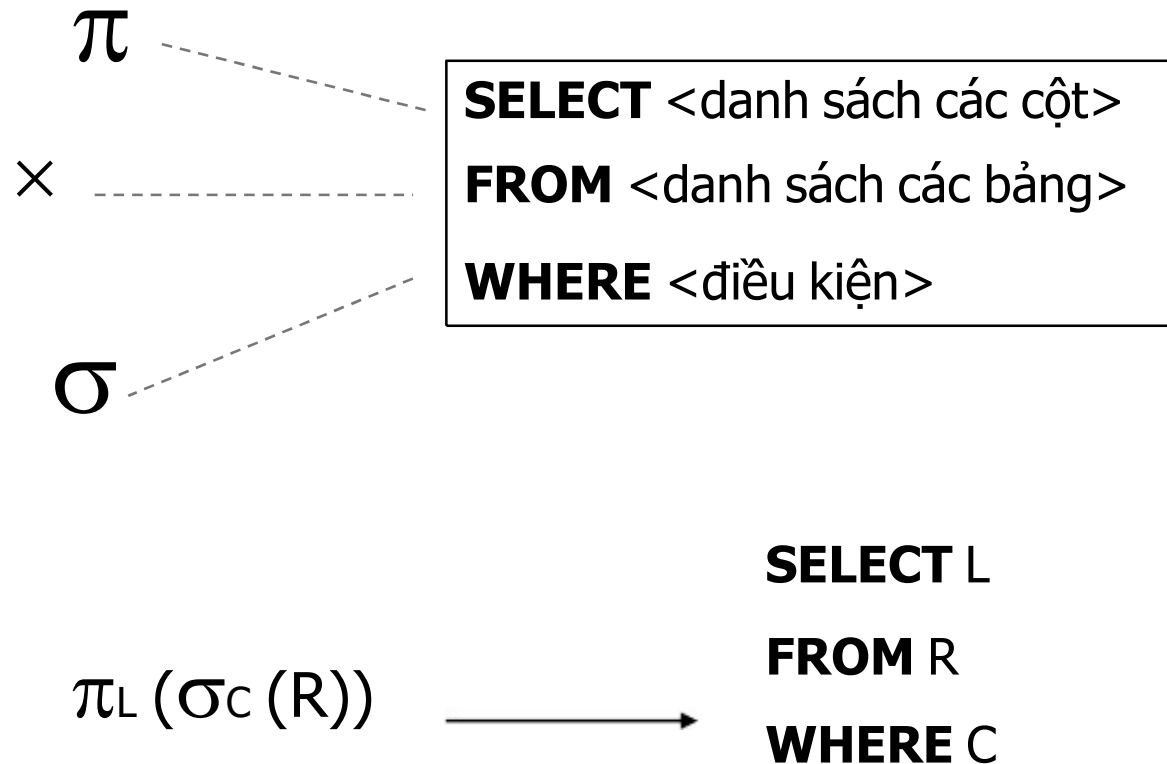
- Tên các bảng liên quan đến câu truy vấn

- <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 (tt)

- SQL và ĐSQH



## Ví dụ

Lấy tất cả các cột của  
quan hệ kết quả

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

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
333445555	Nguyen	Thanh	Tung	12/08/1955	638 NVC Q5	Nam	40000	888665555	5
987987987	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	333445555	5

# Mệnh đề SELECT

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

MANV	HONV	TENLOT	TENNV
33344555	Nguyen	Thanh	Tung
5 98798798	Nguyen	Manh	Hung
7			

# Mệnh đề SELECT (tt)

Tên bí danh

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

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

# Mệnh đề SELECT (tt)

Mở rộng

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

MANV	HO TEN
333445555	Nguyen Thanh Tung
987987987	Nguyen Manh Hung

# Mệnh đề SELECT (tt)

Mở rộng

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

MANV	LUONG10%
333445555	33000
987987987	27500

# Mệnh đề SELECT (tt)

Loại bỏ các dòng trùng nhau

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

LUONG

30000

25000

38000

38000

- Tổn chi phí

- Người dùng muốn thấy



## Ví dụ

- Cho biết MANV và TENNV làm việc ở phòng ‘Nghien cuu’

Select manv, tennv

from nhanvien, phongban

where (nhanvien.mphg = phongban.mphg) and  
(phongban.tenphong = “nghien cuu”)

# Mệnh đề WHERE

SELECT MANV, TENNV  
FROM NHANVIEN, PHONGBAN  
WHERE TENPHG='Nghien cuu' AND PHG=MAPHG

Biểu thức luận lý

TRUE TRUE

## Mệnh đề WHERE (tt)

Độ ưu tiên
------------

```
SELECT MANV, TENNV  
FROM NHANVIEN, PHONGBAN  
WHERE (TENPHG='Nghien cuu' OR TENPHG='Quan ly')  
AND PHG=MAPHG
```

## Mệnh đề WHERE (tt)

**BETWEEN**

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

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

## Mệnh đề WHERE (tt)

NOT BETWEEN

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

# Mệnh đề WHERE (tt)

LIKE

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE 'Nguyễn _ _ _ _ '
```

Ký tự bất kỳ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE 'Nguyễn %'
```

Chuỗi bất kỳ

# Mệnh đề WHERE (tt)

NOT LIKE

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HONV LIKE 'Nguyễn'
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE HONV NOT LIKE 'Nguyễn'
```

# Mệnh đề WHERE (tt)

**ESCAPE**

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE '% Nguyens_%' ESCAPE 's'
```



'Nguyen\_'



## Mệnh đề WHERE (tt)

Ngày giờ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE NGSINH 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
  - $x + 3$  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

## Mệnh đề WHERE (tt)

NULL

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

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

# Mệnh đề FROM

Không sử dụng mệnh đề WHERE

```
SELECT MANV, MAPHG  
FROM NHANVIEN, PHONGBAN
```

```
WHERE TRUE
```

MANV	MAPHG
333445555	1
333445555	4
333445555	5
987987987	1
987987987	4
987987987	5
...	...

## Mệnh đề FROM (tt)

Tên bí danh

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

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

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

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

## Ví dụ 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

Select a.soda, a.mphg, b.tennv, b.ngsinh, b.diachi

From dean a, nhanvien b, phongban c

Where (a.mphb = b.mphg) and (b.manv = c.matrphong)

## Ví dụ 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

Select a.ho, a.lot, a.ten

From nhanvien a, dean b, phancong c

Where (a.manv = c.manv) and (b.soda = c.soda)

and tendean=“San pham X” and (a.maphg=5) and  
(c.thoigian>10)

## Ví dụ 3

- Tìm họ tên của từng nhân viên và người phụ trách trực tiếp nhân viên đó

Select a.tennv, b.tennv

From nhanvien a, nhanvien b

Where a.manql = b.manv



## Ví dụ 4

- Tìm họ tên của những nhân viên được “Nguyen Thanh Tung” phụ trách trực tiếp

Select a.tennv

From nhanvien a, nhanvien b

Where (b.tenv=“Nguyen Thanh Tung”) and  
(a.manql=b.manv)

## 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 <ds các cột>  
FROM <ds các bảng>  
WHERE <điều kiện>  
  
ORDER BY <danh sách các cột><kiểu sắp>
```

  - Trong đó <kiểu sắp> có thể là:
    - ASC: tăng (mặc định)
    - DESC: giảm

## Truy vấn cơ bản

### - Mệnh đề ORDER BY (tt)

- Ví dụ:

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

MANVIEN	SODA
999887777	10
999887777	30
987987987	10
987987987	30
987654321	10
987654321	20
987654321	30

## Nội dung chương IV

---

- Câu lệnh thao tác dữ liệu
  - Truy vấn dữ liệu cơ bản
  - **Tập hợp, so sánh tập hợp, Truy vấn lồng**
  - Hàm kết hợp và gom nhóm
  - Một số dạng truy vấn khác
- Khung nhìn

# Phép toán tập hợp trong SQL

- 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>
```

## Ví dụ 5

- Cho biết các mã đề án có
  - Nhân viên với họ là ‘Nguyen’ tham gia hoặc,
  - Trưởng phòng chủ trì đề án đó với họ là ‘Nguyen’

Select b.soda

From nhanvien a , dean b

Where (a.manv = b.manv) and (a.ho=“Nguyen”)

union

(Select b.soda

From phongban a, dean b, nhanvien c

Where (b.maphg=a.maphg) and (c.manv=a.matrphong)  
and (c.ho=“Nguyen”))

## Ví dụ 6

- Tìm nhân viên có người thân cùng tên và cùng giới tính



## Ví dụ 7

- Tìm những nhân viên không có thân nhân nào

Select manv  
From nhanvien

Except

Select manv  
From thanhan

---

Select tennv  
From nhanvien  
Where manv **not in** (select manv from thannhan)

## Truy vấn lồng

- Các câu 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
- Cú pháp:

```
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>)
```

## Truy vấn lồng (tt)

---

- 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 (tt)

---

- 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, 1 lần
  - 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 (tt)

### - Ví dụ lồng phân cấp

---

- Ví dụ:

```
SELECT MANV, TENNV
FROM NHANVIEN, DDPhong
WHERE DIADIEM = 'TP HCM' AND phong = maphong
```

- Tương đương với câu truy vấn lồng sau:

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE PHONG IN (SELECT MAPHONG
                FROM DDPHONG
                WHERE DIADIEM = 'TP HCM'
                )
```

## Truy vấn lòng (tt)

### - Ví dụ lòng phân cấp

---

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

## Truy vấn lồng (tt)

### - Ví dụ lồng tương quan

---

- Ví dụ:

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE EXISTS (
    SELECT *
    FROM PHONGBAN
    WHERE TENPHONG='Nghien cuu' AND PHONG=MAPHONG)
```

→ Câu truy vấn trên cho kết quả gì? Tương đương với câu truy vấn đơn giản nào?

```
SELECT MANV, TENNV
FROM NHANVIEN, PHONGBAN
WHERE TENPHONG='Nghien cuu' AND PHONG=MAPHONG
```

## Truy vấn lồng (tt)

### - Ví dụ

---

- Tìm nhân viên có người thân cùng tên và cùng giới tính với nhân viên đó



## Truy vấn lồng (tt)

### - Ví dụ

---

- Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên phòng 4
- Tìm những nhân viên có lương lớn hơn lương của tất cả nhân viên phòng 4

# Truy vấn lồng (tt)

## - Nhận xét

- 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

# 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

## Phép chia trong SQL (tt)

- 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 ))
```

## Ví dụ

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

## Ví dụ (tt)

- Tìm tên các nhân viên được phân công làm tất cả các đồ án

Select 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.SoDa=da.MaDa **and**  
pc1.MaNV=pc2.MaNV))

# Nội dung chương IV

---

- Câu lệnh thao tác dữ liệu
  - Truy vấn dữ liệu cơ bản
  - Truy vấn lồng
  - **Hàm kết hợp và gom nhóm**
  - Một số dạng truy vấn khác
- Khung nhìn

# Hàm kết hợp và gom nhóm

---

- Hàm kết hợp:
  - Được sử dụng trong mệnh đề SELECT
  - Có các hàm kết hợp sau:
    - 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



## Hàm kết hợp và gom nhóm (tt)

---

- Ví dụ:
  - Tìm tổng lương, lương cao nhất, lương thấp nhất và lương trung bình của các nhân viên
  - ```
Select    sum(luong)    as    'tong    luong' ,  
min(luong)    as    'luong    nho    nhat' ,  
max(luong)    as    'luong    lon    nhat' ,  
avg(luong) as 'luong tb'  
From NhanVien
```
  - Cho biết số lượng nhân viên của phòng "Nghien cuu"

## Hàm kết hợp và gom nhóm (tt)

2. Có tất cả bao nhiêu nhân viên

```
SELECT count(*) FROM NhanVien
```

3. Bao nhiêu nhân viên có người quản lý

```
Select count(*) FROM NhanVien WHERE manql is not null
```

```
SELECT count(Manql) FROM NhanVien
```

4. Bao nhiêu phòng ban có nhân viên trực thuộc

```
SELECT count(distinct phong) FROM NhanVien
```

## Hàm kết hợp và gom nhóm (tt)

5. Tính lương trung bình của các nhân viên

```
SELECT avg(LUONG) as LUONGTB  
FROM NhanVien
```

6. Tính lương trung bình của các nhân viên theo từng phòng ban.

```
SELECT phong, avg(LUONG) as LUONGTB  
FROM NhanVien  
GROUP BY phong
```

## Hàm kết hợp và gom nhóm (tt)

---

- Gom nhóm:
  - Nhóm các bộ có cùng giá trị ở một tập các thuộc tính lại với nhau
  - Cú pháp:

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

## Hàm kết hợp và gom nhóm (tt)

---

- Ví dụ:
  - Cho biết số lượng nhân viên của từng phòng ban
  - Cho biết thông tin của những phòng ban có nhiều hơn 10 nhân viên

## Hàm kết hợp và gom nhóm (tt)

- Ví dụ:
  - Với mỗi nhân viên cho biết mã số, họ tên, số lượng đề án và tổng thời gian mà họ tham gia

```
SELECT      MANV, HONV, TENNV,  
            COUNT(*) AS SL_DA,  
            SUM(THOIGIAN) AS TONG_TG  
FROM        PHANCONG, NHANVIEN  
WHERE       MANVIEN=MANV  
GROUP BY    MANV, HONV, TENNV
```

- Cho biết những nhân viên tham gia từ 2 đề án trở lên

## Hàm kết hợp và gom nhóm (tt)

- 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: tìm phòng có số lượng nhân viên “Nữ” trên 5 người

|          |                 |
|----------|-----------------|
| SELECT   | phong           |
| FROM     | NhanVien        |
| WHERE    | phai = 'Nữ'     |
| GROUP BY | phong           |
| HAVING   | count(manv) > 5 |

## Hàm kết hợp và gom nhóm (tt)

---

- Ví dụ:
  - Cho biết những phòng ban (TENPHONG) có lương trung bình của các nhân viên lớn hơn 20000



# Hàm kết hợp và gom nhóm (tt)

## - Nhận xét

- 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

# Hàm kết hợp và gom nhóm (tt)

## - Nhận xét

- 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

## Hàm kết hợp và gom nhóm (tt)

---

- Ví dụ:
  - Tìm phòng ban có lương trung bình cao nhất
  - Tìm 3 nhân viên có lương cao nhất
  - Tìm tên các nhân viên được phân công làm tất cả các đồ án

## Nội dung chương IV

---

- Câu lệnh thao tác dữ liệu
  - Truy vấn dữ liệu cơ bản
  - Truy vấn lồng
  - Hàm kết hợp và gom nhóm
  - **Một số dạng truy vấn khác**
- Khung nhìn

## Điều kiện kết ở mệnh đề FROM

- Thông thường điều kiện của phép kết nối được chỉ định trong mệnh đề where của câu truy vấn.
- Chuẩn SQL2 đưa ra một cách khác để biểu diễn cho phép nối. Trong cách biểu diễn này:
  - Điều kiện của phép kết nối được chỉ định ngay trong mệnh đề From.
  - Ưu điểm: Cho phép biểu diễn phép nối và điều kiện nối rõ ràng (đặc biệt trong trường hợp phép nối được thực hiện từ 3 bảng trở lên)

## Điều kiện kết ở mệnh đề FROM (tt)

- Kết nối bằng (kết nối trong)

– Cú pháp:

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

## Điều kiện kết ở mệnh đề FROM (tt)

- Kết nối bằng (kết nối trong)
  - Ví dụ: Hiển thị họ tên đầy đủ, ngày sinh, địa chỉ của các nhân viên phòng “Nghien cuu”.

*Có thể viết theo 2 cách như sau:*

```
select    honv, tenlot, tennv, ngaysinh, diachi
From      NhanVien, PhongBan
Where     phong = maphong
          and tenphong = 'Nghien cuu'
```

- Hoặc có thể viết:

```
select    honv, tenlot, tennv, ngaysinh, diachi
From      NhanVien INNER JOIN PhongBan ON
          phong = maphong
Where     tenphong = 'Nghien cuu'
```

## Điều kiện kết ở mệnh đề FROM (tt)

---

### Nhận xét

- Kết nối trong trả về những dòng có giá trị trong các cột được chỉ định thỏa mãn điều kiện kết nối, *không trích xuất các dòng không thỏa mãn.*
- Để hiển thị cả những thông tin thỏa mãn và không thỏa mãn điều kiện nối ta có thể sử dụng phép nối ngoài. (*những thông tin không thỏa mãn có giá trị NULL*)



## Điều kiện kết ở mệnh đề FROM (tt)

- Kết nối ngoài

- Cú pháp:

**SELECT** <danh sách các cột>

**FROM** R1 **LEFT** | **RIGHT** | **FULL** [**OUTER**] **JOIN** R2

**ON** <biểu thức>

**WHERE** <điều kiện>

## Điều kiện kết ở mệnh đề FROM (tt)

- Kết nối ngoài
  - Ví dụ: Ta có 2 bảng PhongBan và DDPhong như sau:

HAM-PC.QLD... - dbo.PhongBan × QLDuAn\_TruyVan.s...C\Thuy Phan

| maphong | tenphong   | trphong   | ngnhanchuc |
|---------|------------|-----------|------------|
| 1       | Nghien cuu | 987698765 | 2012-03-02 |
| 2       | Kinh doanh | 987987987 | 2010-03-12 |
| 3       | San xuat   | 987987987 | 2009-12-04 |
| 4       | Ke toan    | 999887777 | 2010-02-13 |
| 5       | Hanh chinh | 999887777 | 2011-02-03 |

PHAM-PC.QLD...n - dbo.DDPhong

| maphong | diadiem   |
|---------|-----------|
| 1       | Ha noi    |
| 1       | TPHCM     |
| 2       | TPHCM     |
| 3       | Nha Trang |
| 3       | TPHCM     |
| 4       | Nha Trang |

- Kết nối ngoài trái 2 bảng trên cho kết quả sau:

```
select *  
from PhongBan left join  
    DDPhong on  
PhongBan.maphong =  
DDPhong.maphong
```

| maphong | tenphong   | trphong   | ngnhanchuc | maphong | diadiem   |
|---------|------------|-----------|------------|---------|-----------|
| 1       | Nghien cuu | 987698765 | 2012-03-02 | 1       | Ha noi    |
| 1       | Nghien cuu | 987698765 | 2012-03-02 | 1       | TPHCM     |
| 2       | Kinh doanh | 987987987 | 2010-03-12 | 2       | TPHCM     |
| 3       | San xuat   | 987987987 | 2009-12-04 | 3       | Nha Trang |
| 3       | San xuat   | 987987987 | 2009-12-04 | 3       | TPHCM     |
| 4       | Ke toan    | 999887777 | 2010-02-13 | 4       | Nha Trang |
| 5       | Hanh chinh | 999887777 | 2011-02-03 | NULL    | NULL      |

## Điều kiện kết ở mệnh đề FROM (tt)

- Kết nối ngoài
  - Ví dụ: Ta có 2 bảng PhongBan và DDPhong như sau:

HAM-PC.QLD... - dbo.PhongBan X QLDuAn\_TruyVan.s...C\Thuy Phan

| maphong | tenphong   | trphong   | ngnhanchuc |
|---------|------------|-----------|------------|
| 1       | Nghien cuu | 987698765 | 2012-03-02 |
| 2       | Kinh doanh | 987987987 | 2010-03-12 |
| 3       | San xuat   | 987987987 | 2009-12-04 |
| 4       | Ke toan    | 999887777 | 2010-02-13 |
| 5       | Hanh chinh | 999887777 | 2011-02-03 |

PHAM-PC.QLD...n - dbo.DDPhong

| maphong | diadiem   |
|---------|-----------|
| 1       | Ha noi    |
| 1       | TPHCM     |
| 2       | TPHCM     |
| 3       | Nha Trang |
| 3       | TPHCM     |
| 4       | Nha Trang |

- Kết nối ngoài phải 2 bảng trên cho kết quả sau:

```
select *  
from PhongBan right join  
    DDPhong on  
PhongBan.maphong =  
DDPhong.maphong
```

| maphong | tenphong   | trphong   | ngnhanchuc | maphong | diadiem   |
|---------|------------|-----------|------------|---------|-----------|
| 1       | Nghien cuu | 987698765 | 2012-03-02 | 1       | Ha noi    |
| 1       | Nghien cuu | 987698765 | 2012-03-02 | 1       | TPHCM     |
| 2       | Kinh doanh | 987987987 | 2010-03-12 | 2       | TPHCM     |
| 3       | San xuat   | 987987987 | 2009-12-04 | 3       | Nha Trang |
| 3       | San xuat   | 987987987 | 2009-12-04 | 3       | TPHCM     |
| 4       | Ke toan    | 999887777 | 2010-02-13 | 4       | Nha Trang |

## Điều kiện kết ở mệnh đề FROM (tt)

- Kết nối ngoài
  - Ví dụ: Ta có 2 bảng PhongBan và DDPhong như sau:

HAM-PC.QLD... - dbo.PhongBan X QLDuAn\_TruyVan.s...C\Thuy Phan

| maphong | tenphong   | trphong   | ngnhanchuc |
|---------|------------|-----------|------------|
| 1       | Nghien cuu | 987698765 | 2012-03-02 |
| 2       | Kinh doanh | 987987987 | 2010-03-12 |
| 3       | San xuat   | 987987987 | 2009-12-04 |
| 4       | Ke toan    | 999887777 | 2010-02-13 |
| 5       | Hanh chinh | 999887777 | 2011-02-03 |

PHAM-PC.QLD...n - dbo.DDPhong

| maphong | diadiem   |
|---------|-----------|
| 1       | Ha noi    |
| 1       | TPHCM     |
| 2       | TPHCM     |
| 3       | Nha Trang |
| 3       | TPHCM     |
| 4       | Nha Trang |

- Kết nối ngoài đầy đủ 2 bảng trên cho kết quả sau:

```
select *  
from PhongBan full join  
    DDPhong on  
PhongBan.maphong =  
DDPhong.maphong
```

| maphong | tenphong   | trphong   | ngnhanchuc | maphong | diadiem   |
|---------|------------|-----------|------------|---------|-----------|
| 1       | Nghien cuu | 987698765 | 2012-03-02 | 1       | Ha noi    |
| 1       | Nghien cuu | 987698765 | 2012-03-02 | 1       | TPHCM     |
| 2       | Kinh doanh | 987987987 | 2010-03-12 | 2       | TPHCM     |
| 3       | San xuat   | 987987987 | 2009-12-04 | 3       | Nha Trang |
| 3       | San xuat   | 987987987 | 2009-12-04 | 3       | TPHCM     |
| 4       | Ke toan    | 999887777 | 2010-02-13 | 4       | Nha Trang |
| 5       | Hanh chinh | 999887777 | 2011-02-03 | NULL    | NULL      |

# CASE

- Đánh giá danh sách các điều kiện và trả về một trong nhiều biểu thức kết quả có thể có.
- CASE có thể được sử dụng trong bất kỳ câu lệnh hoặc mệnh đề nào cho phép một biểu thức hợp lệ.
- Ví dụ: Có thể sử dụng CASE trong các câu lệnh như SELECT, UPDATE, DELETE và SET, và trong các mệnh đề như `select_list`, IN, WHERE, ORDER BY và HAVING.

## 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
```

# Case – Example 1

UPDATE CLASS

set TimeSlot =

CASE

when right(ClassCode,1)='G' then '7:30 - 9:30'

when right(ClassCode,1)='H' then '9:30 - 11:30'

when right(ClassCode,1)='I' then '13:30 - 15:30'

when right(ClassCode,1)='K' then '15:30 - 17:30'

when right(ClassCode,1)='L' then '17:30 - 19:30'

when right(ClassCode,1)='M' then '19:30 - 21:30'

ELSE 'unknown'

END

## Case – Example 2

```
SELECT ProductNumber, Name, 'Price Range' =  
    CASE  
        WHEN ListPrice = 0 THEN 'not for resale'  
        WHEN ListPrice < 50 THEN 'Under $50'  
        WHEN ListPrice >= 50 and ListPrice < 250 THEN 'Under $250'  
        WHEN ListPrice >= 250 and ListPrice < 1000 THEN 'Under $1000'  
        ELSE 'Over $1000'  
    END  
FROM Product  
ORDER BY ProductNumber ;
```



## Case – Example 3

update books set [type]=

case

when Totalpage<100 then 'thin'

when Totalpage between 100 and 1000 then  
'normal'

when TotalPage>1000 then 'thick'

end

## Ví dụ

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

## Ví dụ

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

# Câu lệnh SELECT tổng quát

**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>

# Hàm trong Transact-SQL 1-8

- Hàm là tập lệnh Transact-SQL để thực hiện công việc nào đó
- Hàm trong SQL làm việc với dữ liệu, nhóm dữ liệu để trả về một kết quả mong đợi
- Transact-SQL có các loại hàm sau:
  - Các hàm tập hợp
  - Hàm chuyển đổi kiểu dữ liệu
  - Hàm ngày tháng
  - Hàm toán học
  - Các hàm hệ thống
  - Hàm xếp hạng

# Hàm trong Transact-SQL 2-8

- Các hàm tập hợp: để thống kê dữ liệu .

| Tên hàm              | Giá trị trả về                                  | Ví dụ                                                                                                |
|----------------------|-------------------------------------------------|------------------------------------------------------------------------------------------------------|
| <b>SUM(col_name)</b> | Hàm tính tổng, trả về tổng giá trị của col_name | SELECT SUM(OrderQty) AS Total FROM Purchasing.PurchaseOrderDetail                                    |
| <b>AVG(col_name)</b> | Trả tính giá trị trung bình                     | SELECT AVG(UnitPrice * OrderQty) AS AveragePrice FROM Purchasing.PurchaseOrderDetail                 |
| <b>COUNT</b>         | Đếm số bản ghi trong bảng                       | SELECT COUNT(*) AS 'Number of Large Orders' FROM Purchasing.PurchaseOrderDetail WHERE OrderQty > 100 |
| <b>MAX(col_name)</b> | Trả về giá trị lớn nhất                         | SELECT MAX(OrderQty * UnitPrice) AS 'Largest Order' FROM Purchasing.PurchaseOrderDetail              |
| <b>MIN(col_name)</b> | Trả về giá trị nhỏ nhất                         | SELECT MIN(OrderQty * UnitPrice) AS 'Smallest Order' FROM Purchasing.PurchaseOrderDetail             |

# Hàm trong Transact-SQL 3-8

## – Hàm chuyển đổi kiểu dữ liệu

- Hàm chuyển đổi được sử dụng để biến đổi giá trị của một kiểu dữ liệu sang kiểu dữ liệu khác

Cú pháp:

```
CONVERT(datatype[(length)], expression [,style])
```

Trong đó,

*datatype là kiểu dữ liệu cần chuyển sang*

*length chỉ ra độ dài của dữ liệu*

*expression chỉ ra tên cột, hằng số, hàm, biến hoặc là câu truy vấn con*

*style chỉ ra kiểu hiển thị giá trị ngày tháng(dd/mm/yyyy hay mm/dd/yyyy)*

# Hàm trong Transact-SQL 4-8

## – Các hàm ngày tháng

- Hàm ngày tháng được dùng để thao tác giá trị ngày tháng
- Hàm ngày tháng cho phép chúng ta lấy một phần của giá trị ngày tháng như: lấy ngày, tháng hoặc năm
- DATEPART: là một phần giá trị ngày tháng, được kết hợp sử dụng trong hàm ngày tháng để lấy một phần nào đó của giá trị ngày tháng



# Hàm trong Transact-SQL 4-8

Tất cả các DATEPART được hỗ trợ trong Transact-SQL :

| Datepart    | Abbreviation | Values    |
|-------------|--------------|-----------|
| Hour        | hh           | 0-23      |
| Minute      | Mi           | 0-59      |
| Second      | Ss           | 0-59      |
| Millisecond | Ms           | 0-999     |
| Day of Year | Dy           | 1-366     |
| Day         | Dd           | 1-31      |
| Week        | wk           | 1-53      |
| Weekday     | dw           | 1-7       |
| Month       | mm           | 1-12      |
| Quarter     | qq           | 1-4       |
| Year        | yy           | 1753-9999 |

# Hàm trong Transact-SQL 5-8

| Hàm                                     | Giá trị trả về                            | Ví dụ                                                                         |
|-----------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------|
| <b>GETDATE()</b>                        | Trả về ngày hiện tại của hệ thống         | SELECT GETDATE()                                                              |
| <b>DATEADD(datepart, number, date)</b>  | Cộng vào thêm number giá trị vào date     | SELECT DATEADD(mm,4,'01/01/99') - returns 05/01/99 in the current date format |
| <b>DATEDIFF(datepart, date1, date2)</b> | So sánh giữa hai giá trị ngày tháng       | SELECT DATEDIFF(mm,'01/01/99','05/01/99') - returns 4                         |
| <b>DATENAME(datepart, date)</b>         | Trả về giá trị ngày tháng dưới dạng chuỗi | SELECT DATENAME(dw,'01/01/2000') - returns Saturday                           |
| <b>DATEPART(datepart, date)</b>         | Trả về một phần giá trị ngày tháng        | SELECT DATEPART(day,'01/15/2000') - returns 15                                |

# Hàm trong Transact-SQL 6-8

**Thực hiện các phép tính đại số trên trường dữ liệu/giá trị số**

| Hàm                     | Các giá trị trả về                                                                  | Ví dụ                                    |
|-------------------------|-------------------------------------------------------------------------------------|------------------------------------------|
| ABS(num_expr)           | Trả về giá trị tuyệt đối                                                            | SELECT ABS(-43) return 43                |
| CEILING(num_expr)       | Trả về giá trị nhỏ nhất lớn hơn hoặc bằng num_expr                                  | SELECT CEILING(43.5)<br>returns 44       |
| FLOOR(num_expr)         | Trả về giá trị lớn nhất nhỏ hơn hoặc bằng num_expr                                  | SELECT FLOOR(43.5)<br>returns 43         |
| POWER(num_expr,y)       | Hàm lũy thừa                                                                        | SELECT POWER(5,2)<br>returns 25          |
| ROUND(num_expr,length ) | Hàm làm tròn                                                                        | SELECT ROUND(43.543,1)<br>returns 43.500 |
| SIGN(num_expr)          | Trả về +1 nếu num_expr là số dương, -1 nếu num_expr là số âm, 0 nếu num_expr bằng 0 | SELECT SIGN(-43) returns -1              |
| SQRT(float_expr)        | Hàm căn bậc hai                                                                     | SELECT SQRT(9) returns 3                 |

# Hàm trong Transact-SQL 7-8

- Hàm hệ thống: Trả về dữ liệu hệ thống hay thông tin cấu hình của SQL Server.

| Hàm                      | Giá trị trả về                                        |
|--------------------------|-------------------------------------------------------|
| DB_ID(['database_name']) | Trả về số định danh của cơ sở dữ liệu                 |
| DB_NAME([database_id])   | Trả về tên cơ sở dữ liệu                              |
| HOST_ID()                | Trả về số định danh cho máy trạm                      |
| HOST_NAME()              | Trả về tên máy trạm                                   |
| ISNULL(expr,value)       | Nếu expr là NULL thì được thay thế bằng giá trị VALUE |
| OBJECT_ID('obj_name')    | Số định danh đối tượng                                |
| OBJECT_NAME(object_id)   | Trả về tên đối tượng                                  |
| USER_ID(['user_name'])   | Số định danh người dùng                               |
| USER_NAME([user_id])     | Tên người dùng                                        |

# Hàm trong Transact-SQL 8-8

- Hàm xếp hạng: Tạo ra dãy số tuần tự, việc xếp hạng bây giờ dễ dàng thực hiện được trong SQL Server

| Hàm          | Giá trị trả về                                                                    |
|--------------|-----------------------------------------------------------------------------------|
| ROW_NUMBER() | Hàm đánh số hàng, bắt đầu bằng 1 với hàng đầu tiên và 2, 3 với các hàng tiếp theo |
| DENSE_RANK() | Trả về giá trị xếp hạng giữa các hàng mà không có khoảng cách giữa các hạng       |

# Biểu thức trong Transact-SQL

## Biểu thức:

- Biểu thức là kết hợp nhiều phần tử: toán tử, giá trị, định danh mà SQL Server có thể ước tính được để trả về một kết quả.
- Biểu thức có thể được sử dụng trong nhiều trường hợp khi truy cập hoặc thay đổi dữ liệu.

## Sử dụng toán tử trong biểu thức:

- Toán tử được dùng để thực hiện tính toán, so sánh, ghép hoặc gán giá trị.

# Biểu thức trong Transact-SQL

Bảng dưới đây mô tả chi tiết các toán tử khác nhau trong SQL Server:

| Toán tử          | Mô tả                                                                                                            |
|------------------|------------------------------------------------------------------------------------------------------------------|
| So sánh          | So sánh các giá trị dữ liệu                                                                                      |
| Logicl           | Toán tử luận lí, dùng kiểm tra điều kiện đúng hay sai. Các toán tử luận lí: AND, OR, NOT, LIKE, ANY, ALL hoặc IN |
| Số học           | Toán tử thực hiện các phép tính số học như: +, -, *, /                                                           |
| Toán tử một ngôi | Toán tử thực hiện trên một toán hạng                                                                             |
| Nhị phân Bitwise | Các toán tử thực hiện trên các bit dữ liệu                                                                       |
| Hàm ghép chuỗi   | Ghép hai chuỗi thành một chuỗi                                                                                   |
| Toán tử gán      | Gán giá trị cho biến                                                                                             |

## Điều kiện kết ở mệnh đề FROM (tt)

---

- Ví dụ:
  - Tìm mã và tên các nhân viên làm việc tại phòng “Kinh doanh”
  - Cho biết họ tên nhân viên và tên phòng ban mà họ là trưởng phòng



## Điều kiện kết ở mệnh đề FROM (tt)

---

- Ví dụ:
  - Tìm họ tên các nhân viên và tên các đề án nhân viên tham gia (nếu có)
  - Tìm họ tên các nhân viên không có thân nhân nào (not exists / left joint)

## Một số dạng câu truy vấn khác

---

- Tìm họ tên của những trưởng phòng không có thân nhân nào.
- Danh sách những trưởng phòng có tối thiểu một thân nhân (có ít nhất một thân nhân, có từ một thân nhân trở lên
-

## Nội dung chương IV

---

- Câu lệnh thao tác dữ liệu
  - Truy vấn dữ liệu cơ bản
  - Truy vấn lồng
  - Hàm kết hợp và gom nhóm
  - Một số dạng truy vấn khác
- **Khung nhìn**

## Khung nhìn

- 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 (tt)

- 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

# Định nghĩa khung nhìn

- 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

## Ví dụ

```
CREATE VIEW NV_P5 AS
  SELECT MANV, HONV, TENLOT, TENVN
  FROM NHANVIEN
  WHERE PHG=5
```

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

## Truy vấn trên khung nhìn

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

```
SELECT TENNV  
FROM NV_P5  
WHERE HONV LIKE 'Nguyễn'
```

$$NV\_P5 \leftarrow \pi_{MANV, HONV, TENLOT, TENNV} (\sigma_{PHG=5} (NHANVIEN))$$
$$\pi_{TENN} (\sigma_{HONV='Nguyễn'} (NV\_P5))$$



## Truy vấn trên khung nhìn (tt)

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

```
SELECT HONV, TENVN, TENDA, THOIGIAN  
FROM NV_P5, PHANCONG, DEAN  
WHERE MANV=MA_NVLEN AND SODA=MADA
```

$$NV\_P5 \leftarrow \pi_{MANV, HONV, TENLOT, TENNV} (\sigma_{PHG=5} (NHANVIEN))$$
$$TMP \leftarrow NV\_P5 \bowtie_{MANV=MA\_NVLEN} PHONGBAN \bowtie_{SODA=MADA} DEAN$$
$$\pi_{TENN, TENDA, THOIGIAN} (TMP)$$

## 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 (tt)

- Sửa lại họ cho nhân viên mã '123456789' ở phòng 5 là 'Pham'

```
UPDATE NV_P5  
SET HONV='Pham'  
WHERE MANV= '123456789'
```

# BÀI TẬP

1 Cho csdl sau:

- Tacgia(matg, hotentg, namsinh, quequan, madv)
  - Donvi(madv, tendv, socb)
  - Tapchi(matc, tentc, noixb)
  - Chitiettc(macttc, matc, sora, sotrang)
  - Baibao(mabb, tenbb, macttc, ngaydang)
  - Tacgia\_baibao(matg, mabb)
- a) Cho biết ds bài báo đăng vào ngày 19/2/2009
- b) Cho biết ds các chi tiết tạp chí của tạp chí “Khoa học công nghệ”
- c) Cho biết danh sách tác giả (matg, hoten, namsinh, tendv) là tác giả của bài báo có mã "bbcantt01"
- d) Cho biết ds tác giả (matg, hoten) của những tác giả không có bài báo nào

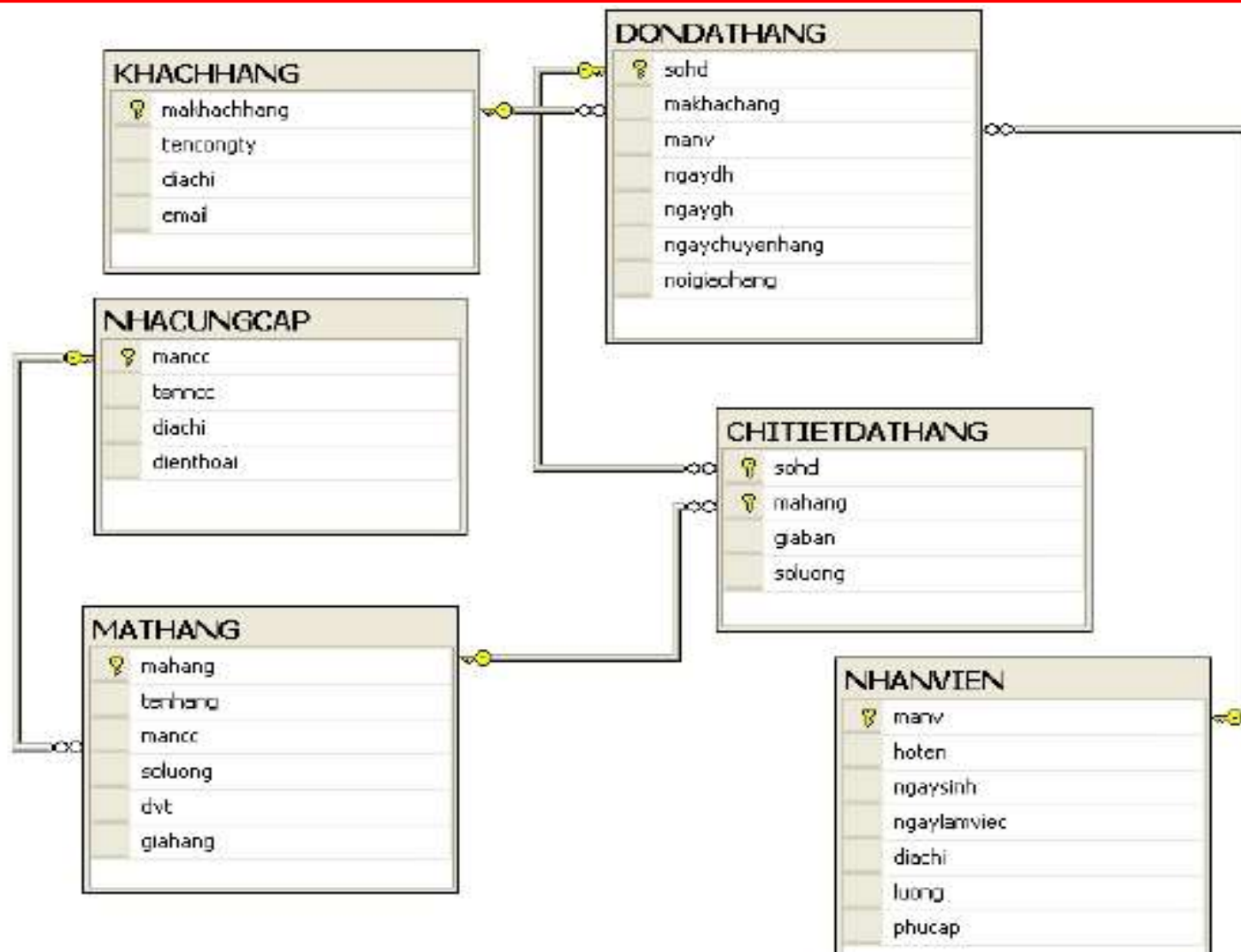
- e) Cho biết ds các bài báo (mabb, tenbb, ngaydang, tentc, sora), của tạp chí “khoa học và công nghệ” số 4)
- f) Cho biết ds tác giả (matg, hoten, quequan, tendv) có đăng bài trên tạp chí “Khoa học và ứng dụng”

## Bài tập 2<sup>3</sup>

- NhanVien(manv, honv, tenlot, tennv, ngaysinh, diachi, phai, luong, manql, phong)
  - PhongBan(maphong, tenphong, trphong, ngnhanchuc)
  - DDPhong(maphong, diadiem)
  - DeAn(tenda, mada, ddiemda, phong)
  - PhanCong(manvien, soda, thoigian)
  - ThanNhan(manvien, tentn, phai, ngaysinh, quanhe)
- a) Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên phòng 4
- b) Tìm những nhân viên có lương lớn hơn lương của tất cả nhân viên phòng 4

## BÀI TẬP chương IV

- Cho lược đồ csdl QLBanHang được mô tả như sau:



## BÀI TẬP chương IV (tt)

---

- Thực hiện các yêu cầu sau bằng SQL
  - Tạo csdl và tạo các bảng, nhập dữ liệu cho các bảng trong csdl
  - Cho biết mã và tên của các mặt hàng có giá lớn hơn 10 và số lượng hiện có ít hơn 20
  - Cho biết thông tin những khách hàng nào đã mua mặt hàng áo Việt Tiến
  - Cho biết thông tin những mặt hàng nào chưa từng được khách hàng đặt mua?
  - Cho biết tổng số lượng bán được của mỗi mặt hàng
  - Bổ sung ràng buộc cho bảng DONDATHANG ràng buộc kiểm tra ngày giao hàng và ngày chuyển hàng phải sau hoặc bằng với ngày đặt hàng.
  - Cho biết thông tin những khách hàng có cùng ngày sinh
  - Thống kê số lượng hóa đơn đã lập của mỗi nhân viên