

# Chương 4: Ngôn ngữ SQL

# Nội dung

1. Giới thiệu
2. Các ngôn ngữ giao tiếp
3. Ngôn ngữ định nghĩa dữ liệu
4. Ngôn ngữ thao tác dữ liệu
5. Ngôn ngữ truy vấn dữ liệu có cấu trúc
6. Ngôn ngữ điều khiển dữ liệu

# 1. Giới thiệu

- ♦ Là ngôn ngữ chuẩn để truy vấn và thao tác trên CSDL quan hệ
- ♦ Là ngôn ngữ phi thủ tục
- ♦ Khởi nguồn của SQL là SEQUEL - *Structured English Query Language*, năm 1974)
- ♦ *Các chuẩn SQL*
  - SQL89
  - SQL92 (SQL2)
  - SQL99 (SQL3)

## 2. Các ngôn ngữ giao tiếp

- ♦ **Ngôn ngữ định nghĩa dữ liệu** (*Data Definition Language - DDL*): cho phép khai báo cấu trúc bảng, các mối quan hệ và các ràng buộc.
- ♦ **Ngôn ngữ thao tác dữ liệu** (*Data Manipulation Language - DML*): cho phép thêm, xóa, sửa dữ liệu.
- ♦ **Ngôn ngữ truy vấn dữ liệu** (*Structured Query Language – SQL*): cho phép truy vấn dữ liệu.
- ♦ **Ngôn ngữ điều khiển dữ liệu** (*Data Control Language – DCL*): khai báo bảo mật thông tin, cấp quyền và thu hồi quyền khai thác trên cơ sở dữ liệu.

# 3. Ngôn ngữ định nghĩa dữ liệu

## 3.1 Lệnh tạo bảng (CREATE)

- 3.1.1 Cú pháp

- 3.1.2 Một số kiểu dữ liệu

## 3.2 Lệnh sửa cấu trúc bảng (ALTER)

- 3.2.1 Thêm thuộc tính

- 3.2.2 Sửa kiểu dữ liệu của thuộc tính

- 3.2.3 Xoá thuộc tính

- 3.2.4 Thêm ràng buộc toàn vẹn

- 3.2.5 Xoá ràng buộc toàn vẹn

## 3.3 Lệnh xóa bảng (DROP)

## 3.1 Lệnh tạo bảng

### 3.1.1 Cú pháp

```
CREATE TABLE <tên_bảng>
(
  <tên_cột1>    <kiểu_dữ_liệu> [not null],
  <tên_cột2>    <kiểu_dữ_liệu> [not null],
  ...
  <tên_cột1>    <kiểu_dữ_liệu> [not null],
  khai báo khóa chính, khóa ngoại, ràng buộc
)
```

## 3.1 Lệnh tạo bảng (2)

### 3.1.2 Một số kiểu dữ liệu

Kiểu dữ liệu	SQL Server
Chuỗi ký tự	varchar(n), char(n), nvarchar(n), nchar(n)
Số	tinyint, smallint, int, numeric(m,n), decimal(m,n), float, real, smallmoney, money
Ngày tháng	smalldatetime, datetime
Luận lý	bit

## 3.1 Lệnh tạo bảng (3)

*Lược đồ CSDL quản lý bán hàng gồm có các quan hệ sau:*

**KHACHHANG** (MAKH, HOTEN, DCHI, SODT, NGSINH,  
DOANHISO, NGDK, CMND)

**NHANVIEN** (MANV,HOTEN, NGVL, SODT)

**SANPHAM** (MASP,TENSP, DVT, NUOCSX, GIA)

**HOADON** (SOHD, NGHD, MAKH, MANV, TRIGIA)

**CTHD** (SOHD,MASP,SL)



## 3.1 Lệnh tạo bảng (4)

Create table KHACHHANG

```
(  
    MAKH                char(4) primary key,  
    HOTEN                varchar(40),  
    DCHI                varchar(50),  
    SODT                varchar(20),  
    NGSINH              smalldatetime,  
    DOANHISO            money,  
    NGDK                smalldatetime,  
    CMND                varchar(10)  
)
```

## 3.1 Lệnh tạo bảng (5)

Create table CTHD

```
(  
    SOHD      int foreign key  
                references HOADON(SOHD),  
    MASP      char(4) foreign key  
                references SANPHAM(MASP),  
    SL        int,  
    constraint PK_CTHD primary key (SOHD, MASP)  
)
```

## 3.2 Sửa cấu trúc bảng(1)

### 3.2.1 Thêm thuộc tính

*ALTER TABLE tênbảng ADD tên cột kiểu dữ liệu*

- Ví dụ: thêm cột Ghi\_chu vào bảng khách hàng

*ALTER TABLE KHACHHANG ADD GHI\_CHU varchar(20)*

### 3.2.2 Sửa kiểu dữ liệu thuộc tính

*ALTER TABLE tênbảng ALTER COLUMN tên cột  
kiểu dữ liệu\_mới*

- ♦ Lưu ý:

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

## 3.2 Sửa cấu trúc bảng(2)

- **Ví dụ:** Sửa Cột Ghi\_chu thành kiểu dữ liệu varchar(50)  
*ALTER TABLE KHACHHANG ALTER COLUMN GHI\_CHU varchar(50)*
- Nếu sửa kiểu dữ liệu của cột Ghi\_chu thành varchar(5), mà trước đó đã nhập giá trị cho cột Ghi\_chu có độ dài hơn 5 ký tự thì không được phép.
- Hoặc sửa từ kiểu chuỗi ký tự sang kiểu số, ...

### 3.2.3 Xóa thuộc tính

*ALTER TABLE tên\_bảng DROP COLUMN tên\_cột*

- Ví dụ: xóa cột Ghi\_chu trong bảng KHACHHANG  
*ALTER TABLE NHANVIEN DROP COLUMN Ghi\_chu*

## 3.2 Sửa cấu trúc bảng(3)

### 3.2.4 Thêm ràng buộc toàn vẹn

**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) **[ON  
DELETE CASCADE] [ON  
UPDATE CASCADE]**

**CHECK** (tên\_cột điều\_kiện)

## 3.2 Sửa cấu trúc bảng(4)

### ♦ Ví dụ

- ALTER TABLE NHANVIEN ADD CONSTRAINT PK\_NV PRIMARY KEY (MANV)
- ALTER TABLE CTHD ADD CONSTRAINT FK\_CT\_SP FOREIGN KEY (MASP) REFERENCES SANPHAM(MASP)
- ALTER TABLE SANPHAM ADD CONSTRAINT CK\_GIA CHECK (GIA >=500)
- ALTER TABLE KHACHHANG ADD CONSTRAINT UQ\_KH UNIQUE (CMND)

## 3.2 Sửa cấu trúc bảng(5)

### 3.2.5 Xóa ràng buộc toàn vẹn

*ALTER TABLE tên\_bảng DROP CONSTRAINT  
tên\_ràng\_buộc*

■ Ví dụ:

- Alter table CTHD drop constraint FK\_CT\_SP
  - Alter table SANPHAM drop constraint ck\_gia
- ♦ Lưu ý: đối với ràng buộc khóa chính, muốn xóa ràng buộc này phải xóa hết các ràng buộc khóa ngoại tham chiếu tới nó

## 3.3 Lệnh xóa bảng

- ♦ **Cú pháp**

DROP TABLE tên\_bảng

- ♦ **Ví dụ:** xóa bảng KHACHHANG.

DROP TABLE KHACHHANG

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



## 4. Ngôn ngữ thao tác dữ liệu

- ♦ Gồm các lệnh:
  - 4.1 Lệnh thêm dữ liệu (INSERT)
  - 4.2 Lệnh sửa dữ liệu (UPDATE)
  - 4.3 Lệnh xóa dữ liệu (DELETE)

## 4.1 Thêm dữ liệu

- ♦ **Cú pháp**

INSERT INTO tên\_bảng (cột1,...,cộtn) VALUES  
(giá\_trị\_1,..., giá\_trị\_n)

INSERT INTO tên\_bảng VALUES (giá\_trị\_1, giá\_trị\_2,  
..., giá\_trị\_n)

- ♦ **Ví dụ:**

- insert into SANPHAM values('BC01','But chi', 'cay', 'Singapore', 3000)
- insert into SANPHAM(masp,tensp,dvt,nuocsx,gia) values ('BC01','But chi','cay','Singapore',3000)

## 4.2 Sửa dữ liệu

- ♦ **Cú pháp**

UPDATE tên\_bảng  
SET cột\_1 = giá\_trị\_1, cột\_2 = giá\_trị\_2 ....  
[WHERE điều\_kiện]

- ♦ **Lưu ý:** cẩn thận với các lệnh xóa và sửa, nếu không có điều kiện ở WHERE nghĩa là xóa hoặc sửa tất cả.

- ♦ **Ví dụ:** Tăng giá 10% đối với những sản phẩm do “Trung Quoc” sản xuất

```
UPDATE SANPHAM  
SET Gia = Gia*1.1  
WHERE Nuocsx='Trung Quoc'
```

## 4.3 Xóa dữ liệu

- ♦ **Cú pháp**

DELETE FROM tên\_bảng [WHERE điều\_kiện]

- ♦ **Ví dụ:**

- Xóa toàn bộ nhân viên

DELETE FROM NHANVIEN

- Xóa những sản phẩm do Trung Quốc sản xuất có giá thấp hơn 10000

DELETE FROM SANPHAM

WHERE (Gia < 10000) and (Nuocsx='Trung Quoc')

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

- 5.1 Câu truy vấn tổng quát
- 5.2 Truy vấn đơn giản
- 5.3 Phép kết
- 5.4 Đặt bí danh, sử dụng \*, distinct
- 5.5 Các toán tử
- 5.6 Câu truy vấn con (subquery)
- 5.7 Phép chia
- 5.8 Hàm tính toán, gom nhóm

## 5.1 Câu truy vấn tổng quát

SELECT [DISTINCT] \*|tên\_cột | hàm  
FROM bảng  
[WHERE điều\_kiện]  
[GROUP BY tên\_cột]  
[HAVING điều\_kiện]  
[ORDER BY tên\_cột ASC | DESC]

## 5.2 Truy vấn đơn giản(1)

### ◆ SELECT

- Tương đương phép chiếu của ĐSQH
- Liệt kê các thuộc tính cần hiển thị trong kết quả

### ◆ WHERE

- Tương ứng với điều kiện chọn trong ĐSQH
- Điều kiện liên quan tới thuộc tính, sử dụng các phép nối luận lý AND, OR, NOT, các phép toán so sánh, BETWEEN

### ◆ FROM

- Liệt kê các quan hệ cần thiết, các phép kết

## 5.2 Truy vấn đơn giản(2)

- ♦ Tìm masp, tensp do “Trung Quoc” sản xuất có giá từ 20000 đến 30000

```
Select      masp,tensp
From        SANPHAM
Where       nuocsx='Trung Quoc'
           and gia between 20000 and 30000
```



## 5.3 Phép kết(1)

- ◆ Inner Join, Left Join, Right Join, Full Join

- ◆ **Ví dụ:**

- In ra danh sách các khách hàng (MAKH, HOTEN) đã mua hàng trong ngày 1/1/2007.

```
select  KHACHHANG.makh,hoten
from    KHACHHANG inner join HOADON on
        KHACHHANG.makh=HOADON.makh
where   nghd='1/1/2007'
```

## 5.3 Phép kết (2)

- ♦ **Ví dụ:** In ra danh sách tất cả các hóa đơn và họ tên của khách hàng mua hóa đơn đó (nếu có)
  - Select    sohd, hoten  
From    HOADON left join KHACHHANG on  
         HOADON.makh=KHACHHANG.makh
  - Select    sohd, hoten  
From    HOADON ,KHACHHANG  
where    HOADON.makh\*=KHACHHANG.makh

## 5.4 Đặt bí danh, sử dụng \*, distinct

- ◆ Đặt bí danh – Alias: cho thuộc tính và quan hệ:  
tên\_cũ AS tên\_mới
  - Select manv,hoten as [ho va ten] From NHANVIEN
- ◆ Liệt kê tất cả các thuộc tính của quan hệ:
  - Select \* from Nhanvien
  - Select NHANVIEN.\* from NHANVIEN
- ◆ Distinct: trùng chỉ lấy một lần
  - Select distinct nuocsx from SANPHAM
- ◆ Sắp xếp kết quả hiển thị: Order by
  - Select \* from SANPHAM order by nuocsx, gia DESC

## 5.5 Toán tử truy vấn(1)

- ♦ Toán tử so sánh: =, >, <, >=, <=, <>
- ♦ Toán tử logic: AND, OR, NOT
- ♦ Phép toán: +, -, \*, /
- ♦ BETWEEN .... AND
- ♦ IS NULL, IS NOT NULL
- ♦ LIKE ( \_ %)
- ♦ IN, NOT IN
- ♦ EXISTS , NOT EXISTS
- ♦ SOME, ALL

## 5.5 Toán tử truy vấn(2)

### ♦ IS NULL, IS NOT NULL

- Select sohđ from HOADON where makh is Null
- Select \* from HOADON where makh is Not Null

### ♦ Toán tử so sánh, phép toán

- Select gia\*1.1 as [gia ban] from SANPHAM where nuocsx<>'Viet Nam'
- Select \* from SANPHAM where (gia between 20000 and 30000) OR (nuocsx='Viet Nam')

### ♦ Toán tử IN, NOT IN

- Select \* from SANPHAM where masp NOT IN ('BB01','BB02','BB03')

## 5.5 Toán tử so sánh(3)

### 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
- **Ví dụ:** Select masp,tensp from SANPHAM  
where masp like 'B%01'

## 5.6 Câu truy vấn con (1)

customers table

customer_id	last_name	first_name
4000	Jackson	Joe
5000	Smith	Jane
6000	Ferguson	Samantha
7000	Reynolds	Allen
8000	Anderson	Paige
9000	Johnson	Derek

## 5.6 Câu truy vấn con (2)

orders table

order_id	customer_id
1	7000
2	5000
3	8000
4	4000
5	7000



## 5.6 Câu truy vấn con (4)

### In hoặc Exists

- ♦ **Ví dụ:** In ra thông tin của khách hàng có mua ít nhất 1 lần
- ♦ **SELECT \***  
FROM customers where customers.customer\_id in  
( SELECT orders.customer\_id  
FROM orders  
)

## 5.6 Câu truy vấn con (5)

### In hoặc Exists

- ♦ **Ví dụ:** In ra thông tin của khách hàng có mua ít nhất 1 lần
- ♦ `SELECT *`  
`FROM customers`  
`where EXISTS`  
`(SELECT *`  
`FROM orders`  
`WHERE customers.customer_id = orders.customer_id)`

## 5.6 Câu truy vấn con (5)

4000	Jackson	Joe
5000	Smith	Jane
7000	Reynolds	Allen
8000	Anderson	Paige

## 5.6 Câu truy vấn con (3)

### In hoặc Exists

- ♦ **Ví dụ:** Tìm các số hóa đơn mua cùng lúc 2 sản phẩm có mã số “BB01” và “BB02”.
  - `select distinct sohd  
from CTHD where masp='BB01' and sohd IN  
(select distinct sohd from CTHD where masp='BB02')  
select distinct A.sohd  
from CTHD A where A.masp='BB01' and  
EXISTS (select * from CTHD B  
where B.masp='BB02' and A.sohd=B.sohd)`
  - A.sohd=B.sohd: Có nghĩa trên cùng một hóa đơn

## 5.6 Câu truy vấn con (6)

### Not In hoặc Not Exists

- ♦ **Ví dụ:** Tìm những khách hàng không mua hàng

- ♦ `SELECT * FROM customers`

`where customers.customer_id not in  
(SELECT orders.customer_id  
FROM orders  
)`

## 5.6 Câu truy vấn con (7)

### **Not In hoặc Not Exists**

- ♦ **Ví dụ:** Tìm những khách hàng không mua hàng
- ♦ `SELECT * FROM customers`  
`WHERE NOT EXISTS`  
`(SELECT *`  
`FROM orders`  
`WHERE customers.customer_id = orders.customer_id);`

## 5.6 Câu truy vấn con (8)

6000	Ferguson	Samantha
9000	Johnson	Derek

## 5.6 Câu truy vấn con (9)

### Not In hoặc Not Exists

- ♦ **Ví dụ:** Tìm các số hóa đơn có mua sản phẩm mã số 'BB01' nhưng không mua sản phẩm mã số 'BB02'.
  - `select distinct sohd`  
`from CTHD where masp='BB01' and sohd NOT IN`  
`(select distinct sohd from CTHD where masp='BB02')`
  - `select distinct A.sohd`  
`from CTHD A where A.masp='BB01' and`  
`NOT EXISTS (select * from CTHD B`  
`where B.masp='BB02' and A.sohd=B.sohd)`



# Nhận xét IN và EXISTS

## ◆ IN

- 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

# EXIST và NOT EXIST

- ♦ **EXIST và NOT EXIST** chỉ trả về giá trị **True** hoặc **False**
- ♦ Giá trị của **EXIST**:
  - **True**: khi câu truy vấn con **có** bộ giá trị trả về
  - **False**: khi câu truy vấn con **không** trả về bộ giá trị nào
- ♦ Giá trị của **NOT EXIST**:
  - **True**: khi câu truy vấn con **không** trả về bộ giá trị nào
  - **False**: khi câu truy vấn con **có** bộ giá trị trả về.

## 5.7 Phép chia (tt)

PHICONG	
MAPC	TENPC
A	Peter
B	John
C	Red

PHICO	
MAPHICO	TENPHICO
X	Boeing-777
Y	Boeing-787
Z	Airbus-A380

CHUNGNHAN	
MAPC	MAPHICO
A	X
A	Y
B	Z
C	X
C	Y
C	Z

## 5.7 Phép chia

### Sử dụng NOT EXISTS

- ♦ Ví dụ: Tìm Phi công có thể lái tất cả phi cơ
- ♦ Select MAPC from PHICONG where **not exists** (1)  
(select \* from PHICO where **not exists** (2)  
(select \* from CHUNGNHAN CN  
where PHICO.MAPHICO=CN.MAPHICO and  
CN.MAPC= PHICONG. MAPC))

## 5.7 Phép chia (tt)

PHICONG	
MAPC	TENPC
A	Peter
B	John
C	Red

PHICO	
MAPHICO	TENPHICO
X	Boeing-777
Y	Boeing-787
Z	Airbus-A380

CHUNGNHAN	
MAPC	MAPHICO
A	X
A	Y
B	Z
C	X
C	Y
C	Z

[A,X] -> (2) sai -> ko chọn  
[A,Y] -> (2) sai -> ko chọn  
[A,Z] -> (2) đúng -> chọn  
=>(1) sai do có Z => ko chọn A

[B,X] -> (2) đúng -> chọn  
[B,Y]-> (2) đúng -> chọn  
[B,Z]> (2) sai -> ko chọn  
=> (1) sai do có X,Y => ko chọn B

[C,X] -> (2) Sai -> ko chọn  
[C,Y] -> (2) sai -> ko chọn  
[C,Z] -> (2) sai -> ko chọn  
=> (1) đúng do ko có gì cả => chọn C  
C là phi công có thể lái tất cả phi cơ

## 5.7 Phép chia

- ♦ **Ví dụ:** Tìm những sản phẩm do “Singapore” sản xuất mà hóa đơn 1001 chưa mua (not in)

**SELECT \***

**FROM SANPHAM**

**WHERE NUOCSX='SINGAPORE' AND MASP**

**NOT IN(--Câu truy vấn con**

**SELECT MASP**

**FROM CTHD**

**WHERE SOHD='1001')**

## 5.7 Phép chia

- ♦ **Ví dụ:** Tìm những sản phẩm do “Singapore” sản xuất mà hóa đơn 1001 chưa mua (Dùng Not exists)

**SELECT \***

**FROM SANPHAM A**

**WHERE NUOCSX='SINGAPORE' AND NOT  
EXISTS(--Câu truy vấn con**

**SELECT \***

**FROM CTHD B**

**WHERE SOHD='1001' AND A.MASP=B.MASP)**

## 5.7 Kết luận

- ◆ Nếu câu truy vấn con không trả về bộ giá trị nào  
=> ĐIỀU KIỆN **NOT EXISTS ĐÚNG**
- ◆ NHƯ VẬY, vấn đề: **Tìm những sản phẩm của Singapore mà SoHD '1001' chưa mua.** đã giải quyết xong!
- ◆ Từ đó ta có thể dễ dàng tìm những sản phẩm của Singapore mà một SoHD **BẤT KỲ** chưa mua.



## 5.7 Phép chia (tt)

- ♦ **Ví dụ:** Tìm số hóa đơn đã mua tất cả những sản phẩm do “Singapore” sản xuất.
- ♦ Theo lý thuyết đại số quan hệ, để giải quyết câu hỏi này, ta sử dụng **phép chia**.

**R1 ← CTHD[SOHD,MASP]**

**R2 ← SANPHAM: NUOCSX= ‘SINGAPORE’**

**R3 ← R2[MASP]**

**R4 ← R1:R3**

## 5.7 Phép chia (tt)

- ♦ Câu hỏi phụ yêu cầu chúng ta **tìm những sản phẩm của Singapore mà một Số hóa đơn bất kỳ chưa mua**
- ♦ Vậy nếu giá trị của câu hỏi phụ trả về là **không có sản phẩm nào** thì có nghĩa là **Số hóa đơn đó đã mua tất cả sản phẩm của Singapore**

## 5.7 Phép chia (tt)

```
SELECT DISTINCT SOHD FROM CTHD A  
WHERE NOT EXISTS(--câu hỏi phụ  
    SELECT * FROM SANPHAM B  
    WHERE NUOCSX='SINGAPORE' AND NOT EXISTS(  
        SELECT * FROM CTHD C  
        WHERE C.MASP=B.MASP AND C.SOHD=A.SOHD) )
```

## 5.7 Phép chia (tt)

- ♦ Select sohd from HOADON where not exists  
(select \* from SANPHAM  
where nuocsx= 'Singapore' and not exists  
(select \* from CTHD where  
HOADON.sohd=CTHD.sohd and  
CTHD.masp=SANPHAM.masp))

## 5.8 Các hàm tính toán và gom nhóm (1)

### 5.8.1 Các hàm tính toán cơ bản

- COUNT: Đếm số bộ dữ liệu của thuộc tính
- MIN: Tính giá trị nhỏ nhất
- MAX: Tính giá trị lớn nhất
- AVG: Tính giá trị trung bình
- SUM: Tính tổng giá trị các bộ dữ liệu

## NHANVIEN

MANV	HOTEN	PHAI	MANQL	PHONG	LUONG
NV001	Nguyễn Ngọc Linh	Nữ	Null	NC	2.800.000
NV002	Đinh Bá Tiến	Nam	NV002	DH	2.000.000
NV003	Nguyễn Văn Mạnh	Nam	NV001	NC	2.300.000
NV004	Trần Thanh Long	Nam	NV002	DH	1.800.000
NV005	Nguyễn Thị Hồng Vân	Nữ	NV001	NC	2.500.000
NV006	Nguyễn Minh	Nam	NV002	DH	2.000.000
NV007	Hà Duy Lập	Nam	NV003	NC	1.800.000
NV008	Trần Kim Duyên	Nữ	NV003	NC	1.800.000
NV009	Nguyễn Kim Anh	Nữ	NV003	NC	2.000.000

# Ví dụ

1. Tính lương thấp nhất, cao nhất, trung bình và tổng lương của tất cả các nhân viên.
2. Có tất cả bao nhiêu nhân viên
3. Bao nhiêu nhân viên có người quản lý
4. Bao nhiêu phòng ban có nhân viên trực thuộc
5. Tính lương trung bình của các nhân viên
6. Tính lương trung bình của các nhân viên theo từng phòng ban

# Giải

1. Tính lương thấp nhất, cao nhất, trung bình và tổng lương của tất cả các nhân viên.

```
SELECT  min(luong) as thapnhat,  
        max(luong) as caonhat,  
        avg(luong) as trungbinh,  
        sum(luong) as tongluong
```

```
FROM    NhanVien
```



# Giải

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
```

## 5.8 Các hàm tính toán và gom nhóm (2)

### 5.8.2 Gom nhóm: mệnh đề GROUP BY

- ◆ Sử dụng hàm gom nhóm trên các bộ trong quan hệ.
- ◆ Mỗi nhóm bộ bao gồm tập hợp các bộ có cùng giá trị trên các thuộc tính gom nhóm
- ◆ Hàm gom nhóm áp dụng trên mỗi bộ độc lập nhau.
- ◆ SQL có mệnh đề GROUP BY để chỉ ra các thuộc tính gom nhóm, các thuộc tính này phải xuất hiện trong mệnh đề SELECT

# Giải

## 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
```

## 5.8 Các hàm tính toán và gom nhóm (3)

### 5.8.3 Điều kiện sau gom nhóm: mệnh đề HAVING

- Lọc kết quả theo điều kiện, sau khi đã gom nhóm
- Điều kiện ở HAVING được thực hiện sau khi gom nhóm, các điều kiện có liên quan đến thuộc tính Group By
- ◆ **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