

Chương 4: Ngôn ngữ SQL

Giảng viên: ThS. Thái Bảo Trân
Thời lượng: 12 tiết

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**

1. Giới thiệu (1)

- **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

- **Lịch sử phát triển**
 - ❖ Khởi nguồn của SQL là SEQUEL - *Structured English QUery Language*, năm 1974)
 - ❖ Được phát triển bởi IBM (1970s)
 - ❖ Được ANSI công nhận và phát triển thành chuẩn
 - SQL89
 - SQL92 (SQL2)
 - SQL99 (SQL3)

1. Giới thiệu (2)

■ 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 sử dụng thuật ngữ

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

1. Giới thiệu (3)

■ SQL gồm

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

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 (1)

■ Là ngôn ngữ mô tả

- ❖ Lược đồ cho mỗi quan hệ
- ❖ Miền giá trị tương ứng của từng thuộc tính
- ❖ Ràng buộc toàn vẹn
- ❖ Chỉ mục trên mỗi quan hệ

■ Gồm

- ❖ CREATE TABLE (tạo bảng)
- ❖ DROP TABLE (xóa bảng)
- ❖ ALTER TABLE (sửa bảng)
- ❖ CREATE DOMAIN (tạo miền giá trị)
- ❖ CREATE DATABASE
- ❖ ...

3. Ngôn ngữ định nghĩa dữ liệu (2)

3.1 Lệnh tạo bảng (CREATE)

3.1.1 Cú pháp

3.1.2 Một số kiểu dữ liệu

3.1.3 Ràng buộc toàn vẹn

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 Xóa thuộc tính

3.2.4 Thêm ràng buộc toàn vẹn

3.2.5 Xóa ràng buộc toàn vẹn

3.3 Lệnh xóa bảng (DROP)

3.4 Lệnh tạo miền giá trị

3.1 Lệnh tạo bảng (1)

3.1.1 Cú pháp

CREATE TABLE <tên_bảng>

(

<tên_cột1> <kiểu_dữ_liệu> [<RBTV>],

<tên_cột2> <kiểu_dữ_liệu> [<RBTV>],

...

<tên_cộtn> <kiểu_dữ_liệu> [<RBTV>],

[<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	<code>varchar(n)</code> , <code>char(n)</code> , <code>nvarchar(n)</code> , <code>nchar(n)</code>
Số	<code>tinyint</code> , <code>smallint</code> , <code>int</code> , <code>numeric(m,n)</code> , <code>decimal(m,n)</code> , <code>float</code> , <code>real</code> , <code>smallmoney</code> , <code>money</code>
Ngày tháng	<code>smalldatetime</code> , <code>datetime</code>
Luận lý	<code>bit</code>

3.1.3 Ràng buộc toàn vẹn

■ <RBTV>

- ❑ NOT NULL
- ❑ NULL
- ❑ UNIQUE
- ❑ DEFAULT
- ❑ PRIMARY KEY
- ❑ FOREIGN KEY / REFERENCES
- ❑ CHECK

■ Đặt tên cho RBTV

CONSTRAINT <Ten_RBTV> <RBTV>

3.1 Ví dụ 1 - Lệnh tạo bảng (1)

Ví dụ 1: Cho lược đồ CSDL quản lý bán hàng gồm có các quan hệ:

KHACHHANG (MAKH, HOTEN, DCHI, SODT, NGSINH, DOANHSONG, 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 Ví dụ 1 - Lệnh tạo bảng (2)

CREATE TABLE KHACHHANG

(

MAKH	char(4),
HOTEN	varchar(40),
DCHI	varchar(50),
SODT	varchar(20),
NGSINH	smalldatetime,
DOANH SO	money,
NGDK	smalldatetime,
CMND	varchar(10)

)

3.1 Ví dụ 1- RBTV (1)

CREATE TABLE KHACHHANG

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

3.1 Ví dụ 1 - RBTV (3)

CREATE TABLE NHANVIEN

(

manv char(4) primary key,

hoten varchar(40),

sodt varchar(20),

ngvl smalldatetime

)

3.1 Ví dụ 1 - RBTV (3)

```
CREATE TABLE SANPHAM  
(  
    masp    char(4) primary key,  
    tensp   varchar(40),  
    dvt     varchar(20),  
    nuocsx  varchar(40),  
    gia     money  
)
```


3.1 Ví dụ 1 - RBTV (4)

CREATE TABLE HOADON

(

sohd int primary key,

nghd smalldatetime,

makh char(4),

manv char(4),

trigia money

)

3.1 Ví dụ 1 - RBTV (5)

CREATE TABLE CTHD

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

3.1 Ví dụ 2 - RBTV (1)

Ví dụ 2: Cho lược đồ CSDL “Quản lý đề án công ty” như sau:

NHANVIEN (MaNV, HoNV, TenDem, TenNV, NTNS, Dchi, GT, Luong, Ma_NQL, PHG)

PHONGBAN (MaPH, TenPH, TRPH, NG_Nhanchuc)

DEAN (MaDA, TenDA, Phong, NamThucHien)

PHANCONG (MaNV, MaDA, ThoiGian)

3.1 Ví dụ 2- RBTV (2)

```
CREATE TABLE NHANVIEN (  
    MANV CHAR(9) PRIMARY KEY,  
    HONV VARCHAR(10) NOT NULL,  
    TENDEM VARCHAR(20) ,  
    TENNV VARCHAR(10) NOT NULL,  
    NTNS DATETIME,  
    DCHI VARCHAR(50),  
    GT CHAR(3) CHECK (GT IN ('Nam', 'Nu')),  
    LUONG INT DEFAULT (2000000),  
    MA_NQL CHAR(9),  
    PHG INT  
)
```

3.1 Ví dụ 2- RBTV (3)

```
CREATE TABLE PHONGBAN (  
    TENPB VARCHAR(20) UNIQUE,  
    MAPHG INT NOT NULL,  
    TRPHG CHAR(9),  
    NG_NHANCHUC DATETIME DEFAULT (GETDATE())  
)
```

```
CREATE TABLE PHANCONG (  
    MA_NVIENT CHAR(9) FOREIGN KEY (MA_NVIENT)  
        REFERENCES NHANVIEN(MANV),  
    SODA INT REFERENCES DEAN(MADA),  
    THOIGIAN DECIMAL(3,1)  
)
```

3.1 Ví dụ 2- RBTV (4)

```
CREATE TABLE NHANVIEN (  
    HONV VARCHAR(10) CONSTRAINT NV_HONV_NN NOT NULL,  
    TENDEM VARCHAR(20) NOT NULL,  
    TENNV VARCHAR(10) NOT NULL,  
    MANV CHAR(9) CONSTRAINT NV_MANV_PK PRIMARY KEY,  
    NS DATETIME,  
    DCHI VARCHAR(50),  
    GT CHAR(3) CONSTRAINT NV_GT_CHK  
        CHECK (GT IN ('Nam', 'Nu')),  
    LUONG INT CONSTRAINT NV_LUONG_DF DEFAULT (2000000),  
    MA_NQL CHAR(9),  
    PHG INT
```

)

3.1 Ví dụ 2- RBTV (5)

```
CREATE TABLE PHANCONG (  
    MA_NVIENT CHAR(9),  
    SODA INT,  
    THOIGIAN DECIMAL(3,1),  
    CONSTRAINT PC_MANVIEN_SODA_PK PRIMARY KEY  
        (MA_NVIENT, SODA),  
    CONSTRAINT PC_MANVIEN_FK FOREIGN KEY (MA_NVIENT)  
        REFERENCES NHANVIEN(MANV),  
    CONSTRAINT PC_SODA_FK FOREIGN KEY (SODA)  
        REFERENCES DEAN(MADA)  
)
```

3.2 Sửa cấu trúc bảng (1)

■ Được dùng để

□ Thay đổi cấu trúc bảng

□ Thay đổi RBTV

■ Thêm cột

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

■ Xóa cột

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

■ Mở rộng cột

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


3.2 Sửa cấu trúc bảng (2)

■ Thêm RBTV

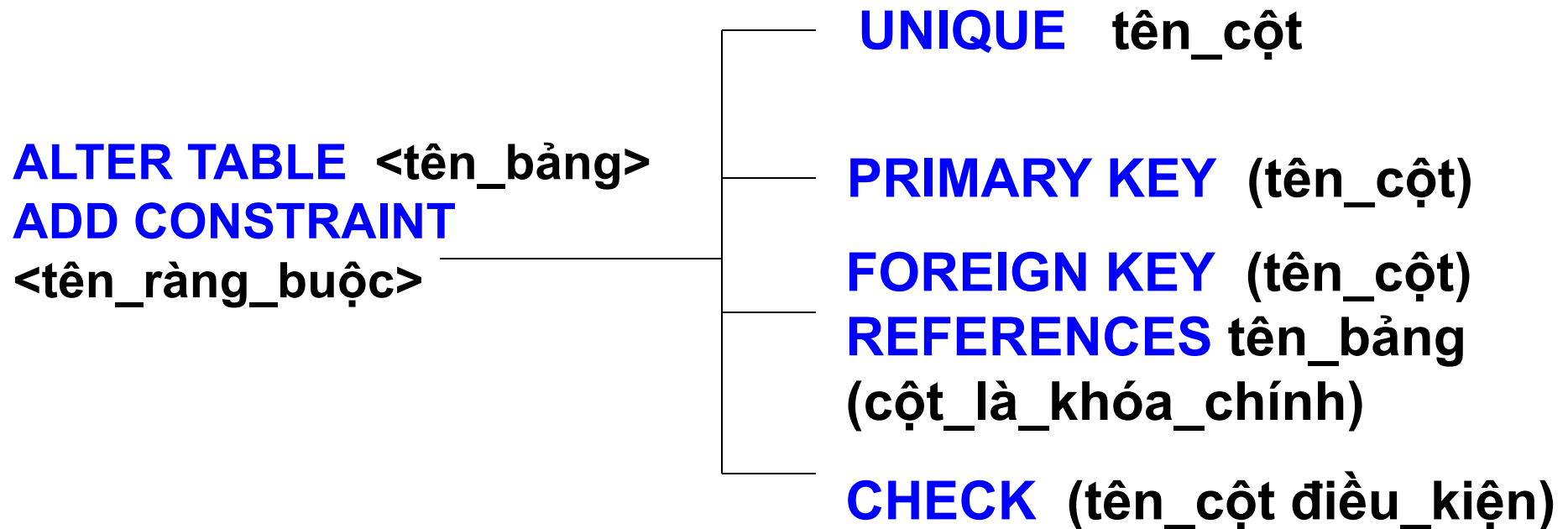
```
ALTER TABLE <Tên_bảng> ADD  
    CONSTRAINT <Ten_RBTV> <RBTV>,  
    CONSTRAINT <Ten_RBTV> <RBTV>,  
    ...
```

■ Xóa RBTV

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

3.2 Sửa cấu trúc bảng (3)

Chú ý: Thêm ràng buộc toàn vẹn



3.2 Ví dụ - Thay đổi cấu trúc bảng

- **Ví dụ 1:** Thêm cột Ghi_chu vào bảng khách hàng
ALTER TABLE KHACHHANG ADD GHI_CHU varchar(20)
- **Ví dụ 2:** Sửa Cột Ghi_chu thành kiểu dữ liệu varchar(50)
ALTER TABLE KHACHHANG ALTER COLUMN GHI_CHU varchar(50)
- ❖ **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, ...
- **Ví dụ 3:** Xóa cột Ghi_chu trong bảng KHACHHANG
ALTER TABLE NHANVIEN DROP COLUMN Ghi_chu

3.2 Ví dụ - Thay đổi RBTV (1)

- ❖ **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 Ví dụ - Thay đổi RBTV (2)

- ❖ **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

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

- **Cú pháp:**

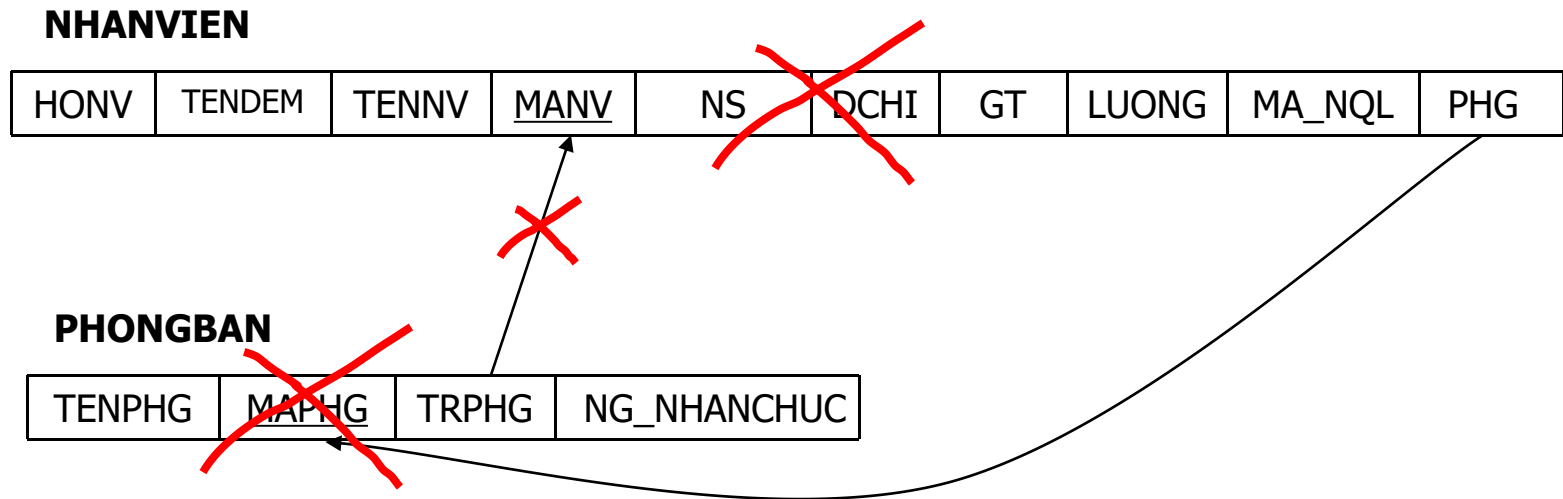
DROP TABLE tên_bảng

- **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.

3.3 Lệnh xóa bảng



3.4 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)
```


4. Ngôn ngữ thao tác dữ liệu

- **Gồm các lệnh:**

4.1 Lệnh thêm dữ liệu (INSERT)

4.2 Lệnh xóa dữ liệu (DELETE)

4.3 Lệnh sửa dữ liệu (UPDATE)

4.1 Lệnh INSERT (1)

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

4.1 Lệnh INSERT (2)

■ Cú pháp (thêm 1 dòng)

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

4.1 Ví dụ (1)

```
INSERT INTO NHANVIEN(HONV, TENDEM, TENNV, MANV)  
VALUES ('Le', 'Van', 'Tuyen', '635635635')
```

```
INSERT INTO NHANVIEN(HONV, TENDEM, TENNV, MANV, DCHI)  
VALUES ('Le', 'Van', 'Tuyen', '635635635', NULL)
```

```
INSERT INTO NHANVIEN  
VALUES ('Le', 'Van', 'Tuyen', '635635635', '12/30/1952', '98 HV', 'Nam',  
'37000', 4)
```

```
INSERT INTO SANPHAM VALUES ('BC01','But chi', 'cay',  
'Singapore', 3000)
```

```
INSERT INTO SANPHAM(masp,tensp,dvt,nuocsx,gia) VALUES  
( 'BC01','Butchi','cay','Singapore',3000)
```

4.1 Nhận xét

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

4.1 Lệnh INSERT (3)

■ Cú pháp (thêm nhiều dòng)

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

□ Cú pháp sao chép 1 bảng dữ liệu:

SELECT * INTO <Tên bảng mới> FROM <Tên bảng cũ>

Ví dụ: sao chép bảng KHACHHANG thành KHACHHANG_NEW

SELECT * INTO KHACHHANG_NEW FROM KHACHHANG

4.1 Ví dụ (2)

```
CREATE TABLE THONGKE_PB (  
    TENPHG VARCHAR(20),  
    SL_NV INT,  
    LUONG_TC INT  
)
```

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

4.2 Lệnh DELETE

- Dùng để xóa các dòng của bảng
- Cú pháp

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


4.2 Ví dụ (1)

DELETE FROM NHANVIEN

WHERE HONV='Tran'

DELETE FROM NHANVIEN

WHERE MANV='345345345'

DELETE FROM NHANVIEN

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

4.2 Nhận xét

- ❑ Số lượng các 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

4.2 Ví dụ (2)

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
453453453	Tran	Thanh	Tam	07/31/1972	543 MTL Q1	Nu	25000	333445555	5
999887777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	987654321	4
987654321	Le	Quynh	Nhu	07620/1951	219 TD Q3	Nu	43000	888665555	4
987987987	Tran	Hong	Quang	04/08/1969	980 LHP Q5	Nam	25000	987654321	4
888665555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

MA_NVIEN	SODA	THOIGIAN
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
453453453	1	20.0

4.2 Ví dụ (3)

TENPHG	MAPHG	MA_NVIEN	NG_NHANCHUC
Nghien cuu	5	333445555	05/22/1988
Dieu hanh	4	987987987	01/01/1995
Quan ly	1	888665555	06/19/1981

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
888665555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1
333445555	Nguyen	Thanh	Tung	12/08/1955	638 NVC Q5	Nam	40000	888665555	NULL
987987987	Nguyen	Manh	Hung	09/15/1962	Ba Ria VT	Nam	38000	333445555	NULL
453453453	Tran	Thanh	Tam	07/31/1972	543 MTL Q1	Nu	25000	333445555	NULL
999887777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	987654321	4
987654321	Le	Quynh	Nhu	07/20/1951	219 TD Q3	Nu	43000	888665555	4
987987987	Tran	Hong	Quang	04/08/1969	980 LHP Q5	Nam	25000	987654321	4

4.3 Lệnh UPDATE (1)

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

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

4.3 Ví dụ (1)

UPDATE NHANVIEN

SET NGSINH='08/12/1965'

WHERE MANV='333445555'

UPDATE NHANVIEN

SET LUONG=LUONG*1.1

UPDATE SANPHAM

SET Gia = Gia*1.1

WHERE Nuocsx='Trung Quoc'

4.3 Ví dụ (2)

- Với đề án có mã số 10, hãy thay đổi nơi thực hiện đề án thành 'Vung Tau' và phòng ban phụ trách là phòng 5
- **DEAN** (MaDA, TenDA, Phong, DIADIEM_DA)

UPDATE DEAN

SET DIADIEM_DA='Vung Tau', PHONG=5

WHERE MADA=10

4.3 Nhận xét

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

5. Ngôn ngữ truy vấn dữ liệu có cấu trúc

5.1 Giới thiệu

5.2 Câu truy vấn tổng quát

- ❑ Truy vấn đơn giản
- ❑ Mệnh đề Order by
- ❑ Đặt bí danh, sử dụng *, distinct
- ❑ Các toán tử
- ❑ Câu truy vấn con (subquery)
- ❑ Phép chia
- ❑ Hàm tính toán, gom nhóm
- ❑ Phép kết

5.1 Giới thiệu

- Là ngôn ngữ rút trích dữ liệu thỏa một số điều kiện nào đó
- Dựa trên

Phép toán ĐSQH

+

Một số bổ sung

- ❖ Cho phép 1 bảng có nhiều dòng trùng nhau
- ❖ Bảng là *bag* \neq quan hệ là *set*

5.1 Giới thiệu (2)

❑ Toán tử so sánh:

- =, >, <, >=, <=, <>
- BETWEEN
- IS NULL, IS NOT NULL
- LIKE (% , _)
- IN, NOT IN
- EXISTS, NOT EXISTS
- SOME, ALL, ANY

❑ Toán tử logic: AND, OR.

❑ Các phép toán: +, -, *, /

❑ Các hàm xử lý ngày (DAY()), tháng (MONTH()), năm (YEAR())

❑ 5 hàm: COUNT(), SUM(), MAX(), MIN(), AVG()

5.2 Câu truy vấn tổng quát (1)

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 Câu truy vấn tổng quát (2)

Phân loại câu SELECT:

- ❑ **SELECT đơn giản**
- ❑ **SELECT có mệnh đề ORDER BY**
- ❑ **SELECT lồng (câu SELECT lồng câu SELECT khác)**
- ❑ **SELECT gom nhóm (GROUP BY)**
- ❑ **SELECT gom nhóm (GROUP BY) có điều kiện HAVING**

Truy vấn đơn giản (1)

■ Gồm 3 mệnh đề

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

➤ <danh sách các cột>

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

➤ <danh sách các bảng>

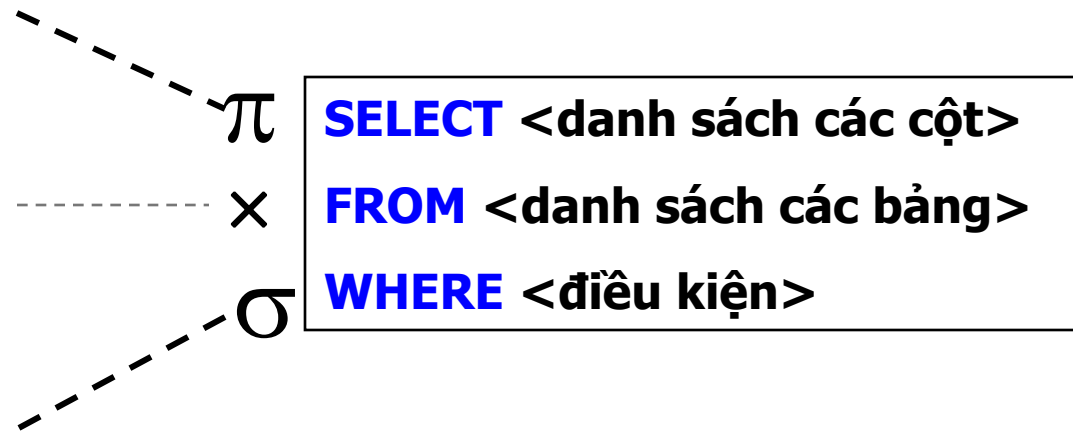
- ❖ Tên các bảng liên quan đến câu truy vấn

➤ <điều kiện>

- ❖ Biểu thức boolean xác định dòng nào sẽ được rút trích
- ❖ Nối các biểu thức: AND, OR, và NOT
- ❖ Phép toán: < , > , ≤ , ≥ , ≠ , =, LIKE và BETWEEN

Truy vấn đơn giản (2)

■ SQL và ĐSQH



SELECT L
FROM $\pi(\sigma_C(R))$ →
WHERE C

Truy vấn đơn giản (3)

■ SELECT

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

■ WHERE

- ❖ Là 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.
- ❖ Là phép tích Descartes các quan hệ (không có từ khóa JOINT, chỉ là dấu phẩy giữa các quan hệ).
- ❖ Là phép kết các quan hệ (nếu có từ khóa JOINT)

Ví dụ - Truy vấn đơn giản (1)

- **NHANVIEN** (MaNV, HoNV, TenDem, TenNV, NTNS, Dchi, GT, Luong, Ma_NQL, PHG)
- **PHONGBAN** (MaPH, TenPH, TRPH, NG_Nhanchuc)
- **DEAN** (MaDA, TenDA, Phong, NamThucHien)
- **PHANCONG** (MaNV, MaDA, ThoiGian)

Ví dụ - Truy vấn đơn giản (2)

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

Lấy tất cả các cột
của quan hệ kết
quả

MANV	HONV	TENDEM	TENNV	NS	DCHI	GT	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

Ví dụ - Mệnh đề SELECT (1)

```
SELECT MANV, HONV, TENDEM, TENNV  
FROM NHANVIEN  
WHERE PHG=5 AND GT='Nam'
```

MANV	HONV	TENDEM	TENNV
333445555	Nguyen	Thanh	Tung
987987987	Nguyen	Manh	Hung

Ví dụ - Mệnh đề SELECT (2)

Tên bí danh

```
SELECT MANV, HONV AS 'HO', TENDEM AS 'TEN DEM', TENNV AS 'TEN'  
FROM NHANVIEN  
WHERE PHG=5 AND GT='Nam'
```

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

Ví dụ - Mệnh đề SELECT (3)

Mở
rộng

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

MANV	HO TEN
333445555	Nguyen Thanh Tung
987987987	Nguyen Manh Hung

Ví dụ - Mệnh đề SELECT (4)

Mở
rộng

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

MANV	LUONG10%
333445555	33000
987987987	27500

Ví dụ - Mệnh đề SELECT (5)

Loại bỏ các dòng trùng nhau

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

LUONG

30000

25000

28000

38000

- Tồn chi phí

- Người dùng muốn thấy

Ví dụ - Mệnh đề SELECT (6)

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

NHANVIEN (MaNV, HoNV, TenDem, TenNV, NTNS, Dchi, GT, Luong, Ma_NQL, PHG)

PHONGBAN (MaPH, TenPH, TRPH, NG_Nhanchuc)

DEAN (MaDA, TenDA, Phong, NamThucHien)

PHANCONG (MaNV, MaDA, ThoiGian)

Ví dụ - Mệnh đề WHERE (1)

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

Biểu thức logic

↓

TRUE

↓

TRUE

Ví dụ - Mệnh đề WHERE (2)

Độ ưu tiên

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

Ví dụ - Mệnh đề WHERE (3)

BETWEEN

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

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

Ví dụ - Mệnh đề WHERE (4)

NOT BETWEEN

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

Ví dụ - Mệnh đề WHERE (5)

LIKE

SELECT MANV, TENNV

FROM NHANVIEN

WHERE DCHI LIKE 'Nguyễn _ _ _ _ '

SELECT MANV, TENNV

FROM NHANVIEN

WHERE DCHI LIKE 'Nguyễn %'

Ký tự bất kỳ

Chuỗi bất kỳ

Ví dụ - Mệnh đề WHERE (6)

NOT LIKE

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

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

Ví dụ - Mệnh đề WHERE (7)

ESCAPE

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



'Nguyen_'

Ví dụ - Mệnh đề WHERE (8)

Ngày giờ

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE NGSINH BETWEEN '1985-12-08' AND '1986-07-19'
```

'1985-12-08' YYYY-MM-DD

'12/08/1985' MM/DD/YYYY

'December 8, 1985'

'17:30:00' HH:MI:SS

'05:30 PM'

'1985-12-08 17:30:00'

Ví dụ - Mệnh đề WHERE (9)

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

Ví dụ - Mệnh đề WHERE (10)

NULL

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

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

Ví dụ - Mệnh đề FROM (1)

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
...	...

Ví dụ - Mệnh đề FROM (2)

Tên bí danh

```
SELECT TENPHG, DIADIEM
FROM PHONGBAN, AS DIBM, DPHG AS DD
WHERE MA_PHB=MA_DPHG MA_PHB=MA_DPHG
```

```
SELECT TENNV, NGSINH, TEN, TNGSINH
FROM NHANVIEN, NV, TN NHAN TN
WHERE MANV=MA_NVN
```

Mệnh đề ORDER BY (1)

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

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

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

Mệnh đề ORDER BY (2)

■ Ví dụ

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

MA_NVIEN	SODA
999887777	10
999887777	30
987987987	10
987987987	30
987654321	10
987654321	20
987654321	30

Một số dạng truy vấn khác

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

Truy vấn con ở mệnh đề FROM

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

SELECT <danh sách các cột>

FROM R1, R2, (<truy vấn con>) **AS** tên_bảng

WHERE <điều kiện>

Điều kiện kết ở mệnh đề FROM

■ Kết bằng

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

■ Kết ngoài

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

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
```

Phép toán tập hợp trong SQL (1)

- SQL có cài đặt các phép toán
 - Hợp (**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 (2)

■ 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 1- Phép toán tập hợp

- 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 MADA
FROM NHANVIEN, PHANCONG
WHERE MANV = MA_NVNIEN AND HONV = 'Nguyen'
UNION
SELECT MADA
FROM NHANVIEN, PHONGBAN, DEAN
WHERE MANV = TRPHP AND MAPHG=PHONG AND HONV = 'Nguyen'
```

Ví dụ 2- Phép toán tập hợp

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

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

~~INTERSET~~

```
SELECT TENTN, PHAI, MANV FROM THANNHAN
```

```
SELECT NV.*
```

```
FROM NHANVIEN NV, THANNHAN TN
```

```
WHERE NV.MANV = TN.MA_NVLEN
```

```
AND NV.TENNV = TN.TENNV AND NV.PHAI = TN.PHAI
```

Ví dụ 3 - Phép toán tập hợp

- Tìm những nhân viên không có thân nhân nào

```
SELECT MANV FROM NHANVIEN
```

~~EXCEPT~~

```
SELECT MA_NVIAN AS MANV FROM THANNHAN
```

```
SELECT MANV
```

```
FROM NHANVIEN
```

```
WHERE MANV NOT IN
```

```
(SELECT MANV FROM THANNHAN)
```

Truy vấn lồng (1)

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

Câu truy vấn ngoài
(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 trong
(Subquery)

Truy vấn lồng (2)

- 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 trong cùng một mệnh đề **WHERE** được kết hợp bằng phép nối logic
- Mệnh đề **WHERE** của câu truy vấn ngoài
 - <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 (3)

■ Có 2 loại truy vấn lồng

□ Lồng phân cấp

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

□ Lồng tương quan

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

Ví dụ 1- Lòng phân cấp

```
SELECT MANV, TENNV  
FROM NHANVIEN, DIADIEM_PHG  
WHERE DIADIEM='TP HCM' AND PHG=MAPHG
```

SELECT MANV, TENNV

FROM NHANVIEN

WHERE PHG IN (1, 5)

SELECT MAPHG

FROM DIADIEM_PHG

WHERE DIADIEM='TP HCM')

Ví dụ 2 - Lòng phân cấp

- Tìm những nhân viên không có thân nhân nào

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

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

Ví dụ 3 - Lòng phân cấp

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

```
SELECT *  
FROM NHANVIEN  
WHERE LUONG > ALL  
  (SELECT LUONG  
   FROM NHANVIEN  
   WHERE PHG = 4)
```

```
SELECT *  
FROM NHANVIEN  
WHERE LUONG >  
  (SELECT MAX(LUONG)  
   FROM NHANVIEN  
   WHERE PHG = 4)
```

Ví dụ 4 - Lòng phân cấp

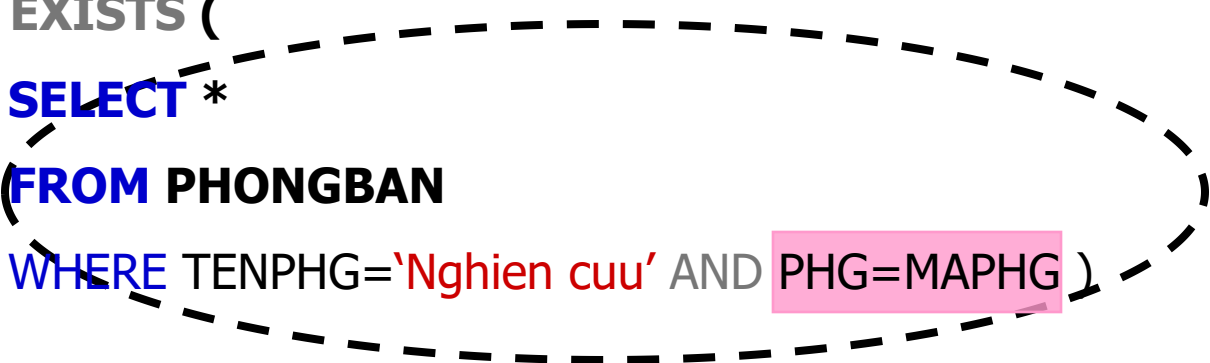
- Tìm những trưởng phòng có tối thiểu một thân nhân

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

Ví dụ 5- Lòng tương quan

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

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE EXISTS (  
    SELECT *  
    FROM PHONGBAN  
    WHERE TENPHG='Nghien cuu' AND PHG=MAPHG )
```



Ví dụ 6 - Lòong tương quan

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

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


Nhận xét IN và EXISTS

■ IN

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

■ 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 trong
- ❑ Những câu truy vấn có = ANY hay IN đều có thể chuyển thành câu truy vấn có EXISTS

Phép chia trong SQL (1)

R	A	B	C	D	E
	α	a	α	a	1
	α	a	γ	a	1
	α	a	γ	b	1
	β	a	γ	a	1
	β	a	γ	b	3
	γ	a	γ	a	1
	γ	a	γ	b	1
	γ	a	β	b	1

S	D	E
b_i	a	1
	b	1

$R \div S$	A	B	C
a_i	α	a	γ
	γ	a	γ

- $R \div S$ 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 (2)

- 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 - Phép chia

- 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_NVIEN, SODA)
 - Tập chia: DEAN(MADA)
 - Tập kết quả: KQ(MA_NVIEN)
 - Kết KQ với NHANVIEN để lấy ra TENNV

Ví dụ - Phép chia (tt)

```
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))
```

Ví dụ - Phép chia (tt)

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

Hàm kết hợp

■ COUNT

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

■ MIN

■ MAX

■ SUM

■ AVG

■ Các hàm kết hợp được đặt ở mệnh đề SELECT

Ví dụ 1 – Hàm kết hợp

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

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


Ví dụ 2 – Hàm kết hợp

- Cho biết số lượng nhân viên của phòng 'Nghien cuu'

```
SELECT COUNT(*) AS SL_NV  
FROM NHANVIEN, PHONGBAN  
WHERE PHG = MAPHG AND TENPHG = 'Nghien cuu'
```

Ví dụ 3 – Hàm kết hợp

- Cho biết số lượng nhân viên của từng phòng ban

PHG	SL_NV
5	3
4	3
1	1

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
453453453	Tran	Thanh	Tam	07/31/1972	543 MTL Q1	Nu	25000	333445555	5
999887777	Bui	Ngoc	Hang	07/19/1968	33 NTH Q1	Nu	38000	987654321	4
987654321	Le	Quynh	Nhu	07/20/1951	219 TD Q3	Nu	43000	888665555	4
987987987	Tran	Hong	Quang	04/08/1969	980 LHP Q5	Nam	25000	987654321	4
888665555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1

Gom nhóm

■ Cú pháp

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>

GROUP BY <danh sách các cột gom nhóm>

■ Sau khi gom nhóm

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

Ví dụ 1 – Gom nhóm

- Cho biết số lượng nhân viên của từng phòng ban

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

Ví dụ 2 – Gom nhóm

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

MA_NVIEN	SODA	THOIGIAN
123456789	1	32.5
123456789	2	7.5
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
453453453	1	20.0
453453453	2	20.0

Ví dụ 2 – Gom nhóm (tt)

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

Ví dụ 3 – Gom nhóm

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

MA_NVIEN	SODA	THOIGIAN
123456789	1	32.5
123456789	2	7.5
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
888665555	20	20.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
453453453	1	20.0
453453453	2	20.0

bị loại ra

Ví dụ 3 – Gom nhóm (tt)

```
SELECT MANV  
FROM PHANCONG  
GROUP BY MANV  
HAVING COUNT(*) >= 2
```


Điều kiện trên nhóm

■ 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 – Điều kiện gom nhóm

- Cho biết những phòng ban (TENPHG) có lương trung bình của các nhân viên lớn hơn 2tr

```
SELECT PHONG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN  
GROUP BY PHONG  
HAVING AVG(LUONG) > 2000000
```

```
SELECT TENPHG, AVG(LUONG) AS LUONG_TB  
FROM NHANVIEN, PHONGBAN  
WHERE PHG = MAPHG  
GROUP BY TENPHG  
HAVING AVG(LUONG) > 2000000
```

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

Nhận xét (tt)

- 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**

Bài tập - MỆNH ĐỀ GROUP BY

Quan hệ NV

nhóm

Q	S
a	10
a	2
b	9
b	5
c	10
c	8
c	6
c	4
c	10
d	16
d	18
d	50

Chia các dòng thành các nhóm dựa trên tập thuộc tính chia nhóm

Q	Count(S)
a	2
b	2
c	5
d	3

Tương tự cho các hàm SUM, MIN, MAX, AVG

Các thuộc tính GROUP BY: Q

Câu SQL:

```
Select Q, count(S)  
From NV  
Group by Q
```

Bài tập: Các hàm tính toán và gom nhóm (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

Bài tập: Các hàm tính toán và gom nhóm (2)

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

Bài tập (3)

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

Đáp án (1)

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
```

Đáp án (2)

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

Các hàm tính toán và gom nhóm (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

Đáp án (3)

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
```

Các hàm tính toán và gom nhóm (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