



# CƠ SỞ DỮ LIỆU

## CHƯƠNG 4: NGÔN NGỮ SQL

### (STRUCTURED QUERY LANGUAGE)

ThS. LÊ NGÔ THỰC VI



# MỤC TIÊU



1. **Nắm rõ khái niệm ngôn ngữ SQL, phân loại SQL, các thao tác dữ liệu trên một hệ quản trị cơ sở dữ liệu (DBMS) thông qua các phép toán đại số quan hệ.**
2. **Lập luận các giải pháp thực hiện một câu truy vấn SQL.**



# NỘI DUNG

1. Giới thiệu
2. Định nghĩa dữ liệu
3. Cập nhật dữ liệu
4. Truy vấn dữ liệu



# GIỚI THIỆU

1



# 1. Giới thiệu

- **SQL (Structured Query Language)**: Là ngôn ngữ chuẩn để truy vấn và thao tác trên CSDL quan hệ
- Được phát triển bởi IBM (1970s) gọi là **SEQUEL (Structured English Query Language)**
- Được ANSI và ISO công nhận và phát triển thành chuẩn
  - SQL86
  - SQL92
  - SQL99
  - SQL2003, SQL2008, SQL2011, SQL2016,...





# 1. Giới thiệu

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

Lý thuyết : Chuẩn SQL-99



# NGÔN NGỮ ĐỊNH NGHĨA DỮ LIỆU

2



## 2. Ngôn ngữ định nghĩa dữ liệu - DDL

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





## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh tạo bảng

- **CREATE TABLE** dùng để định nghĩa một bảng, gồm các thành phần
  - Tên bảng
  - Các thuộc tính
    - Tên thuộc tính
    - Kiểu dữ liệu
    - Các RBTV trên thuộc tính
- Cú pháp

```
CREATE TABLE <Tên_bảng>  
(  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],  
    <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],  
    ...  
    [<RBTV>]  
)
```



## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh tạo bảng

- Một số kiểu dữ liệu

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



## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh tạo bảng

Ví dụ: tạo bảng

```
CREATE TABLE NHANVIEN  
(  
    MANV    CHAR(9),  
    HONV    VARCHAR(10),  
    TENLOT  VARCHAR(20),  
    TENNV   VARCHAR(10),  
    NGSINH  DATETIME,  
    DCHI    VARCHAR(50),  
    PHAI    CHAR(3),  
    LUONG   INT,  
    MA_NQL  CHAR(9),  
    PHG     INT  
)
```

Cú pháp đơn giản nhất



## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh tạo bảng

- <RBTV>
  - NOT NULL
  - NULL
  - UNIQUE
  - DEFAULT
  - PRIMARY KEY
  - FOREIGN KEY / REFERENCES
  - CHECK
- Đặt tên cho RBTV

**CONSTRAINT** <Ten\_RBTV> <RBTV>



## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh tạo bảng

Ví dụ: tạo bảng có RBTV

```
CREATE TABLE NHANVIEN
(
    HONV  VARCHAR(10)  NOT NULL,
    TENLOT VARCHAR(20)  NOT NULL,
    TENNV  VARCHAR(10)  NOT NULL,
    MANV   CHAR(9)      PRIMARY KEY,
    NGSINH DATETIME,
    DCHI   VARCHAR(50),
    PHAI   CHAR(3)      CHECK (PHAI IN ('Nam', 'Nu')),
    LUONG  INT          DEFAULT (10000),
    MA_NQL CHAR(9),
    PHG    INT
)
```





## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh tạo bảng

**Ví dụ: tạo bảng có RBTV**

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

getdate() có chức năng lấy time ngay thời điểm nhập d liệu

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



## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh tạo bảng

Ví dụ: tạo bảng có RBTV – Đặt tên cho RBTV

```
CREATE TABLE NHANVIEN (  
    HONV    VARCHAR(10)    CONSTRAINT NV_HONV_NN NOT NULL,  
    TENLOT  VARCHAR(20)    NOT NULL,  
    TENNV   VARCHAR(10)    NOT NULL,  
    MANV    CHAR(9)        CONSTRAINT NV_MANV_PK PRIMARY KEY,  
    NGSINH  DATETIME,  
    DCHI    VARCHAR(50),  
    PHAI    CHAR(3)        CONSTRAINT NV_PHAICHK CHECK (PHAI IN ('Nam', 'Nu')),  
    LUONG   INT            CONSTRAINT NV_LUONG_DF DEFAULT (10000),  
    MA_NQL  CHAR(9),  
    PHG     INT  
)
```



## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh tạo bảng

Ví dụ: tạo bảng có RBTV – Đặt tên cho RBTV

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



## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh sửa bảng

- **ALTER TABLE** được dùng để:

- Thay đổi cấu trúc bảng
- Thay đổi RBTV

- Thêm thuộc tính: **ALTER TABLE** <Tên\_bảng> **ADD** <Tên\_cột> <Kiểu\_dữ\_liệu> [<RBTV>]

- Ví dụ: **ALTER TABLE NHANVIEN ADD NGHENGHIEP CHAR(20)**

- Xóa thuộc tính: **ALTER TABLE** <Tên\_bảng> **DROP COLUMN** <Tên\_cột>

- Ví dụ: **ALTER TABLE NHANVIEN DROP COLUMN NGHENGHIEP**

- Sửa kiểu dữ liệu: **ALTER TABLE** <Tên\_bảng> **ALTER COLUMN** <Tên\_cột> <Kiểu\_dữ\_liệu\_mới>

- Ví dụ: **ALTER TABLE NHANVIEN ALTER COLUMN NGHENGHIEP CHAR(50)**
- **Lưu ý:** Không phải sửa bất kỳ kiểu dữ liệu nào cũng được



## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh sửa bảng

- **ALTER TABLE** được dùng để:
  - Thay đổi cấu trúc bảng
  - Thay đổi RBTV

- Thêm RBTV:

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

- Xóa RBTV:

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

- **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





## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh sửa bảng

Ví dụ: Thêm RBTV

```
CREATE TABLE PHONGBAN (  
    TENPB    VARCHAR(20),  
    MAPHG    INT NOT NULL,  
    TRPHG    CHAR(9),  
    NG_NHANCHUC DATETIME  
)  
ALTER TABLE PHONGBAN ADD  
    CONSTRAINT PB_MAPHG_PK PRIMARY KEY (MAPHG),  
    CONSTRAINT PB_TRPHG FOREIGN KEY (TRPHG) REFERENCES NHANVIEN(MANV),  
    CONSTRAINT PB_NGNHANCHUC_DF DEFAULT (GETDATE()) FOR (NG_NHANCHUC),  
    CONSTRAINT PB_TENPB_UNI UNIQUE (TENPB)
```



## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh xóa bảng

- **DROP TABLE** được dùng để xóa cấu trúc bảng

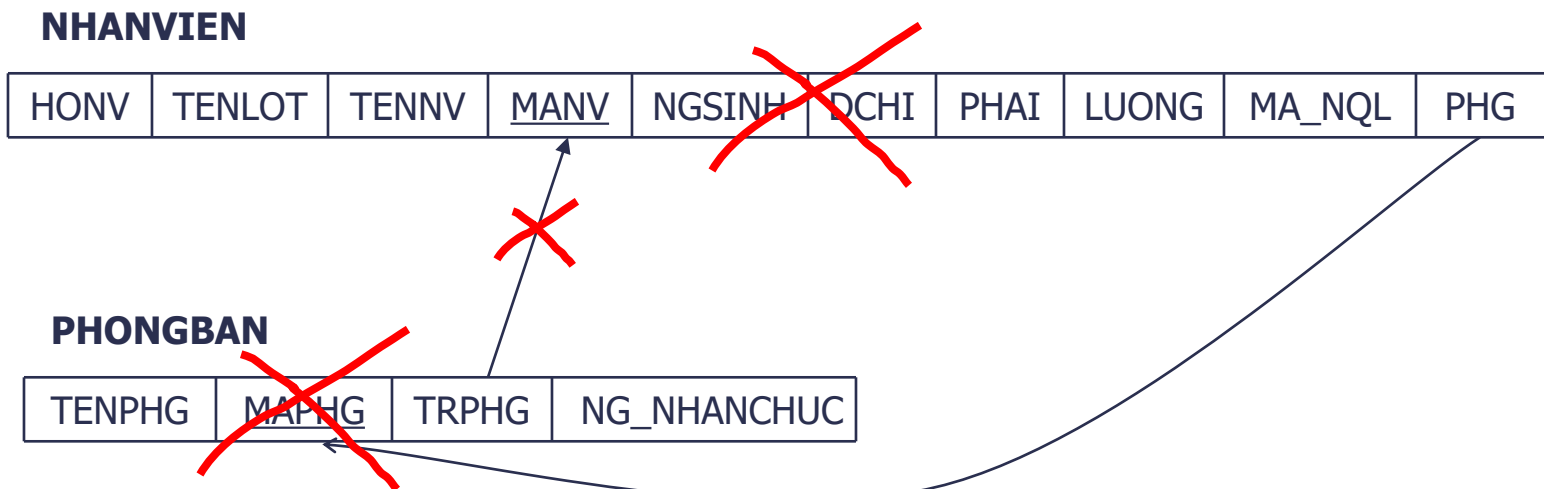
- Tất cả dữ liệu của bảng cũng bị xóa

- Cú pháp

**DROP TABLE** <Tên\_bảng>

- Ví dụ: **DROP TABLE** NHANVIEN

- **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





## 2. Ngôn ngữ định nghĩa dữ liệu – Lệnh tạo MGT

- **CREATE DOMAIN** được dùng để 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)



# NGÔN NGỮ CẬP NHẬT DỮ LIỆU

INSERT – UPDATE – DELETE

3



### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Insert

- **INSERT** được dùng để thêm 1 hay nhiều dòng vào bảng, bao gồm
  - Tên quan hệ
  - Danh sách các thuộc tính cần thêm dữ liệu
  - Danh sách các giá trị tương ứng

- **Cú pháp**

- **Thêm 1 dòng:**

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

- Ví dụ:

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

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

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





### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Insert

- **INSERT** được dùng để thêm 1 hay nhiều dòng vào bảng, bao gồm
  - Tên quan hệ
  - Danh sách các thuộc tính cần thêm dữ liệu
  - Danh sách các giá trị tương ứng
- Cú pháp
  - Thêm 1 dòng:

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



### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Insert

- **INSERT** được dùng để thêm 1 hay nhiều dòng vào bảng, bao gồm
  - Tên quan hệ
  - Danh sách các thuộc tính cần thêm dữ liệu
  - Danh sách các giá trị tương ứng
- Cú pháp
  - Thêm nhiều dòng:
  - Ví dụ:

**INSERT INTO** <tên bảng tạo trước>([<danh sách các thuộc tính>])  
<câu truy vấn từ bảng có sẵn>

```
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
```

- **Lưu ý:** số thuộc tính + kiểu dữ liệu của bảng tạo trước phải tương ứng thuộc tính trong mệnh đề Select của bảng có sẵn



### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Insert

- **INSERT** được dùng để thêm 1 hay nhiều dòng vào bảng, bao gồm
  - Tên quan hệ
  - Danh sách các thuộc tính cần thêm dữ liệu
  - Danh sách các giá trị tương ứng

- Cú pháp

- Thêm nhiều dòng:

**SELECT** <danh sách các thuộc tính> **INTO** <tên bảng mới>  
**FROM** <bảng có sẵn> [**WHERE** <điều kiện>]

- Ví dụ:

```
SELECT HONV, TENLOT, TENNV, MANV INTO NHANVIEN_NEW  
FROM NHANVIEN
```

- **Lưu ý:** bảng mới không cần tạo trước



### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Update

- **UPDATE** được dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng

- Cú pháp

```
UPDATE <tên bảng>  
SET      <tên thuộc tính 1>=<giá trị mới 1>,  
          <tên thuộc tính 2>=<giá trị mới 2>,  
          ...  
[WHERE <điều kiện>]
```

- Ví dụ:

```
UPDATE  NHANVIEN  
SET     NGSINH='1965/08/12'  
WHERE   MANV='333445555'
```

```
UPDATE  NHANVIEN  
SET     LUONG=LUONG*1.1
```



### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Update

- **UPDATE** được dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng
- Nhận xét:
  - Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ được 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
- Ví dụ: Với đề án có mã số 10, hãy thay đổi nơi thực hiện đề án thành 'Vũng Tàu' và phòng ban phụ trách là phòng 5

```
UPDATE DEAN  
SET DIADIEM_DA=N'Vũng Tàu', PHONG=5  
WHERE MADA=10
```

có 2 cách x lí:  
+ 1 là thay i  
+ 2 là on update cascade





### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Delete

- **DELETE** được dùng để xóa các dòng của bảng

- Cú pháp

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

- Ví dụ: **DELETE FROM** NHANVIEN  
**WHERE** HONV= **N'Trần'**

**DELETE FROM** NHANVIEN

- Ví dụ: Xóa đi những nhân viên ở phòng Nghiên cứu

**DELETE FROM** NHANVIEN

**WHERE** PHG IN (**SELECT** MAPHG **FROM** PHONGBAN **WHERE** TENPHG= **N'Nghiên cứu'**)



### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Delete

- **DELETE** được dùng để xóa các dòng của bảng
- Cú pháp

**DELETE FROM** <tên bảng>  
[**WHERE** <điều kiện>]
- Nhận xét:
  - Số lượng số dòng bị xóa phụ thuộc vào điều kiện ở mệnh đề WHERE
  - Nếu không chỉ định điều kiện ở mệnh đề WHERE, tất cả các dòng trong bảng sẽ bị xóa
  - Lệnh DELETE có thể gây ra vi phạm RB tham chiếu
    - Không cho xóa
    - Xóa luôn những dòng có giá trị đang tham chiếu đến: CASCADE
    - Đặt NULL cho những giá trị tham chiếu



### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Delete

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_NVIENT	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

xóa toàn b



### 3. Ngôn ngữ cập nhật dữ liệu – Lệnh Delete

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

xóa phg

MANV	HONV	TENLOT	TENNV	NGSINH	DCHI	PHAI	LUONG	MA_NQL	PHG
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
888665555	Pham	Van	Vinh	11/10/1945	450 TV HN	Nam	55000	NULL	1



# NGÔN NGỮ' TRUY VẤN DỮ' LIỆU

4



## 4. NGÔN NGỮ TRUY VẤN DỮ LIỆU SQL

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



 + 




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



## 4. TRUY VẤN CƠ BẢN

- Câu truy vấn cơ bản gồm 3 mệnh đề

**SELECT** [DISTINCT/ALL] <danh sách các cột> | hàm

**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,.....



## 4. TRUY VẤN CƠ BẢN

### ▪ SQL vs. ĐSQH

pi

kt

o

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

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

#### Ví dụ 1

Lấy tất cả các cột của  
quan hệ kết quả

**SELECT** \*  
**FROM** NHANVIEN  
**WHERE** PHG=5 and PHAI = 'Nam'

$\sigma_{PHG=5 \wedge PHAI='Nam'}(NHANVIEN)$





# 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ SELECT

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

$\pi_{MANV, HONV, TENLOT, TENNV}(\sigma_{PHG=5 \wedge PHAI='Nam'}(NHANVIEN))$

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

Tên bí danh

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

hoc N'tên lót'

$\rho_{MANV, HO, TEN LOT, TEN}(\pi_{MANV, HONV, TENLOT, TENNV}(\sigma_{PHG=5 \wedge PHAI='Nam'}(NHANVIEN)))$

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



# 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ SELECT

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

$$\rho_{MANV,HO\ TEN}(\pi_{MANV,HONV+TENLOT+TENN\cancel{V}}(\sigma_{PHG=5\wedge PHAI='Nam'}(NHANVIEN))))$$

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

MANV	LUONG10%
333445555	33000
987987987	27500

$$\rho_{MANV,LUONG10\%}(\pi_{MANV,LUONG*1.1}(\sigma_{PHG=5\wedge PHAI='Nam'}(NHANVIEN))))$$



## 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ SELECT

- Loại bỏ các dòng trùng nhau

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

LUONG
30000
25000
28000
38000

- Tồn chi phí
- Người dùng muốn thấy



## 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ WHERE

- Ví dụ 2: Cho biết MANV và TENNV làm việc ở phòng ‘Nghien cuu’

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

$KQ \leftarrow \pi_{\text{MANV}, \text{TENNV}} (\sigma_{\text{TENPHG}=\text{'Nghien cuu'}}(R1))$

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

Biểu thức luận lý

TRUE TRUE



## 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ WHERE

### BETWEEN

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

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

### NOT BETWEEN

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

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



## 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ WHERE

**NGÀY THÁNG**

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE NG SINH BETWEEN '1955-12-08' AND '1966-07-19'
```

**ĐỘ ƯU TIÊN**

Cho biết MANV và TENNV làm việc ở phòng 'Nghiên cứu' hoặc phòng 'Quan lý'

```
SELECT MANV, TENNV  
FROM NHANVIEN, PHONGBAN  
WHERE (TENPHG='Nghiên cứu' OR TENPHG='Quan lý') AND PHG=MAPHG
```



## 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ WHERE

### LIKE

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE 'Nguyen _ _ _ _ _'  
/   
Ký tự bất kỳ
```

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE DCHI LIKE 'Nguyen %'  
/   
Chuỗi bất kỳ
```

### NOT LIKE

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

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

### ESCAPE

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



## 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ WHERE

### NULL

- Sử dụng trong trường hợp
  - Không biết (unknown value)
  - Không thể áp dụng (inapplicable value/attribute)
  - Giá trị bị che giấu (withheld value)
- 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 có giá trị là NULL
  - $x = 3$  cho ra kết quả là UNKNOWN
  - $x = 3$  là một so sánh không hợp lệ trong SQL

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

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





## 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ WHERE

- Không sử dụng mệnh đề Where

SELECT MANV, PHG

FROM NHANVIEN

mc nh

WHERE TRUE

MANV	PHG
333445555	1
333445555	4
333445555	5
987987987	1
987987987	4
987987987	5
...	...



## 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ FROM

Tên bí danh

```
SELECT TENPHG, DIADIEM
FROM PHONGBAN, AS DIADIEM_PHG AS DD
WHERE MA_PHG=MA_DIADIEM
```

t có AS học cha không trng

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



## 4. TRUY VẤN CƠ BẢN – MỆNH ĐỀ ORDER BY

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

- **Cú pháp**

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

- ASC: tăng (mặc định)
- DESC: giảm

```
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



## 4. TRUY VẤN CƠ BẢN – BÀI TẬP

**VD 3.** 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. Kết quả xuất theo mã đề án tăng dần.

**VD 4.** 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

**VD 5.** Tìm họ tên của từng nhân viên và người phụ trách trực tiếp nhân viên đó

**VD 6.** Tìm họ tên của những nhân viên được “Nguyen Thanh Tung” phụ trách trực tiếp