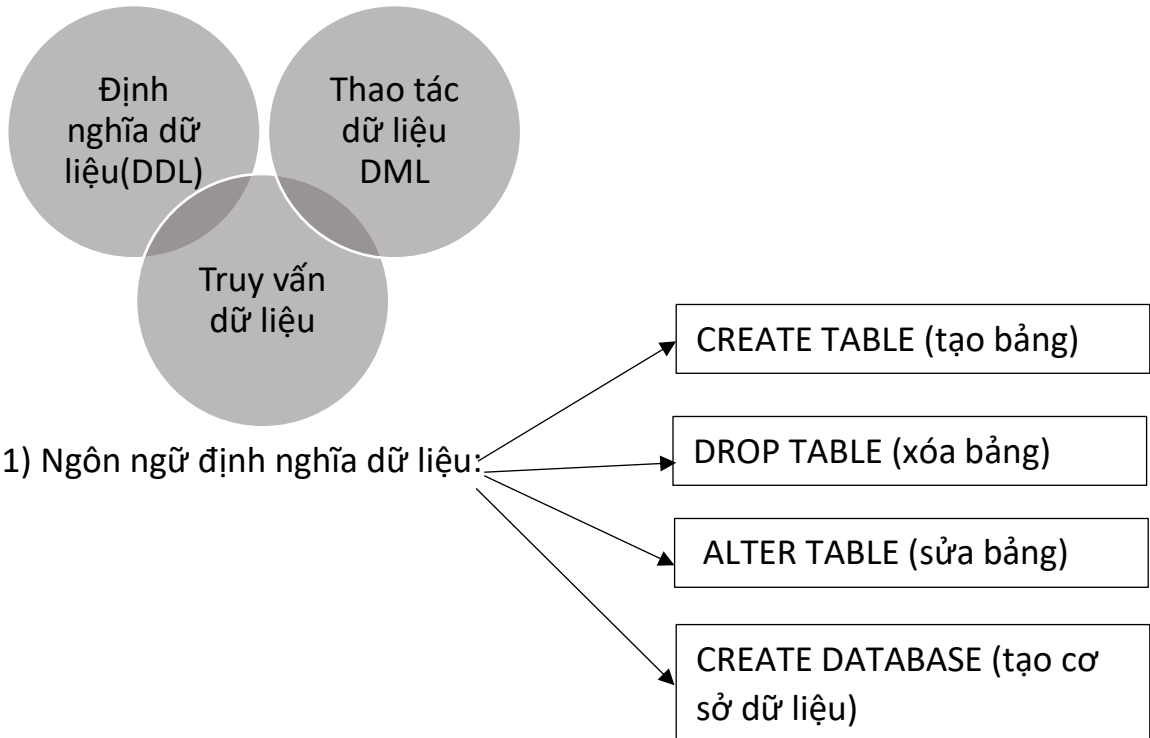


## Chương 4(tt)

### Ngôn ngữ SQL (ngôn ngữ truy vấn dữ liệu)

#### 1. Giới thiệu:

SQL gồm:



### TẠO BẢNG

```
CREATE TABLE < Tên bảng>(  
<Tên cột><Kiểu dữ liệu>[<Ràng buộc toàn vẹn>],  
<Tên cột><Kiểu dữ liệu>[<Ràng buộc toàn vẹn>],  
....  
[<Ràng buộc toàn vẹn>]  
)
```



## KHAI BÁO Ràng BUỘC TOÀN VỆN <RBTV>

- NOT NULL
- NULL
- UNIQUE tên thuộc tính
- DEFAULT (giá trị mặc định)
- PRIMARY KEY (các thuộc tính khóa chính)
- FOREIGN KEY / REFERENCES
- CHECK (tên thuộc tính điều kiện)

!!!Đặt tên cho RBTV

**CONSTRAINT** <Tên RBTV><RBTV>

**Ví dụ:**

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

## SỬA BẢNG

- Thêm cột:

**ALTER TABLE** <Tên bảng>

**ADD** <Tên cột><Kiểu dữ liệu>



- Xóa cột:

```
ALTER TABLE <Tên bảng>  
DROP COLUMN <Tên cột><Kiểu dữ liệu>
```

- Sửa cột:

```
ALTER TABLE <Tên bảng>  
ALTER COLUMN <Tên cột><Kiểu dữ liệu mới>
```

- Thêm RBTV:

```
ALTER TABLE <Tên bảng> ADD  
CONSTRAINT<Tên RBTV><RBTV>,  
CONSTRAINT<Tên RBTV><RBTV>,  
....
```

- Xóa RBTV:

```
ALTER TABLE <Tên bảng>  
DROP <Tên RBTV>
```

**Ví dụ:**

```
ALTER TABLE NHANVIEN ADD NGHENGHIEP CHAR(20)
```

```
ALTER TABLE NHANVIEN ALTER COLUMN NGHENGHIEP CHAR(50)
```

```
ALTER TABLE NHANVIEN DROP COLUMN NGHENGHIEP
```

## XÓA BẢNG

```
DROP TABLE <Tên bảng>
```

### 2. Truy vấn cơ bản:

```
SELECT <danh sách các cột>  
FROM<danh sách các bảng>  
[WHERE<Điều kiện>]
```



- <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

## MỆNH ĐỀ SELECT

- Loại bỏ các dòng trùng nhau: Sử dụng DISTINCT trong mệnh đề SELECT  
Ví dụ: SELECT DISTINCT LUONG FROM NHANVIEN WHERE PHG=5
- Tên bí danh:  
Ví dụ: SELECT MANV, HONV AS HO, TENLOT AS 'TEN LOT', TENNV AS TEN FROM NHANVIEN WHERE PHG=5 AND PHAI='Nam'
- Có thể chứa các biểu thức toán học liên quan tới +, -, \*, / và thực hiện trên các hằng số hay thuộc tính của bộ  
Ví dụ: SELECT MANV, LUONG\*1.1 AS 'LUONG10%' FROM NHANVIEN WHERE PHG=5 AND PHAI='Nam'

## MỆNH ĐỀ WHERE

- Mô tả các điều kiện mà kết quả phải thỏa
- Điều kiện kết hợp với nhau bởi AND, OR hay NOT
- Các so sánh có thể được áp dụng trên kết quả của các biểu thức toán học  
**Ví dụ:** Tìm tất cả nhân viên có lương tối thiểu là 20000 và lương tối đa là 45000  
**C1:** SELECT \*FROM NHANVIEN WHERE LUONG>=20000 AND LUONG<=45000  
**C2:** SELECT \* FROM NHANVIEN WHERE LUONG BETWEEN 20000 AND 45000
- Độ ưu tiên  
**Ví dụ:**  
SELECT MANV, TENNV  
FROM NHANVIEN, PHONGBAN  
WHERE(TENPHG='Nghien cuu' OR TENPHG='Quan ly')AND PHG=MAPHG
- So sánh chuỗi
  - + Sử dụng bất kỳ phép so sánh nào chẳng hạn , =, ...



+ Để so sánh gần đúng sử dụng LIKE :

- '%' thay thế chuỗi ký tự bất kỳ
- '\_' thay thế 1 ký tự bất kỳ

**Ví dụ:** a) Cho biết tên các nhân viên bắt đầu bằng chữ 'A'

```
SELECT TENNV FROM NHANVIEN WHERE TENNV LIKE 'A%'
```

b) Cho biết tên các nhân viên bắt đầu bằng chữ 'N' và có bốn ký tự

```
- SELECT TENNV FROM NHANVIEN WHERE TENNV LIKE 'A%'
```

```
- SELECT TENNV FROM NHANVIEN WHERE TENNV LIKE 'N____'
```

- Giá trị NULL
- Các bộ có thể có giá trị NULL ở một số thuộc tính
- IS NULL (IS NOT NULL) được dùng để kiểm tra giá trị null

**Ví dụ:**

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

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

- Kết quả của biểu thức toán học nào liên quan tới null là null

**Ví dụ:** 9 + null trả về null

- Các hàm tính toán bỏ qua các giá trị null
- Sử dụng IN để kiểm tra một giá trị thuộc tính có thuộc một tập giá trị cho trước không

**Ví dụ:** Cho biết tên các nhân viên thuộc một trong các phòng P1, P2, P3

```
SELECT TENNV FROM NHANVIEN WHERE PHG IN ('P1', 'P2', 'P3')
```

## MỆNH ĐỀ FROM

- Nhiều bảng có thể được truy vấn trong một lệnh SQL bằng cách liệt kê chúng trong mệnh đề FROM

**Ví dụ:** SELECT\*

```
FROM PHONGBAN, DEAN
```

```
WHERE PHONGBAN.MAPHG = DEAN.PHONG
```

**3. Truy vấn sử dụng phép toán tập hợp:**

```
SQL1 UNION [ALL] SQL2
```

```
SQL1 INTERSECT [ALL] SQL2
```

```
SQL1 EXCEPT [ALL] SQL2
```



**ALL: yêu cần giữ lại các dòng trùng**

**Ví dụ:** Cho biết mã nhân viên tham gia dự án với nhiệm vụ là Quan lý hay Tu van

**C1:**

```
SELECT DISTINCT MANV
FROM PHANCONG
WHERE NVU='Quan lý'
OR NVU='Tu van'
```

**C2:**

```
SELECT DISTINCT MANV
FROM PHANCONG WHERE
NVU='Quan lý'
UNION
SELECT DISTINCT MANV
FROM PHANCONG WHERE
NVU='Tu van'
```

**Ví dụ:** Cho biết mã nhân viên tham gia dự án với nhiệm vụ là Quan lý và Tu van

**C1:**

```
SELECT DISTINCT MANV
FROM PHANCONG WHERE
NVU='Quan lý'
INTERSECT
SELECT DISTINCT MANV
FROM PHANCONG WHERE
NVU='Tu van'
```

**C2:**

```
SELECT DISTINCT PC1.MANV
FROM PHANCONG PC1,
PHANCONG PC2
WHERE PC1.MANV=PC2.MANV
AND PC1.NVU='Quan lý'
AND PC2.NVU='Tu van'
```

**Ví dụ:** Cho biết mã nhân viên không tham gia dự án

```
SELECT DISTINCT MANV
FROM NHANVIEN
EXCEPT
SELECT DISTINCT MANV
FROM PHANCONG
```

## MỆNH ĐỀ ORDER BY

- Dùng để hiển thị kết quả câu truy vấn theo một thứ tự nào đó
- **Cú pháp:**
- SELECT <danh sách các cột>  
FROM <danh sách các bảng>  
WHERE <Điều kiện>  
ORDER BY <danh sách các cột>
- Nếu muốn thứ tự tăng dần, sử dụng ASC (mặc định)
- Nếu muốn thứ tự giảm dần, sử dụng DESC

**Ví dụ:** Tìm tất cả mã nhân viên và tên nhân viên chức vụ LT, sắp xếp theo phòng giảm dần và sau đó là ngày sinh tăng dần

```
SELECT MANV, TENNV FROM NHANVIEN WHERE NVU='LT'  
ORDER BY PHG DESC, NGSINH ASC
```

### 4. Truy vấn lồng:

```
SELECT [DISTINCT] <Danh sách các thuộc tính / hàm >  
FROM <Danh sách các bảng>  
WHERE <Điều kiện> (Câu truy vấn con)
```

- **<Điều kiện>:** Truy vấn con trả về giá trị tập hợp
  - <biểu thức> [NOT] IN (<truy vấn con>)
  - <biểu thức> <phép toán so sánh> ANY (<truy vấn con>)
  - <biểu thức> <phép toán so sánh> ALL (<truy vấn con>)
- **<Điều kiện>:** Kiểm tra sự tồn tại trong kết quả của truy vấn con
  - [NOT] EXISTS (<truy vấn con>):
    - Trả về True nếu có ít nhất một bộ trong truy vấn con
    - Trả về False nếu ngược lại.

 Ví dụ: **Liệt kê các mã nhân viên thực hiện cả 2 đề án DA01 và DA02**

Cách 1:



```
SELECT MaNV
FROM PHANCONG
WHERE MaDA = 'DA01' AND MaNV IN (SELECT MaNV
                                  FROM PHANCONG
                                  WHERE MaDA = 'DA02')
```

Cách 2:

```
SELECT MaNV
FROM PHANCONG
WHERE MaDA = 'DA01' AND EXISTS (SELECT *
                                FROM PHANCONG
                                WHERE MaDA = 'DA02' AND
                                PC_DA2.MaNV = PC_DA1.MaNV)
```

✚ Ví dụ: Tìm nhân viên có lương cao hơn lương của tất cả nhân viên thuộc phòng Kế toán

```
SELECT MaNV, HoTen
FROM NHANVIEN
WHERE Luong > ALL (SELECT Luong
                  FROM NHANVIEN NV, PHONGBAN PB
                  WHERE NV.Phong = PB.MaPH AND TenPH = 'Ke Toan')
```

##### 5. Truy vấn sử dụng các hàm tính toán trên nhóm:

- Sử dụng các hàm tính toán, có thể có gom nhóm dữ liệu và tính toán trên nhóm
- Các hàm có đầu vào là một tập giá trị và trả về một giá trị đơn: Min(), Max(), Avg(), Sum(), Count()

```
SELECT [DISTINCT] <Danh sách các thuộc tính / hàm >
FROM <Danh sách các bảng>
[WHERE <Điều kiện> ]
[GROUP BY <Danh sách thuộc tính gom nhóm> ]
[HAVING <Điều kiện trên nhóm> ]
[ORDER BY {<Thuộc tính sắp xếp> ASC / DESC }]
```





Ví dụ: Tìm phòng ban có số lượng nhân viên lớn hơn 10. Lấy ra thông tin mã phòng, tên phòng, số lượng nhân viên. Sắp xếp theo số lượng nhân viên giảm dần.

```
SELECT PB.MaPH, TenPH, Count (MaNV) 'SLNV'  
FROM NHANVIEN NV, PHONGBAN PB  
WHERE NV.Phong = PB.MaPH  
GROUP BY MaPH, TenPH  
HAVING Count (MaNV) > 10  
ORDER BY Count (MaNV) DESC
```

#### 6. Truy vấn sử dụng kết ngoài:

- Sử dụng kết ngoài trong một số trường hợp để hạn chế việc bỏ sót dữ liệu
- Cách thức sử dụng tương tự inner join trong kết bằng.

- Phép kết phải:

Table1 **left outer join** Table2 on <điều kiện kết>

- Phép kết phải:

Table1 **right outer join** Table2 on <điều kiện kết>

- Phép kết đầy đủ:

Table1 **full outer join** Table2 on <điều kiện kết>



Ví dụ: Cho biết những mã nhân viên không tham gia đề án nào

```
SELECT NV.MaNV, HoTen  
FROM NHANVIEN NV left outer join PHANCONG PC on NV.MaNV=PC.MaNV  
WHERE MaDA IS NULL
```

