



SQL SERVER 2012

--- Môn 1 ---

Bài 4: SQL - Structured Query Language

1. Các câu lệnh truy vấn dữ liệu
2. Các câu lệnh cập nhật dữ liệu
3. Các chức năng nâng cao

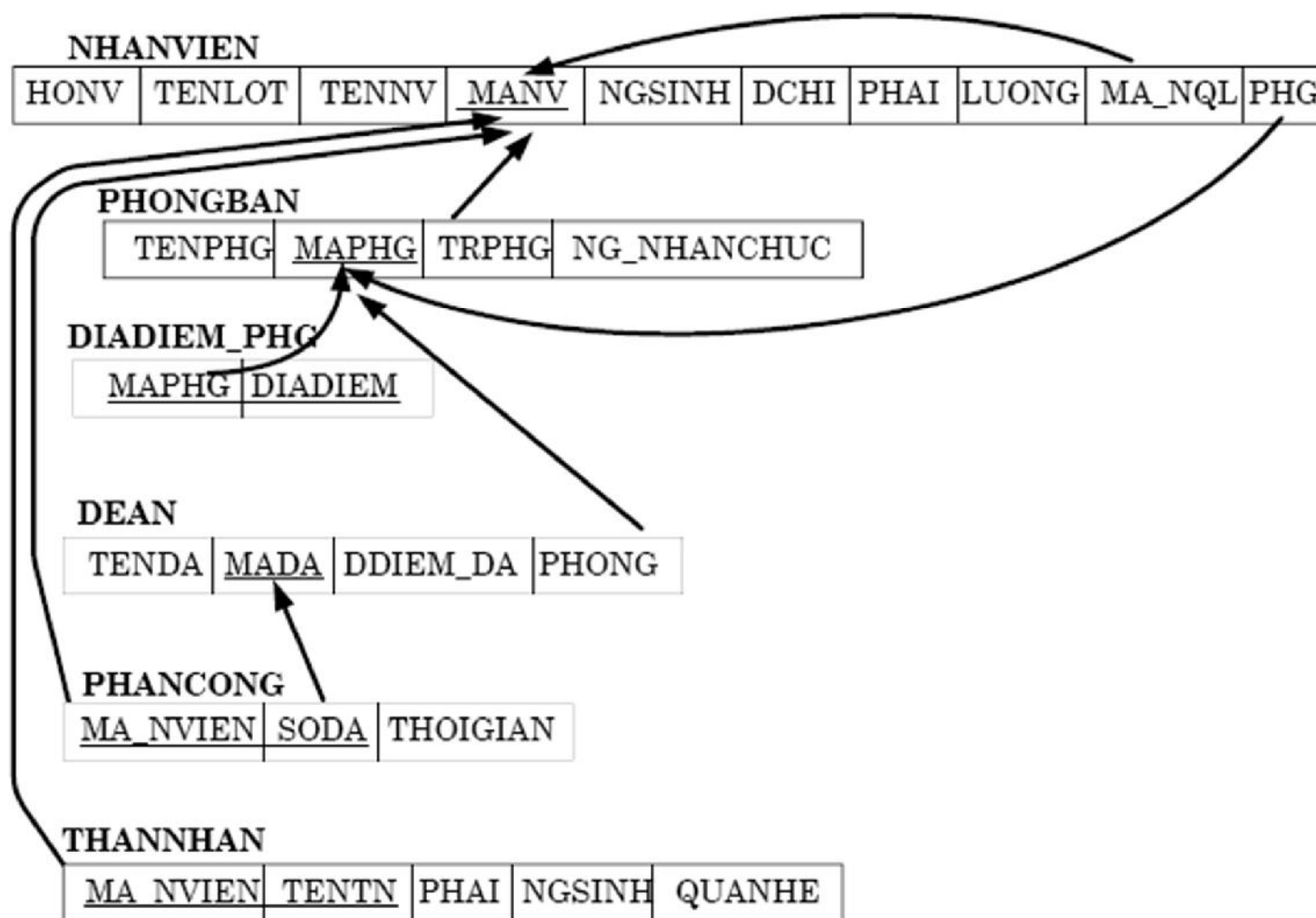
Bài 4.1: Các câu lệnh truy vấn dữ liệu

1. Lệnh truy vấn dữ liệu
2. Giới hạn dòng và sắp xếp dữ liệu
3. Truy vấn dữ liệu từ nhiều bảng
4. Sử dụng biểu thức CASE trong truy vấn
5. Gom nhóm dữ liệu
6. Truy vấn con
7. Tích hợp kết quả từ nhiều câu truy vấn
8. Phần mở rộng của mệnh đề GROUP BY
9. Một số mệnh đề khác trong SELECT

6. Truy vấn con

- Truy vấn con ?
- Cú pháp truy vấn con
- Truy vấn con – WHERE & HAVING
- Truy vấn con – FROM
- Truy vấn con kết hợp
- Truy vấn con – Cập nhật dữ liệu
- Một số lưu ý

Cơ sở dữ liệu minh họa



Truy vấn con

1. Cho biết các nhân viên của phòng nghiên cứu có tham gia đề án
2. Cho biết các đề án mà nhân viên 'Vũ' tham gia với thời gian trên 20 giờ
3. Liệt kê các phòng có ít hơn 3 nữ nhân viên (MaPH, TenPH)
4. Tìm tên của các nhân viên làm việc cho tất cả các đề án mà phòng 5 chủ trì.
5. Danh sách những nhân viên (HONV, TENNV) không có thân nhân nào.
6. Danh sách những nhân viên (HONV, TENNV) có cùng tên với người thân.
7. Liệt kê danh sách nhân viên có lương cao hơn lương trung bình
8. Cho biết phòng nào có đông nhân viên nhất

Truy vấn con

```
SELECT ...  
FROM ...  
WHERE ...
```

```
(SELECT ...  
FROM ...  
WHERE ...)
```

Cú pháp truy vấn con

```
SELECT    ds_cot
FROM      bang
WHERE     bieu_thuc toan_tu
          (SELECT      bieu_thuc
           FROM        bang)
```

- Truy vấn con thi hành một lần trước truy vấn chính (truy vấn cha)
- Kết quả trả về từ truy vấn con được sử dụng trong câu truy vấn chính.

Một số quy tắc khi sử dụng truy vấn con

- Câu truy vấn con phải nằm trong cặp ngoặc đơn
- Đặt truy vấn con bên phải điều kiện so sánh
- Sử dụng các toán tử một dòng với các truy vấn con trả về một dòng và sử dụng các toán tử nhiều dòng với các truy vấn con trả về nhiều dòng.

Các dạng truy vấn con

- Truy vấn con một dòng



- Truy vấn con nhiều dòng



Các dạng truy vấn con

- Truy vấn con một dòng
 - Trả về duy nhất 1 dòng
 - Sử dụng các toán tử so sánh một dòng

Toán tử	Ý nghĩa
=	Bằng
>	Lớn hơn
>=	Lớn hơn hoặc bằng
<	Nhỏ hơn
<=	Nhỏ hơn hoặc bằng
<>	Không bằng

Câu lệnh sau sai ở chỗ nào?

```
SELECT manhanvien, tennv  
FROM qlns.nhanvien  
WHERE mucluong =  
      (SELECT MIN(mucluong)  
       FROM qlns.nhanvien  
       GROUP BY maphong)
```

→ Toán tử một dòng được sử dụng với truy vấn con nhiều dòng

Các dạng truy vấn con

- Truy vấn con nhiều dòng
 - Trả về nhiều dòng
 - Sử dụng các toán tử so sánh nhiều dòng

Toán tử	Ý nghĩa
IN	Bằng một trong các giá trị
ANY	Chỉ cần thỏa một trong các giá trị trả về bởi truy vấn con
ALL	Phải thỏa tất cả các giá trị trả về bởi truy vấn con

Các dạng truy vấn con

- Truy vấn con nhiều dòng
 - Sử dụng toán tử ALL

```
SELECT manhanvien, tennv, macongviec, mucluong
FROM qlns.nhanvien
WHERE mucluong < ALL
      (SELECT mucluong
       FROM qlns.nhanvien
       WHERE macongviec = 'TPHONG')
AND macongviec <> 'TPHONG';
```

9.000.000, 3.000.000

	manhanvien	tennv	macongviec	mucluong
1	124	Tôi	KTTM	1500000
2	141	Hùng	TTHI	1500000
3	142	Sơn	BHANG	1200000
4	149	Hạnh	VPHONG	2500000

Thực hành

- Bài toán: Quản lý kết quả học tập
 - SINHVIEN(MSSV, Ho, Ten, Ngành, Ngaysinh, Phai, Diachi, DTB)
 - MONHOC (MaMH, TenMH, Khoa, SoTC)
 - KETQUA(MSSV, MaMH, Diem)
- Yêu cầu:
 1. Cho biết các SV sống ở Quận 1 học chung Ngành với SV “Nguyễn Đức Thịnh”.
 2. Liệt kê danh sách SV khoa CNTT học môn “Toán cao cấp” có điểm <5
 3. Cho biết Ngành có số SV ít nhất.
 4. Tính lại điểm trung bình của sinh viên.

Phần mở rộng của Truy vấn con

- Truy vấn con trả về nhiều cột
- Truy vấn con trong mệnh đề FROM
- Biểu thức truy vấn con đơn trị (scalar subquery)
- Viết truy vấn con kết hợp (Correlated subquery)
- Sử dụng hàm **EXISTS**

Truy vấn con trong mệnh đề FROM

```
SELECT  a.tennv, a.mucluong,  
        a.maphong, b.tb_luong  
FROM    qlns.nhanvien a, (SELECT  maphong,  
                                AVG(mucluong) tb_luong  
                                FROM    qlns.nhanvien  
                                GROUP BY maphong) b  
WHERE   a.maphong = b.maphong  
AND     a.mucluong > b.tb_luong;
```

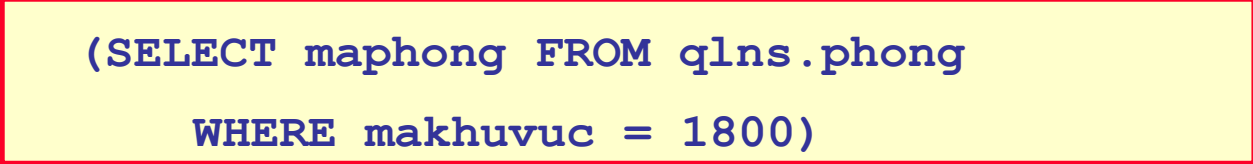
	tennv	mucluong	maphong	tb_luong
1	Huyền	5000000	20	4125000.000000
2	Học	7000000	20	4125000.000000
3	Thùy	10000000	60	6500000.000000
4	Toàn	4000000	80	2600000.000000
5	Khoài	9000000	110	5250000.000000
6	Hồng	6000000	190	4250000.000000

Biểu thức truy vấn con đơn trị

- Biểu thức truy vấn con đơn trị (A scalar subquery expression) là một truy vấn con trả về duy nhất một cột và một dòng
- Truy vấn con đơn trị có thể sử dụng:
 - Trong các điều kiện và biểu thức của CASE
 - Trong tất cả các mệnh đề của SELECT ngoại trừ GROUP BY

Biểu thức truy vấn con đơn trị

- Trong biểu thức **CASE**

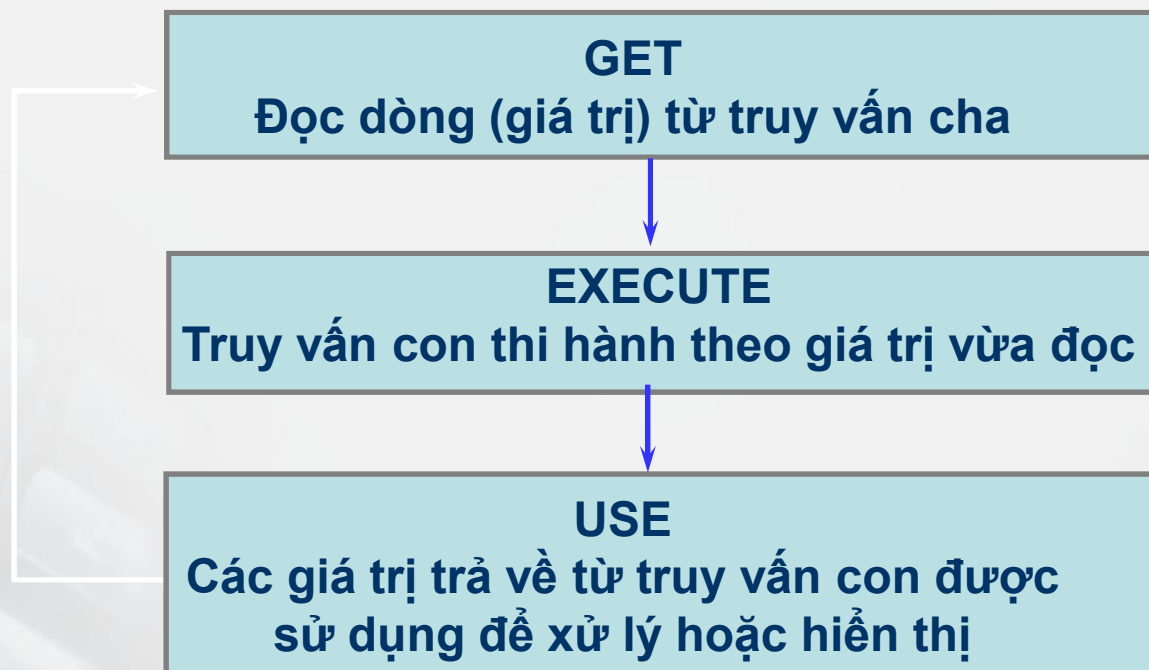
```
SELECT manhanvien, tennv,  
      (CASE  
        WHEN maphong = 20  
        THEN N'Sài gòn' ELSE N'Hà nội' END) diaban  
FROM   qlns.nhanvien
```

- Trong mệnh đề **ORDER BY**

```
SELECT   manhanvien, tennv  
FROM     qlns.nhanvien nv  
ORDER BY (SELECT tenphong  
           FROM qlns.phong ph  
           WHERE nv.maphong = ph.maphong)
```

Truy vấn con kết hợp

- Truy vấn con kết hợp (Correlated subqueries) được sử dụng cho việc xử lý từng dòng. Mỗi truy vấn con sẽ được thi hành một lần ứng với mỗi dòng của truy vấn cha



Truy vấn con kết hợp

```
SELECT cot1, cot2, ...  
FROM bang1 bang_cha  
WHERE cot1 toan_tu  
      (SELECT cot1, cot2  
        FROM bang2  
        WHERE bieu_thuc1 =  
              bang_cha.bieu_thuc2)
```

Câu truy vấn con liên kết với cột từ bảng của truy vấn cha

Sử dụng hàm EXISTS

- Hàm **EXISTS** kiểm tra tồn tại kết quả trả về từ câu truy vấn con.

Sử dụng hàm EXISTS

- Ví dụ: Tìm những nhân viên có quản lý ít nhất một người

```
SELECT manhanvien, tennv, macongviec, maphong
FROM qlns.nhanvien BangCha
WHERE EXISTS ( SELECT 'X'
                FROM qlns.nhanvien
                WHERE manguoiquanly =
                  BangCha.manhanvien)
```

Thực hành

- Tạo truy vấn con nhiều cột
- Viết truy vấn con kết hợp
- Sử dụng truy vấn con đơn trị
- Sử dụng hàm EXISTS

Hết