

Chương 4

Ngôn ngữ truy vấn SQL (tt)

TS. Nông Thị Hoa

Mobile: 0949.276.430

Email: nongthihoa@duytan.edu.vn

NỘI DUNG CHI TIẾT

4.3. Truy vấn dữ liệu

4.3.2 Tập hợp, so sánh tập hợp

4.3.3 Truy vấn lồng

4.3.4 Hàm kết hợp và gom nhóm

4.3.2 PHÉP TOÁN TẬP HỢP TRONG SQL

- 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

4.3.2 PHÉP TOÁN TẬP HỢP TRONG SQL (TT)

- 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>
```

4.3.2 PHÉP TOÁN TẬP HỢP TRONG SQL (TT)

- Ví dụ 1: 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 SODA FROM NHANVIEN, PHANCONG
```

```
WHERE MANV=MA_NVN AND HONV='Nguyen'
```

```
UNION
```

```
SELECT MADA FROM NHANVIEN, PHONGBAN, DEAN
```

```
WHERE MANV=TRPHG AND MAPHG=PHONG
```

```
AND HONV='Nguyen'
```

4.3.2 PHÉP TOÁN TẬP HỢP TRONG SQL (TT)

- Ví dụ 2: Tìm nhân viên có người thân cùng tên và cùng giới tính

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

```
INTERSECT
```

```
SELECT TENTN, PHAI, MA_NVNEN FROM THANNHAN
```

4.3.2 PHÉP TOÁN TẬP HỢP TRONG SQL (TT)

- Ví dụ 3: Tìm nhân viên không có thân nhân nào

```
SELECT MANV FROM NHANVIEN
```

```
EXCEPT
```

```
SELECT MA_NVNEN AS MANV FROM THANNNHAN
```

4.3.3 TRUY VẤN LỒNG

Câu truy vấn cha
(Outer query)

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <so sánh tập hợp> (

Câu truy vấn con
(Subquery)

SELECT <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>)

4.3.3 TRUY VẤN LỒNG (TT)

- Các câu lệnh SELECT có thể lồng nhau ở nhiều mức
- Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic
- Câu truy vấn con thường trả về một tập các giá trị

4.3.3 TRUY VẤN LỒNG (TT)

- Mệnh đề WHERE của câu truy vấn cha
 - <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

4.3.3 TRUY VẤN LỒNG (TT)

Ví dụ 1: Tìm những nhân viên làm việc trong các phòng ban thuộc địa điểm TP HCM

```
SELECT MANV, TENNV  
FROM NHANVIEN  
WHERE PHG IN ( (1, 5)
```

```
SELECT MAPHG  
FROM DIADIEM_PHG  
WHERE DIADIEM='TP HCM' )
```

4.3.3 TRUY VẤN LỒNG (TT)

Ví dụ 2: Tìm những nhân viên không có thân nhân nào

```
Cách 1: SELECT *  
FROM NHANVIEN  
WHERE MANV NOT IN (  
                SELECT MA_NVIEN  
                FROM THANNHAN )
```

```
Cách 2: SELECT *  
FROM NHANVIEN  
WHERE MANV <> ALL (  
                SELECT MA_NVIEN  
                FROM THANNHAN )
```

4.3.3 TRUY VẤN LỒNG (TT)

Ví dụ 3: Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên phòng 4

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

4.3.3 TRUY VẤN LỒNG (TT)

Ví dụ 4: 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_NVLEN  
    AND NV.TENNV=TN.TENTN  
    AND NV.PHAI=TN.PHAI )
```

4.3.3 TRUY VẤN LỒNG (TT)

Ví dụ 5: Tìm những nhân viên không có thân nhân nào

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

4.3.3 TRUY VẤN LỒNG (TT)

- Tìm những nhân viên có lương lớn hơn lương của ít nhất một nhân viên phòng 4
- Tìm những trưởng phòng có tối thiểu một thân nhân

→ Sv làm bài tại lớp và Gv sửa trực tiếp trên lớp

4.3.4 DÙNG IN VÀ EXISTS

○ IN

- <tên cột> IN <câu truy vấn con>
- 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

4.3.4 DÙNG IN VÀ EXISTS

○ 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

4.3.5 CÁC 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

○ Các hàm khác: MIN, MAX, SUM, AVG

○ Các hàm kết hợp được đặt ở mệnh đề SELECT

4.3.5 CÁC HÀM KẾT HỢP (TT)

- Ví dụ 1: Tìm tổng lương, lương cao nhất, lương thấp nhất và lương trung bình của các nhân viên

```
SELECT SUM(LUONG) AS TL, MAX(LUONG) AS LN,  
MIN(LUONG) AS NN, AVG(LUONG) AS TB FROM NHANVIEN
```

4.3.5 CÁC HÀM KẾT HỢP (TT)

- Ví dụ 2: 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'
```

4.3.6 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

4.3.6 GOM NHÓM (TT)

- Ví dụ 1: Cho biết số lượng nhân viên của từng phòng ban

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

4.3.6 GOM NHÓM (TT)

- Ví dụ 2: với mỗi nhân viên cho biết mã số, họ tên, số lượng đề án và tổng thời gian mà họ tham gia

MA_NVIENT	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

4.3.6 GOM NHÓM (TT)

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

```
SELECT MA_NVIEN, HONV, TENNV, COUNT(*) AS  
SL_DA, SUM(THOIGIAN) AS TONG_TG  
FROM PHANCONG, NHANVIEN  
WHERE MA_NVIEN=MANV  
GROUP BY MA_NVIEN, HONV, TENNV
```

4.3.6 GOM NHÓM (TT)

- Ví dụ 3: Cho biết những nhân viên tham gia từ 2 đề án trở lên

→ sinh viên làm tại lớp

4.3.7 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>

4.3.7 ĐIỀU KIỆN TRÊN NHÓM (TT)

- Ví dụ 1: Cho biết những nhân viên tham gia từ 2 đề án trở lên

```
SELECT MA_NVIEN, COUNT(*) AS SLDA  
FROM PHANCONG  
GROUP BY MA_NVIEN  
HAVING COUNT(*) >= 2
```

4.3.7 ĐIỀU KIỆN TRÊN NHÓM (TT)

- Ví dụ 2: Cho biết những phòng ban (TENPHG) có lương trung bình của các nhân viên lớn hơn 20000

→ Sv làm bài và gv sửa bài tại lớp

4.3.8 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 các bộ
- Sau khi gom nhóm điều kiện trên nhóm mới được thực hiện

4.3.9 CẤU TRÚC TRUY VẤN TỔNG QUÁT

SELECT <danh sách các cột>

FROM <danh sách các bảng>

[**WHERE** <điều kiện>]

[**GROUP BY** <các thuộc tính gom nhóm>]

[**HAVING** <điều kiện trên nhóm>]

[**ORDER BY** <các thuộc tính sắp thứ tự>]

LINK YOUTUBE

- Link: <https://www.youtube.com/watch?v=u5mJ6esGBMs>

CÂU HỎI THẢO LUẬN

- 1) Khi nào chúng ta nên thực hiện câu truy vấn lồng ghép?
- 2) SQL Server cho phép câu lệnh lồng ghép bao nhiêu mức?
- 3) Các hàm gom nhóm được đặt sau mệnh đề nào trong câu truy vấn?

→ Trả lời vào đầu buổi học tiếp theo

BÀI TẬP VỀ NHÀ

- Sử dụng SQL trong CSDL “Quản lý giảng dạy” để viết các yêu cầu sau:
 - 1) Đếm số giảng viên của từng khoa, hiển thị mã khoa và tên khoa trong kết quả.
 - 2) Hiển thị tổng số sinh viên của từng khoa.
 - 3) Hiển thị giảng viên có giờ dạy nhiều nhất

