

# Ngôn ngữ SQL – Các dạng truy vấn khác

## Các loại truy vấn khác

### A. Truy vấn con ở mệnh đề SELECT

- Với mỗi nhân viên, cho biết họ, tên nhân viên và số thân nhân của họ

```
SELECT HONV, TENLOT, TENNV, ( select count(*)
                             from THANNHAN
                             where MA_NVIEN = nv.MANV) as SoTN
FROM NHANVIEN nv
```

- Với mỗi phòng ban, cho biết tên phòng ban và lương trung bình của phòng ban

```
SELECT pb.TENPHG, ( select avg(LUONG)
                   from NHANVIEN
                   where PHG = pb.MAPHG) as LuongTB
FROM PHONGBAN pb
```

### B. 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

**VD:**

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

```
SELECT TENPHG, TEMP.LUONG_TB
FROM PHONGBAN, (SELECT PHG, AVG(LUONG) AS LUONG_TB
                FROM NHANVIEN
                GROUP BY PHG
                HAVING AVG(LUONG) > 20000 ) AS TEMP
WHERE MAPHG=TEMP.PHG
```

### C. Điều kiện kết ở mệnh đề FROM

**VD:**

- Tìm mã và tên các nhân viên làm việc tại phòng ‘Nghiên cứu’

```
SELECT MANV, TENNV
FROM NHANVIEN INNER JOIN PHONGBAN ON PHG=MAPHG
WHERE TENPHG='Nghiên cứu'
```

- Cho biết họ tên nhân viên và tên phòng ban mà họ là trưởng phòng nếu có

```
SELECT TENNV, HONV, TENPHG
FROM PHONGBAN RIGHT JOIN NHANVIEN ON MANV=TRPHG
```

- Tìm họ tên các nhân viên và tên các đề án nhân viên tham gia nếu có

```
SELECT NV.TENNV, NV.TENDA
FROM (PHANCONG PC JOIN DEAN DA ON SODA=MA DA)
RIGHT JOIN NHANVIEN NV ON PC.MA_NVIEEN=NV.MANV
```

## D. Cấu trúc Case

### 1. Cấu trúc 1

```
CASE input_expression
  WHEN when_expression THEN result_expression
  [ ...n ]
  [
    ELSE else_result_expression
  ]
END
```

Ví dụ :

- Cho biết họ tên các nhân viên và năm về hưu

```
SELECT HONV, TENNV,
(CASE PHAI
  WHEN 'Nam' THEN YEAR(NGSINH) + 60
  WHEN 'Nu' THEN YEAR(NGSINH) + 55
END ) AS NAMVEHUU
FROM NHANVIEN
```

- Cho biết họ tên các nhân viên đã đến tuổi về hưu (nam 60 tuổi, nữ 55 tuổi)

```
SELECT HONV, TENNV
FROM NHANVIEN
WHERE YEAR(GETDATE()) - YEAR(NGSINH) >= ( CASE PHAI
  WHEN 'Nam' THEN 60
  WHEN 'Nu' THEN 55
END )
```

### 2. Cấu trúc 2

```
CASE
  WHEN Boolean_expression THEN result_expression
  [ ...n ]
  [
    ELSE else_result_expression
  ]
END
```

Ví dụ :

- Cho biết sinh viên và xếp loại học lực của sinh viên

```
select MASV, CASE
      WHEN DIEMTB >= 8 THEN N'Giỏi'
      WHEN DIEMTB >= 7 THEN N'Khá'
      WHEN DIEMTB >= 6 THEN N'Trung bình khá'
      WHEN DIEMTB >= 5 THEN N'Trung bình'
    END
from SINHVIEN
```

Ví dụ :

- Cho biết họ tên các nhân viên đã đến tuổi về hưu (nam 60 tuổi, nữ 55 tuổi)

```
SELECT HONV, TENNV
FROM NHANVIEN
WHERE YEAR(GETDATE()) - YEAR(NGSINH) >= ( CASE PHAI
      WHEN 'Nam' THEN 60
      WHEN 'Nu' THEN 55
    END )
```

- Cho biết họ tên các nhân viên và năm về hưu

```
SELECT HONV, TENNV,
(CASE PHAI
  WHEN 'Nam' THEN YEAR(NGSINH) + 60
  WHEN 'Nu' THEN YEAR(NGSINH) + 55
END ) AS NAMVEHUU
FROM NHANVIEN
```

### E. Phép hội (UNION)

Phép hội (Union) sử dụng để tổng hợp dữ liệu từ các bảng → 1 bảng

- UNION : Các dòng trùng lặp sẽ được bỏ đi
- UNION ALL : Lấy tất cả các dòng của các bảng

Điều kiện để thực hiện được Union : Các bảng phải có cùng số lượng thuộc tính và tương ứng kiểu dữ liệu giữa các cột.

Các cột của bảng kết xuất chính là các cột trong bảng đầu tiên.

<u>TABLE1</u>		<u>TABLE2</u>	
a	b	a	b
1	2	2	7
3	4	3	4
2	3	2	3
4	5	1	6

UNION	UNION ALL																																
<pre>select * from table1 union select * from table2</pre>	<pre>select * from table1 union all select * from table2</pre>																																
<p><b><u>Kết quả:</u></b></p> <table border="1"> <thead> <tr> <th>a</th><th>b</th></tr> </thead> <tbody> <tr><td>1</td><td>2</td></tr> <tr><td>1</td><td>6</td></tr> <tr><td>2</td><td>3</td></tr> <tr><td>2</td><td>7</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>4</td><td>5</td></tr> </tbody> </table>	a	b	1	2	1	6	2	3	2	7	3	4	4	5	<p><b><u>Kết quả :</u></b></p> <table border="1"> <thead> <tr> <th>a</th><th>b</th></tr> </thead> <tbody> <tr><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td></tr> <tr><td>2</td><td>7</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>2</td><td>3</td></tr> <tr><td>1</td><td>6</td></tr> </tbody> </table>	a	b	1	2	3	4	2	3	4	5	2	7	3	4	2	3	1	6
a	b																																
1	2																																
1	6																																
2	3																																
2	7																																
3	4																																
4	5																																
a	b																																
1	2																																
3	4																																
2	3																																
4	5																																
2	7																																
3	4																																
2	3																																
1	6																																

### F. Phép giao (Intersect)

Sử dụng EXISTS hoặc IN để thực hiện phép giao

- Tìm các nhân viên có làm đề án của phòng nghiên cứu và vừa là trưởng phòng

**Nhận xét :** nhân viên có làm đề án của phòng nghiên cứu và vừa là trưởng phòng = Nhân viên làm đề án của phòng nghiên cứu  $\cap$  Nhân viên là trưởng phòng

- Sử dụng ...IN (...) AND ...IN (...)  $\rightarrow$  Lồng phân cấp

```
select *
from NHANVIEN
where MANV in ( select TRPHG
                from PHONGBAN pb)
and MANV in ( select pc.MA_NVIEN
                from PHANCONG pc, DEAN da, PHONGBAN pb
                where pc.MADA=da.MADA and
                      da.PHONG=pb.MAPHG and
                      pb.TENPHG = N'Nghiên Cứu')
```

- Sử dụng ... EXISTS  $\rightarrow$  Lồng tương quan

```

select *
from NHANVIEN nv, PHONGBAN pb
where pb.TRPBG = nv.MANV and exists (
    select *
    from PHANCONG pc, DEAN da, PHONGBAN pb
    where pc.MADA=da.MADA and
          da.PHONG=pb.MAPBG and
          pb.TENPBG = N'Nguyễn Cửu' and
          pc.MA_NVN=nv.MANV)

```

- Sử dụng ... IN → Lồng phân cấp

```

select *
from NHANVIEN nv, PHONGBAN pb
where pb.TRPBG = nv.MANV and MANV in (
    select pc.MA_NVN
    from PHANCONG pc, DEAN da, PHONGBAN pb
    where pc.MADA=da.MADA and
          da.PHONG=pb.MAPBG and
          pb.TENPBG = N'Nguyễn Cửu')

```

Sử dụng EXISTS và NOT EXISTS để thực hiện phép giao và phép trừ

### G. Phép trừ

Sử dụng NOT EXISTS hoặc NOT IN để thực hiện phép trừ

- Tìm các nhân viên không tham gia đề án nào

Nhận xét : Nhân viên không tham gia đề án = Tất cả nhân viên – Nhân viên có tham gia đề án

Sử dụng NOT EXISTS

```

SELECT NV.HONV, NV.TENLOT, NV.TENNV
FROM NHANVIEN NV
WHERE NOT EXISTS (
    SELECT *
    FROM PHANCONG
    WHERE MA_NVN=NV.MANV)

```

SỬ DỤNG NOT IN

```

SELECT NV.HONV, NV.TENLOT, NV.TENNV
FROM NHANVIEN NV
WHERE NV.MANV NOT IN(
    SELECT MA_NVN
    FROM PHANCONG)

```

- Đề án không có nhân viên nào tham gia

Nhận xét : Đề án không có nhân viên tham gia = Tất cả các đề án – Đề án có nhân viên tham gia

Sử dụng NOT EXISTS

```
SELECT *  
FROM DEAN da  
WHERE NOT EXISTS (  
    SELECT *  
    FROM PHANCONG  
    WHERE MADA = da.MADA)
```

Sử dụng NOT IN

```
SELECT *  
FROM DEAN da  
WHERE MADA NOT IN (  
    SELECT MADA  
    FROM PHANCONG)
```