

Ngôn ngữ SQL – Truy vấn gom nhóm

Câu truy vấn sử dụng Group By

A. Các hàm tính toán

SQL sử dụng các hàm sau: Count, Max, Min, Sum, Avg. Hàm Count dùng đối số * có nghĩa là đếm tất cả các mẫu tin thỏa điều kiện đếm mà không cần quan tâm đến bất kỳ cột nào.

- Có tất cả bao nhiêu sinh viên trong lớp th01

```
SELECT COUNT(*)
FROM SINHVIEN
WHERE LOP = 'TH01'
```

B. Mệnh đề group by

Dùng để gom nhóm dữ liệu, thường dùng kết hợp với một hàm tính toán kể trên.

- Tính điểm trung bình của từng sinh viên, biết rằng điểm số lưu trong bảng KETQUA(MASV, MAMH, DIEM)

```
SELECT MASV, AVG(DIEM)
FROM KETQUA
GROUP BY MASV
```

- Cho biết lương lớn nhất trong từng phòng ban

NHANVIEN(MANV, TENNV, PHAI, LUONG, PHG)

PHONGBAN(MAPB, TENPB, TRPHG)

THANNHAN(MA_NV, TENTN, PHAI, QUANHE)

```
SELECT PHG, MAX(LUONG)
FROM NHANVIEN
GROUP BY PHG
```

C. Mệnh đề having

Mệnh đề HAVING thường được sử dụng cùng với mệnh đề GROUP BY. Sau HAVING là biểu thức điều kiện. Biểu thức điều kiện này không tác động vào toàn bảng được chỉ ra ở mệnh đề from mà chỉ tác động lần lượt từng nhóm các mẫu tin đã chỉ ra trong mệnh đề group by.

- Cho biết các sinh viên có điểm trung bình lớn hơn hoặc bằng 8.0

```
SELECT MASV, AVG(DIEM)
FROM KETQUA
GROUP BY MASV
HAVING AVG(DIEM) >= 8.0
```

II. Truy vấn lồng

A. Tìm kiếm có lượng từ EXISTS, ANY và ALL

- Cho danh sách các nhân viên có ít nhất 1 thân nhân.

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE EXISTS (SELECT TENTN
FROM THANHMANH
WHERE THANHMANH.MA_NHVIEN = NHANVIEN.MANV)
```

Câu này có thể viết lại như sau:

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE (SELECT COUNT (*)
FROM THANHMANH
WHERE THANHMANH.MA_NHVIEN = NHANVIEN.MANV) > 0
```

Chú ý: = ANY tương đương với toán tử IN

- Cho biết nhân viên có lương lớn nhất.

```
SELECT MANV, LUONG
FROM NHANVIEN
WHERE LUONG >= ALL (SELECT LUONG FROM NHANVIEN)
```

Hoặc có thể viết như sau:

```
SELECT MANV, LUONG
FROM NHANVIEN
WHERE LUONG = (SELECT MAX(LUONG) FROM NHANVIEN)
```

- Cho biết sinh viên có điểm trung bình lớn nhất.

```
SELECT MASV, AVG(DIEM)
FROM KETQUA
GROUP BY MASV
WHERE AVG(DIEM) >= ALL (SELECT AVG(DIEM)
FROM KETQUA
GROUP BY MASV)
```

Có 2 loại truy vấn lồng

B. Loại 1: Lồng phân cấp

Mệnh đề WHERE của truy vấn con không tham chiếu đến thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha

Khi thực hiện, câu truy vấn con sẽ được thực hiện trước

Ví dụ:

- Cho biết các nhân viên cùng phòng với nhân viên “Nguyễn Văn A”

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE PHG IN (
    SELECT PHG
    FROM NHANVIEN
    WHERE TENNV = 'Nguyễn Văn A'
)
```

Quan hệ **NHANVIEN** ở truy vấn con không liên quan đến quan hệ **NHANVIEN** ở truy vấn cha

- Tìm những nhân viên có lương lớn hơn lương của tất cả nhân viên ở phòng 4.

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE LUONG > (
    SELECT MAX(LUONG)
    FROM NHANVIEN
    WHERE PHG = 4
)
```

- Tìm phòng ban có đông nhân viên nhất (gom nhóm + truy vấn lồng phân cấp)

```
select pb.TENPHG, count(*) as SOLUONG
from NHANVIEN nv, PHONGBAN pb
where nv.PHG = pb.MAPHG
group by nv.PHG, pb.TENPHG
having count(*) >= ALL (
    select count(*)
    from NHANVIEN
    group by PHG)
```

- Cho biết họ tên nhân viên (HONV, TENLOT, TENNV) có mức lương lớn hơn mức lương của một nhân viên nào đó của phòng "Nghiên cứu"

```
select HONV, TENLOT, TENNV, LUONG, PHG
from NHANVIEN
where LUONG > any(select nv.LUONG
    from NHANVIEN nv, PHONGBAN pb
    where nv.PHG=pb.MAPHG
    and pb.TENPHG=N'Nghiên cứu' )
```

-

C. Loại 2: Lồng tương quan

Mệnh đề WHERE của truy vấn con tham chiếu ít nhất một thuộc tính của các quan hệ trong mệnh đề FROM ở truy vấn cha.

Khi thực hiện, câu truy vấn con sẽ được thực hiện nhiều lần, mỗi lần tương ứng với một bộ của truy vấn cha.

Ví dụ:

- Tìm những nhân viên không có thân nhân nào:

```
SELECT MANV, TENNV
FROM NHANVIEN n
WHERE NOT EXISTS
```

```
(SELECT *
FROM THANNHAN t
WHERE t.MANV = n.MANV)
```

Trong truy vấn con này có tham chiếu đến thuộc tính **MANV** của quan hệ **NHANVIEN n** trên truy vấn cha

- *Tìm tất cả các nhân viên làm việc ở phòng nghiên cứu*

```
SELECT MANV, TENNV
FROM NHANVIEN n
WHERE EXISTS
```

```
(SELECT *
FROM PHONGBAN p
WHERE TENPHG = 'Nghiên cứu'
and p.MAPHG = n.PHG)
```

Trong truy vấn con này có tham chiếu đến thuộc tính **PHG** của quan hệ **NHANVIEN n** trên truy vấn cha

III. Phép chia

Có 2 cách thực hiện:

Cách 1: Sử dụng NOT EXISTS + NOT IN hoặc NOT EXISTS + NOT EXISTS

Cách 2: Sử dụng mệnh đề GROUP BY + HAVING

VD:

- Tìm nhân viên được phân công làm việc trong tất cả các đề án do phòng **Nghiên cứu** quản lý

Cách 1:

Sử dụng NOT EXISTS + NOT IN

```
SELECT n.MANV, n.TENNV
FROM NHANVIEN n
WHERE NOT EXISTS
(
    SELECT *
    FROM PHONGBAN pb, DEAN d
    WHERE d.PHONG = pb.MAPHG AND pb.TENPHG = 'Nghiên cứu' AND
    d.MADA NOT IN
    (
        SELECT pc.MADA
        FROM PHANCONG pc
        WHERE pc.MA_NVNIEN = n.MANV
    )
)
```

Sử dụng NOT EXISTS + NOT EXISTS

```

SELECT n.MANV, n.TENNV
FROM NHANVIEN n
WHERE NOT EXISTS
(
    SELECT *
    FROM PHONGBAN pb, DEAN d
    WHERE d.PHONG = pb.MAPHG AND pb.TENPHG = 'Nghiên cứu'
    AND NOT EXISTS
    (
        SELECT *
        FROM PHANCONG pc
        WHERE pc.MA_NVIEN = n.MANV AND pc.MADA = d.MADA
    )
)

```

Cách 2: Sử dụng GROUP BY + HAVING

```

SELECT n.MANV, n.TENNV
FROM NHANVIEN n, PHANCONG pc, PHONGBAN pb1, DEAN d1
WHERE n.MANV = pc.MA_NVIEN AND pc.MADA = d1.MADA
and d1.PHONG = pb1.MAPHG AND pb1.TENPHG = 'Nghiên cứu'
GROUP BY n.MANV, n.TENNV
HAVING COUNT (DISTINCT pb.MADA) =
    (SELECT COUNT(DISTINCT d2.MADA)
    FROM DEAN d2, PHONGBAN pb2
    WHERE d2.PHONG = pb2.MAPHG AND pb2.TENPHG = 'Nghiên cứu')

```

IV. 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 ‘Nghien cuu’

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

- 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=MADA)
RIGHT JOIN NHANVIEN NV ON PC.MA_NVN=NV.MANV
```

D. Cấu trúc Case

- 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
```