Ngôn ngữ SQL – Phần 2

I. Nội dung cần quan tâm

1) Tổng quát câu truy vấn.
2) Các loại truy vấn đơn giản.
3) Câu truy vấn group by.
4) Truy vấn lồng và Phép chia
5) Các dạng truy vấn khác.

II. Tổng quát

Một cách tổng quát, khối select gồm có 3 mệnh đề chính:

Select: Xác định các cột cần đưa ra kết quả.

From: Xác định các bảng cần lấy thông tin ra.

Where: Xác định các mẫu tin thỏa yêu cầu chọn lọc để đưa ra kết quả.

Ngoài ra, để mở rộng khả năng của ngôn ngữ, khối select-from-where còn được bổ sung thêm các mệnh đề **group by, having, order by,** các hàm hỗ trợ tính toán: **max, min, count, sum, avg**.

Sau đây là cú pháp tổng quát của câu truy vấn dữ liệu:

SELECT [tính chất] <danh sách các thuộc tính_1>

FROM <danh sách các table hoặc query/view [as alias] >

[**WHERE** <điều kiện_1>]

[**GROUP BY** <danh sách các thuộc tính_2>]

[**HAVING** <điều kiện_2>]

[**ORDER BY** <danh sách các thuộc tính_3 [ASC | DESC]>

Diễn giải :

1. Tính chất : Một trong các từ khóa: ALL (chọn ra tất cả các dòng trong bảng), DISTINCT (loại bỏ các cột trùng lắp thông tin), DISTINCTROW (loại bỏ các dòng trùng lắp thông tin), TOP <n> (chọn n dòng đầu tiên thỏa mãn điều kiện).
2. Danh sách các thuộc tính_1: tên các thuộc tính cho biết thông tin cần lấy.
   Chú ý: Các thuộc tính cách nhau bởi dấu ‘,’

Nếu lấy tất cả các thuộc tính của 1 bảng tbl thì dùng: tbl.*

Nếu sau FROM chỉ có 1 table và lấy tất cả các field của table đó thì dùng select *

Nếu tồn tại 1 thuộc tính sau select xuất hiện ở 2 table sau FROM thì phải chỉ định rõ thuộc tính đó thuộc table nào.

3. Danh sách các table: các table chứa thông tin cần lấy. Khi tìm kiếm thông tin trên nhiều hơn 2 table thì phải kết các table lại với nhau (điều kiện kết đặt sau where)
4. Alias: bí danh (tên tắt) của bảng dùng cho các bảng có tên quá dài.
5. Điều kiện_1: là điều kiện để lọc dữ liệu.
6. Danh sách các thuộc tính_2: dữ liệu sẽ được gom nhóm theo các cột này, ưu tiên từ trái sang.
7. Điều kiện_2: điều kiện lọc lại dữ liệu sau khi đã thực hiện tính tóan trên dữ liệu. Điều kiện này được áp dụng trên dữ liệu thỏa mãn điều kiện_1.
8. Danh sách các thuộc tính_3:sắp xếp dữ liệu theo cột nào, thứ tự là tăng (ASC) hoặc giảm (DESC). Mặc định là dữ liệu được sắp theo thứ tự tăng dần. Việc sắp xếp được thực hiện theo thứ tự ưu tiên từ trái qua phải.

III. Truy vấn đơn giản

SELECT <danh sách thuộc tính>

FROM tên_bảng

Sau select, * được dùng với ý nghĩa lấy toàn bộ các cột của bảng.

Dùng từ khoá **distinct** để loại bỏ các bộ trùng nhau và **all** để lấy tất cả các bộ dữ liệu. Mặc định không để gì cả chính là có dùng từ khóa all.

Sau select có thể dùng các biểu thức số học như: +, -, *, /, và có thể thực hiện các toán tử trên thuộc tính.

VD:

- Cho biết danh sách tất cả các nhân viên với tất cả các thông tin

```
SELECT *
FROM NHANVIEN
```

A. Tìm kiếm có sắp xếp

Để sắp xếp thứ tự dữ liệu, ta sử dụng mệnh đề ORDER BY:

SELECT…

FROM…

ORDER BY thuộc_ tính_1[ASC|DESC], thuộc_tính_2[ASC|DESC], ...

Tập_thuộc_tính gồm 1 thuộc tính hoặc nhiều thuộc tính và độ ưu tiên tính từ trái sang phải.

VD:

Với câu lệnh: *select * from Table1 order by B desc,A asc* trên bảng dưới đây:

A	B
An	8
Binh	8
Chi	9
Hung	10

Ta sẽ được kết quả sau:

A	B
Hung	10
Chi	9
An	8
Binh	8

Đầu tiên là xếp thứ tự theo B trước, sau đó, với những giá trị B ngang nhau thì sẽ xếp theo A.

VD:

- Cho biết danh sách các nhân viên sắp tên theo thứ tự Alphabet

```
SELECT MANV, TENNV, PHAI, LUONG
FROM NHANVIEN
ORDER BY TENNV
```

- Cho biết danh sách các nhân viên theo từng phòng ban, trong từng phòng ban tên nhân viên sắp theo thứ tự

```
SELECT PHG, MANV, TENNV, PHAI, LUONG
FROM NHANVIEN
ORDER BY PHG, TENNV
```

B. Tìm kiếm với điều kiện đơn giản

Để hỗ trợ tìm kiếm có điều kiện, sử dụng mệnh đề WHERE trong câu lệnh SELECT với vị trí như sau:

1. AND và OR

```
SELECT…

FROM…
```

WHERE (điều_kiện_1) AND/OR ....(điều_kiện_n)

VD:

SINHVIEN (MASV, HOTEN, NGSINH, LOP)

- Cho danh sách các sinh viên của lớp TH01.

```
SELECT *
FROM SINHVIEN
WHERE LOP = 'TH01'
```

Lưu ý: Khi thuộc tính có thể nhận giá trị null, cần cẩn thận khi sử dụng để so sánh với nhiều điều kiện liên tiếp.

2. BETWEEN...AND , NOT BETWEEN ... AND

- Cho biết các nhân viên sinh trong khoảng năm 1955 đến 1960

```
SELECT *
FROM NHANVIEN
WHERE NGSINH between '1/1/1955' and '12/31/1960'
```

Hoặc

```
SELECT *
FROM NHANVIEN
WHERE year(NGSINH) between 1955 and 1960
```

Hoặc

```
SELECT *
FROM NHANVIEN
WHERE year(NGSINH)>= 1955 and year(NGSINH)<= 1960
```

3. IS NULL và IS NOT NULL

IS NULL và IS NOT NULL : Để kiểm tra một giá trị có phải là NULL | NOT NULL hay không

- Cho biết các nhân viên không có người quản lý trực tiếp

```
SELECT HONV, TENLOT, TENNV
FROM NHANVIEN
WHERE MA_NQL IS NULL
```

- Cho biết các nhân viên có người quản lý trực tiếp

```
SELECT HONV, TENLOT, TENNV
FROM NHANVIEN
WHERE MA_NQL IS NOT NULL
```

4. IN và NOT IN

IN và NOT IN dùng để kiểm tra một giá trị nằm trong hay không nằm trong một tập hợp nào đó hay không.

- Cho biết các đơn đặt hàng có đặt mặt hàng H1, H2, H3.

```
SELECT MADH
FROM DONHANG
WHERE MAMH IN ('H1','H2','H3')
```

C. Tìm kiếm có xử lý xâu ký tự

Để xử lý với các dữ liệu thuộc dạng xâu ký tự, ngôn ngữ SQL có hỗ trợ phép LIKE. Thông thường khi so sánh thuộc tính có kiểu dữ liệu thuộc dạng xâu ký tự thì người ta thường dùng LIKE chứ không dùng phép bằng =

VD:

- Hiện ra các sinh viên tên Trang

```
SELECT *
FROM SINHVIEN
WHERE HOTEN LIKE '%Trang'
```

% : dùng để đại diện cho nhiều ký tự đứng trước từ 'Trang'

Ngoài ra còn có các ký tự sau để mô tả mẫu cần tìm:

_ thay thế cho ký tự bất kỳ.

Chú ý:

Like "ab\%cd%" cho ra những chuỗi bắt đầu với "ab%cd"

Like "ab\\cd%" cho ra những chuỗi bắt đầu với "ab\cd"

D. Tìm kiếm có điều kiện liên quan đến ngày tháng

VD:

DDH(MADH, NGAYDH, MAKH)

CTDH(MADH, MAHH, SOLUONG, DONGIA)

- Cho biết những đơn đặt hàng đặt trước ngày 01/01/2001

```
SELECT MADH, NGAYDH
FROM DDH
WHERE DATEDIFF(D,NGAYDH,'01/01/2001') > 0
```

- Cho biết những đơn đặt hàng đặt trước ngày 01/01/2001 là 1 tuần

```
SELECT MADH, NGAYDH
FROM DDH
WHERE DATEDIFF(D,NGAYDH,'01/01/2001') > 7
```

Lưu ý :

- Cho biết các nhân viên sinh ngày 30/4/1975

Cách 1 :

```
select *
from NHANVIEN
where NGSINH = '1/1/1965'
```

Cách 2 :

```
select *
from NHANVIEN
where DATEDIFF(d, NGSINH, '1/1/1965')= 0
```

→ Cách 2 : Chính xác hơn

E. Sử dụng các hàm trong khi tìm kiếm

- Sử dụng hàm trong mệnh đề where
- Sử dụng hàm trong mệnh đề select : Trong mệnh đề select ngoài việc được sử dụng các toán tử như +, -, *, / ta còn có thể sử dụng hàm đối với các thuộc tính.
  - Các hàm về ngày tháng :
    - DateDiff
    - DatePart
    - GetDate
    - Year
    - Month
    - Day
    - DateAdd
  - Các hàm về chuỗi
  - Các hàm chuyển đổi kiểu dữ liệu
  - Các hàm toán học

- …

Để xem thông tin chi tiết về các hàm có thể sử dụng Book Onlines

- Cho biết họ tên nhân viên và tuổi của nhân viên

```
select HONV, TENLOT, TENNV, datediff(yyyy, NGSINH, getdate()) as TUOI
from NHANVIEN
```

- Cho biết năm sinh của nhân viên

```
select HONV, TENLOT, TENNV, year(NGSINH) as NAMSINH
from NHANVIEN
```

- Cho biết họ và tên đầy đủ của nhân viên

```
select HONV + ' ' + TENLOT + ' ' + TENNV as HoVaTen
from NHANVIEN
```

F. Tìm kiếm từ nhiều bảng

Để tìm kiếm thông tin mà thông tin đó nằm ở nhiều bảng khác nhau thì khai báo sử dụng các bảng đó tại mệnh đề FROM. Tùy theo thông tin cần hiển thị mà chúng ta sẽ sử dụng điều kiện tại mệnh đề WHERE sao cho thích hợp.

VD:

- Cho biết mã nhân viên, tên nhân viên, tên phòng ban mà nhân viên trực thuộc.

```
SELECT MANV, TENNV, TENPB
FROM NHANVIEN, PHONGBAN
WHERE NHANVIEN.MAPB = PHONGBAN.MAPB
```

G. Dùng toán tử θ some, θ all, exists, not exists

Lưu ý: <> some và not in, <> all = not in.

IV. 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_NVIEN, 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
```

V. 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 THANNHAN
WHERE THANNHAN.MA_NVIEN = NHANVIEN.MANV)
```

Câu này có thể viết lại như sau:

```
SELECT MANV, TENNV
FROM NHANVIEN
WHERE (SELECT COUNT(*)
FROM THANNHAN
WHERE THANNHAN.MA_NVIEN = 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

VI. 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 lí

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_NVIEN = 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')
```

VII. 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 lơn 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_NVIEN=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
```