

Ngôn ngữ SQL – Phần 2

I. Thực hành

1. Tổng quát câu truy vấn
2. Các loại truy vấn đơn giản
3. Các truy vấn group by
4. Truy vấn lồng nhau 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] <danhsach các thuộc tính_1>

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

[WHERE <điều kiện_1>]

[GROUP BY <danhsach các thuộc tính_2>]

[HAVING <điều kiện_2>]

[ORDER BY <danhsach 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 lặp thông tin), DISTINCTROW (loại bỏ các dòng trùng lặp 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 toán 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
```

1. 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
```

2. 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:

@ AND và OR

SELECT...

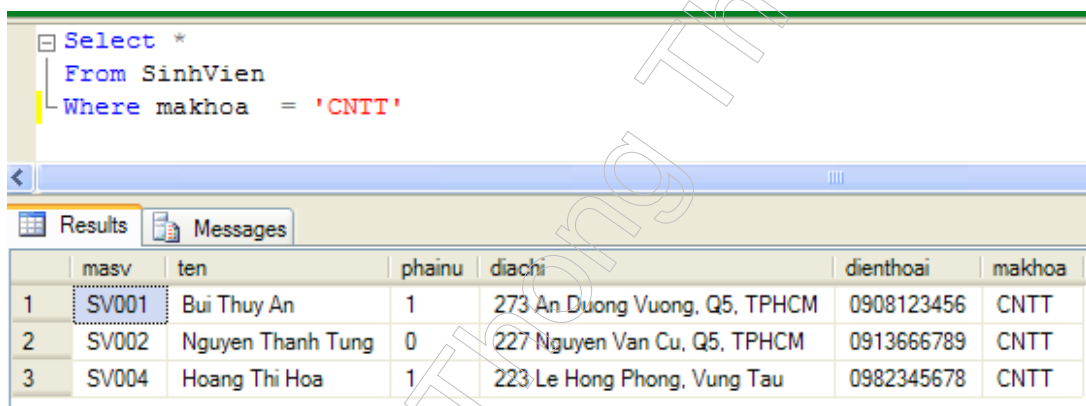
FROM...

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

VD:

SinhVien(masv, hoten, ngsinh, makhoa)

Cho danh sách các sinh viên có makhoa = 'CNTT'



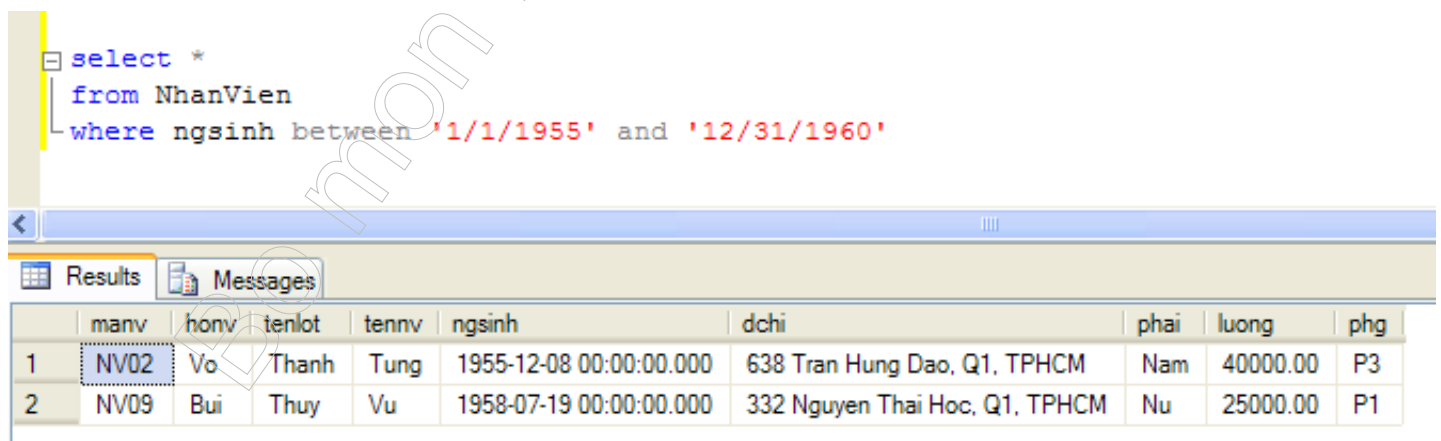
```
Select *
From SinhVien
Where makhoa = 'CNTT'
```

	masv	ten	phainu	diachi	dienthoai	makhoa
1	SV001	Bui Thuy An	1	273 An Duong Vuong, Q5, TPHCM	0908123456	CNTT
2	SV002	Nguyen Thanh Tung	0	227 Nguyen Van Cu, Q5, TPHCM	0913666789	CNTT
3	SV004	Hoang Thi Hoa	1	223 Le Hong Phong, Vung Tau	0982345678	CNTT

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.

@ 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'
```

	manv	honv	tenlot	tennv	ngsinh	dchi	phai	luong	phg
1	NV02	Vo	Thanh	Tung	1955-12-08 00:00:00.000	638 Tran Hung Dao, Q1, TPHCM	Nam	40000.00	P3
2	NV09	Bui	Thuy	Vu	1958-07-19 00:00:00.000	332 Nguyen Thai Hoc, Q1, TPHCM	Nu	25000.00	P1

Hoặc:

```
--OR:
select * from NhanVien
Where YEAR(ngsinh) between 1955 and 1960

--OR:
select * from NhanVien
Where YEAR(ngsinh) >= 1955 and YEAR(ngsinh) <= 1960
```

@ 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 những nhân viên tham gia các đề án DA5, DA4, DA3.

```
----Cho biết những nhân viên tham gia các đề án DA5, DA4, DA3--
select * from PhanCong
where mada in ('DA5', 'DA4', 'DA3')
```

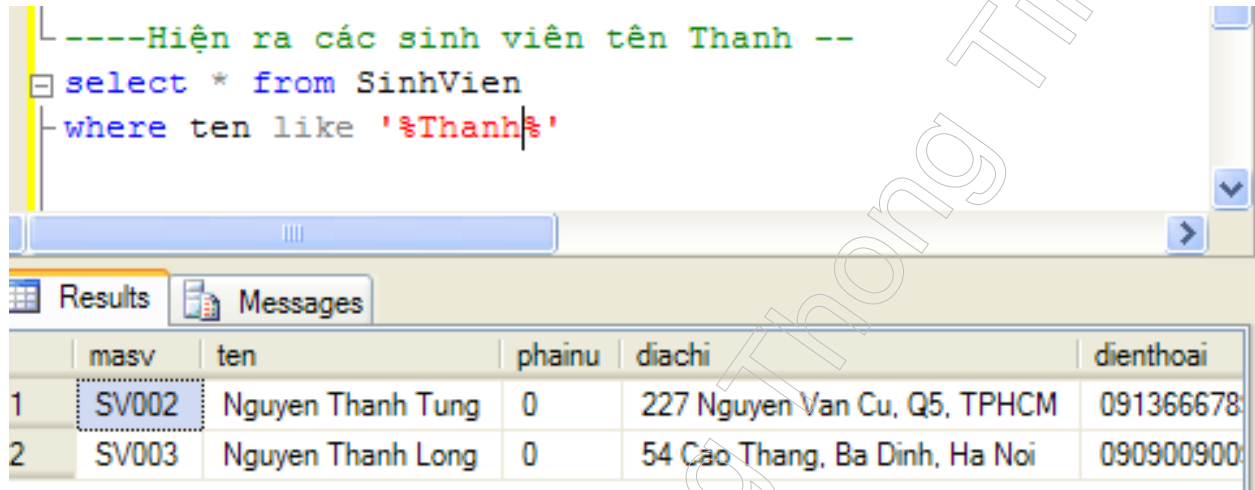
	manv	mada	thoigian
1	NV01	DA3	10
2	NV02	DA3	40
3	NV02	DA4	20
4	NV02	DA5	20
5	NV04	DA3	10
6	NV05	DA4	10
7	NV05	DA5	10
8	NV08	DA4	10

3. Tìm kiếm có xử lý chuỗi ký tự

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

VD:

- Hiện ra các sinh viên tên Thanh



```
--Hiện ra các sinh viên tên Thanh --
select * from SinhVien
where ten like '%Thanh%'
```

	masv	ten	phainu	điachi	đienthoai
1	SV002	Nguyen Thanh Tung	0	227 Nguyen Van Cu, Q5, TPHCM	091366678
2	SV003	Nguyen Thanh Long	0	54 Cao Thang, Ba Dinh, Ha Noi	090900900

% : 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"

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

VD: DDH(MADH, NGÀYDH, MAKH)

CTDH(MADH, MAHH, SOLUONG, DONGIA)

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

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

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

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

5. 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.
- o Các hàm về ngày tháng :
- DateDiff
 - DatePart
 - GetDate
 - Year
 - Month
 - Day
 - DateAdd
- o Các hàm về chuỗi
- o Các hàm chuyển đổi kiểu dữ liệu
- o Các hàm toán học
- o 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

```
--Cho biết họ tên nhân viên và tuổi của nhân viên --
select honv, tenlot, tennv, DATEDIFF (YYYY,ngsinh,GETDATE()) as tuoi
from NhanVien
```

	honv	tenlot	tennv	tuoi
1	Dinh	Ba	Tien	44
2	Vo	Thanh	Tung	54
3	Tran	Thanh	Tam	37
4	Vo	Manh	Hung	47
5	Le	Ngoc	Quyen	32
6	Tran	Minh	Khang	47
7	Le	Thi	Nhan	48
8	Tran	Hong	Quang	30
9	Bui	Thuy	Vu	51

- Cho biết năm sinh của nhân viên

```

L ----Cho biết năm sinh của nhân viên --
| Select honv , tenlot , tennv , YEAR(ngsinh) as namsinh
| From NhanVien

```

	honv	tenlot	tennv	namsinh
1	Dinh	Ba	Tien	1965
2	Vo	Thanh	Tung	1955
3	Tran	Thanh	Tam	1972
4	Vo	Manh	Hung	1962
5	Le	Ngoc	Quyen	1977
6	Tran	Minh	Khang	1962
7	Le	Thi	Nhan	1961
8	Tran	Hong	Quang	1979
9	Bui	Thuy	Vu	1958

- Cho biết họ và tên đầy đủ của nhân viên

```

L -----Cho biết họ và tên đầy đủ của nhân viên --
| select honv + ' ' + tenlot + ' ' + tennv as HovaTen
| From NhanVien

```

	HovaTen
1	Dinh Ba Tien
2	Vo Thanh Tung
3	Tran Thanh Tam
4	Vo Manh Hung
5	Le Ngoc Quyen
6	Tran Minh Khang
7	Le Thi Nhan
8	Tran Hong Quang
9	Bui Thuy Vu

6. 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, tenphg
from NhanVien as nv, PhongBan as pb
where nv.phg = pb.maphg
```

	manv	tennv	tenphg
1	NV01	Tien	Nghien Cuu
2	NV02	Tung	Nghien Cuu
3	NV03	Tam	Dieu Hanh
4	NV04	Hung	Dieu Hanh
5	NV05	Quyên	Quan Ly
6	NV06	Khang	Quan Ly
7	NV07	Nhan	Nghien Cuu
8	NV08	Quang	Dieu Hanh
9	NV09	Vu	Quan Ly

IV. Câu truy vấn sử dụng Group By

1. 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 Khoa CNTT

```
select COUNT (*)  
from SinhVien  
where makhoa = 'cntt'
```

2. 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)

```
-----Tính điểm trung bình của từng sinh viên --  
Select masv, AVG(diem)  
From KetQua  
Group by masv
```

	masv	(No column name)
1	SV001	8.75
2	SV002	7
3	SV003	8
4	SV004	4
5	SV005	7

- Cho biết lương lớn nhất trong từng phòng ban

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

PHONGBAN(MAPB, TENPB, TRPHG)

```
-----Cho biết lương lớn nhất trong từng phòng ban --  
select phg, MAX(luong)  
from NhanVien  
Group by phg
```

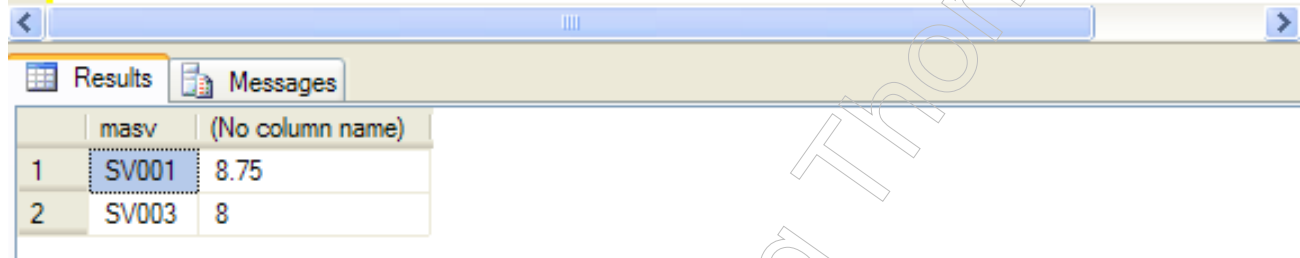
	phg	(No column name)
1	P1	58000.00
2	P2	38000.00
3	P3	43000.00

3. 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

```
-----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
```



The screenshot shows a SQL query window with the following text: `-----Cho biết các sinh viên có điểm trung bình lớn hơn hoặc bằng 8.0`, `select masv, AVG(diem)`, `From KetQua`, `group by masv`, and `having AVG(diem) >= 8.0`. Below the query window, the 'Results' tab is active, displaying a table with two columns: 'masv' and '(No column name)'. The table contains two rows: row 1 with 'SV001' and '8.75', and row 2 with 'SV003' and '8'.

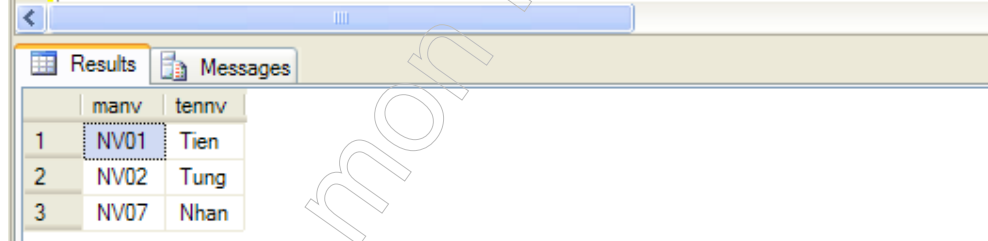
	masv	(No column name)
1	SV001	8.75
2	SV003	8

V. Truy vấn lồng

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

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

```
-----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.manv = NhanVien.manv
                )
```



The screenshot shows a SQL query window with the following text: `-----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.manv = NhanVien.manv`, and `)`. Below the query window, the 'Results' tab is active, displaying a table with two columns: 'manv' and 'tennv'. The table contains three rows: row 1 with 'NV01' and 'Tien', row 2 with 'NV02' and 'Tung', and row 3 with 'NV07' and 'Nhan'.

	manv	tennv
1	NV01	Tien
2	NV02	Tung
3	NV07	Nhan

Câu này có thể viết lại như sau:

```
---OR:
select manv, tennv
from NhanVien
where (select COUNT(*)
      from ThanNhan
      where ThanNhan.manv = NhanVien.manv ) > 0
```

	manv	tennv
1	NV01	Tien
2	NV02	Tung
3	NV07	Nhan

- Cho biết nhân viên có lương lớn nhất.

```
-----Cho biết nhân viên có lương lớn nhất. --
select manv, lương
from NhanVien
where lương >= All (select lương from NhanVien)
```

	manv	lương
1	NV06	58000.00

Hoặc có thể viết như sau:

```
-----Hay có thể viết như sau --
select manv, lương
from NhanVien
where lương = (select MAX(lương) from NhanVien )
```

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

```
-----Cho biết sinh viên có điểm trung bình lớn nhất. --
select masv, (AVG(diem))
from KetQua
group by masv
having avg(diem) >= All (
                        select AVG(diem)
                        from KetQua
                        Group by masv )
```

	masv	(No column name)
1	SV001	8.75

Có 2 loại truy vấn lồng

2. 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 "Hung".

```
-----Cho biết các nhân viên cùng phòng với nhân viên "Hung" --
select manv, tennv, phg
from NhanVien
where phg in (
    select phg
    from NhanVien
    where tennv = 'Hung' )
```

Quan hệ NhanVien ở truy vấn con không liên quan đến quan hệ NhanVien ở truy vấn cha

	manv	tennv	phg
1	NV03	Tam	P2
2	NV04	Hung	P2
3	NV08	Quang	P2

- Tìm những nhân viên có lương lớn hơn lương của tất cả nhân viên ở phòng 3.

```
-----Tìm những nhân viên có lương lớn hơn lương của tất cả nhân viên ở phòng 3. --
select manv, tennv, luong
from NhanVien
where luong > (
    select MAX(luong)
    from NhanVien
    where phg = 'P3' )
```

	manv	tennv	luong
1	NV05	Quyen	55000.00
2	NV06	Khang	58000.00

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

```

----Tìm phòng ban có đông nhân viên nhất ---
select pb.tenphg , COUNT(*) as SoLuong
from NhanVien as nv, PhongBan as pb
where nv.phg = pb.maphg
group by nv.phg, pb.tenphg
having COUNT(*) >= ALL ( select COUNT(*)
                        from NhanVien
                        group by phg
                      )

```

	tenphg	SoLuong
1	Quan Ly	3
2	Dieu Hanh	3
3	Nghien Cuu	3

- 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"

```

----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 = 'Nghien Cuu')

```

	honv	tenlot	tennv	luong	phg
1	Vo	Thanh	Tung	40000.00	P3
2	Vo	Manh	Hung	38000.00	P2
3	Le	Ngoc	Quyen	55000.00	P1
4	Tran	Minh	Khang	58000.00	P1
5	Le	Thi	Nhan	43000.00	P3

Chú ý: = ANY tương đương với toán tử IN

3. 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:

```
-----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 các thuộc tính **manv** của quan hệ **NhanVien** **n** trên truy vấn cha

	manv	tennv
1	NV03	Tam
2	NV04	Hung
3	NV05	Quyen
4	NV06	Khang
5	NV08	Quang
6	NV09	Vu

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

```
-----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 = 'Nghien Cuu' 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

	manv	tennv
1	NV01	Tien
2	NV02	Tung
3	NV07	Nhan

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 lý

Cách 1:

Sử dụng NOT EXISTS + NOT IN

```
select n.manv , n.tennv
from NhanVien n
where not exists ( select *
                  from PhongBan p, DeAn d
                  where d.phong = p.maphg and p.tenphg='Nghiên Cứu'
                     and d.mada not in ( select pc.mada
                                         from PhanCong pc
                                         where pc.manv = n.manv
                                       )
                )
```

Results Messages

manv	tennv
NV01	Tien

Sử dụng NOT EXISTS + NOT EXISTS

```
-----Sử dụng NOT EXISTS + NOT EXISTS
select n.manv , n.tennv
from NhanVien n
where not exists ( select *
                  from PhongBan p, DeAn d
                  where d.phong = p.maphg and p.tenphg='Nghiên Cứu'
                     and not exists ( select *
                                     from PhanCong pc
                                     where pc.manv = n.manv
                                       and pc.mada = d.mada
                                   )
                )
```

Results Messages

manv	tennv
1 NV01	Tien

Cách 2:

Sử dụng GROUP BY + HAVING

```
-----Cách 2: Sử dụng GROUP BY + HAVING
select n.manv , n.tennv
from NhanVien n, PhanCong pc, PhongBan p1, DeAn d1
where n.manv = pc.manv and pc.mada = d1.mada
    and d1.phong = p1.maphg and p1.tenphg = 'Nghien Cuu'
Group by n.manv , n.tennv
Having COUNT(DISTINCT pc.mada) = ( select COUNT(DISTINCT d2.mada)
    from DeAn d2, PhongBan p2
    where d2.phong=p2.maphg and p2.tenphg = 'Nghien Cuu'
    )
```

Results Messages

manv	tennv
NV01	Tien

VII. Các loại truy vấn khác

1. Truy vấn con ở mệnh đề SELECT

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

```
-----Với mỗi nhân viên, cho biết họ, tên nhân viên và số thân nhân của họ
select n.honv , n.tennv , (
    select COUNT(*)
    from ThanNhan t
    where t.manv = n.manv
) as SoThanNhan
from NhanVien n
```

Results Messages

	honv	tennv	SoThanNhan
1	Dinh	Tien	3
2	Vo	Tung	3
3	Tran	Tam	0
4	Vo	Hung	0
5	Le	Qu...	0
6	Tran	Kh...	0
7	Le	Nhan	1
8	Tran	Qu...	0
9	Bui	Vu	0

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

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

```

select p.tenphg,      (   select AVG(luong)
                        from NhanVien n
                        where n.phg = p.mapgh
                        ) as LuongTB
from PhongBan p

```

	tenphg	LuongTB
1	Quan Ly	46000.00
2	Dieu Hanh	29333.3333
3	Nghien Cuu	37666.6666

2. 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 30000

-----Cho biết những phòng ban (TENPHG) có lương trung bình của các nhân viên lớn hơn 30000

```

select tenphg, Tam.LuongTB
from PhongBan, (select phg, AVG(luong) as LuongTB
                from NhanVien
                Group by phg
                Having AVG(Luong) > 30000) as Tam
where mapgh = Tam.phg

```

	tenphg	LuongTB
1	Quan Ly	46000.00
2	Nghien Cuu	37666.6666

3. Điều kiện kết ở mệnh đề FROM

VD:

- Tìm mã và tên các nhân viên làm việc tại phòng 'Nghien cuu'

----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'
```

	manv	tennv
1	NV01	Tien
2	NV02	Tung
3	NV07	Nhan

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

----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 = trgphg
```

	tennv	honv	tenphg
1	Tien	Dinh	NULL
2	Tung	Vo	Nghien Cuu
3	Tam	Tran	NULL
4	Hung	Vo	NULL
5	Quyen	Le	Quan Ly
6	Khang	Tran	NULL
7	Nhan	Le	NULL
8	Quang	Tran	Dieu Hanh
9	Vu	Bui	NULL

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

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

```

select n.tennv, d.tenda
from (PhanCong pc join DeAn d on pc.mada = d.mada )
Right join NhanVien n on pc.manv = n.manv

```

	tennv	tenda
1	Tien	San pham X
2	Tien	San pham Y
3	Tien	San pham Z
4	Tung	San pham Z
5	Tung	Tin hoc hoa
6	Tung	Cap quang
7	Tam	NULL
8	Hung	San pham Z
9	Quyen	Tin hoc hoa
10	Quyen	Cap quang
11	Khang	NULL
12	Nhan	Dao tao
13	Quang	Tin hoc hoa
14	Quang	Dao tao
15	Vu	NULL

4. Cấu trúc Case

- Cho biết họ tên các nhân viên đã đến tuổi về hưu (nam 50 tuổi, nữ 40 tuổi)

```
-----Cho biết họ tên các nhân viên đã đến tuổi về hưu (nam 50 tuổi, nữ 40 tuổi)
select honv, tennv, phai, YEAR(getdate()) - YEAR(ngsinh)
from NhanVien
where ( YEAR(getdate()) - YEAR(ngsinh) )
      >= ( CASE phai when 'Nam' then 50
            when 'Nu' then 40
            END
        )
```

	honv	tennv	phai	(No column name)
1	Vo	Tung	Nam	54
2	Le	Nhan	Nu	48
3	Bui	Vu	Nu	51

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

```
-----Cho biết họ tên các nhân viên và năm về hưu
select n.manv , n.honv , n.tennv,
      ( CASE Phai
        when 'Nam' then YEAR(ngsinh) + 50
        when 'Nu' then YEAR(ngsinh) + 40
        END
      ) as NamVeHuu
from NhanVien n
```

	manv	honv	tennv	NamVeHuu
1	NV01	Dinh	Tien	2015
2	NV02	Vo	Tung	2005
3	NV03	Tran	Tam	2022
4	NV04	Vo	Hung	2012
5	NV05	Le	Quyen	2017
6	NV06	Tran	Khang	2012
7	NV07	Le	Nhan	2001
8	NV08	Tran	Quang	2029
9	NV09	Bui	Vu	1998