

# Ngôn ngữ SQL – Phần 3

---

## I. Mục lục

I.	Mục lục.....	1
II.	Sử dụng alias trong câu truy vấn.....	2
III.	Một số lưu ý về phép kết .....	2
A.	Inner joins (Kết bằng) .....	3
B.	Right (Outer) joins (Kết phải) .....	4
C.	Left (Outer) joins (Kết trái) .....	5
D.	Full (Outer) joins .....	5
IV.	Phép chia, phép hội, phép giao và phép trừ .....	7
A.	Phép chia .....	7
B.	Phép hội (UNION).....	8
C.	Phép giao (Intersect) .....	9
D.	Phép trừ .....	10

## II. Sử dụng alias trong câu truy vấn

- Cách khai báo và sử dụng alias :

Khai báo : <TênBảng> **as** <TênAlias> hoặc <TênBảng> <TênAlias>

Sử dụng : <TênAlias>.<TênThuộcTÍNH>

- Ví dụ

NHANVIEN(MaNV, TenNV, MaPhong, MaNQL)

PHONGBAN(MaPhong, TenPhong)

- Cho biết danh sách các nhân viên cùng với tên phòng ban của nhân viên

```
SELECT nv.HoNV, nv.TenNV, pb.TenPhong
FROM PHONGBAN pb, NHANVIEN nv
Where pb.MaPhong = nv.MaPhong
```

Hoặc

```
SELECT nv.HoNV, nv.TenNV, pb.TenPhong
FROM PHONGBAN as pb, NHANVIEN as nv
Where pb.MaPhong = nv.MaPhong
```

- Nên sử dụng alias trong các trường hợp sau :
  - Khi câu truy vấn lấy thông tin từ nhiều bảng (từ 2 bảng trở lên) → dùng alias để đọc câu truy vấn dễ dàng hơn, và khi các thuộc tính ở các bảng trùng tên → dùng alias để phân biệt các thuộc tính giống nhau ở các bảng.
  - Khi tên bảng trong câu truy vấn là quá dài → dùng alias để câu truy vấn gọn hơn
- Bắt buộc sử dụng alias → Khi sử dụng nhiều thể hiện của một bảng trong câu truy vấn → **bắt buộc** phải sử dụng alias
- Cho biết danh sách các nhân viên cùng với tên phòng ban của nhân viên

```
SELECT nv.HOTEN, nql.HOTEN
FROM NHANVIEN nv, NHANVIEN nql
WHERE nv.MANQL = nql.MANV
```

Bắt buộc sử dụng alias

⇒ **Lời khuyên** : Nên dùng alias nếu sau mệnh đề FROM có từ 2 thể hiện của bảng trở lên

## III. Một số lưu ý về phép kết

SINHVIEN

LOP

	malop	tenlop		masv	hoten	malop
1	L1	10A	1	01	A	L1
2	L2	10B	2	02	B	L2
3	L3	10C	3	03	C	L2
			4	04	D	L1
			5	05	E	L1

Yêu cầu: Cho biết sĩ số của mỗi lớp

### A. Inner joins (Kết bằng)

Phép kết Inner joins giữa 2 bảng A và B → là một bảng C = { các bộ trong đó mỗi bộ là sự kết hợp của các bộ trong A với các bộ trong B sao cho điều kiện kết được thỏa mãn }

Phép kết giữa SINHVIEN và LOP

- Phép kết bằng được diễn đạt trong mệnh đề where như sau

```
SELECT *
FROM SINHVIEN sv, LOP l
WHERE sv.malop = l.malop
```

Điều kiện kết

- Phép kết bằng được diễn đạt bằng cách sử dụng join

```
select *
from SINHVIEN sv join LOP l on sv.malop = l.malop
```

Điều kiện kết

Kết quả

	masv	hoten	malop	malop	tenlop
1	01	A	L1	L1	10A
2	02	B	L2	L2	10B
3	03	C	L2	L2	10B
4	04	D	L1	L1	10A
5	05	E	L1	L1	10A

Nhận xét : Thông tin về lớp 10C bị mất

- Tính sĩ số của lớp

1

```
select l.malop, l.tenlop, count(*) as siSo
from SINHVIEN sv join LOP l on sv.malop = l.malop
group by l.malop, l.tenlop
```

Điều kiện kết

Kết quả :

	malop	tenlop	SiSo
1	L1	10A	3
2	L2	10B	2

Nhận xét : Sĩ số của lớp 10C (bằng 0) không được xuất ra, vì thông tin lớp 10C đã bị mất sau phép kết bảng

### B. Right (Outer) joins (Kết phải)

Phép kết Right Outer joins giữa 2 bảng A và B → là một bảng C = {các bộ trong đó mỗi bộ là sự kết hợp của các bộ trong A với các bộ trong B sao cho điều kiện kết được thỏa mãn} + {các bộ còn lại trong B mà không thỏa điều kiện kết với bất kỳ một bộ trong A nào}

- Phép kết Right (Outer) Joins giữa SINHVIEN và LOP

```
select *
from SINHVIEN sv right join LOP l on sv.malop = l.malop
```

Kết quả :

	masv	hoten	malop	malop	tenlop
1	01	A	L1	L1	10A
2	04	D	L1	L1	10A
3	05	E	L1	L1	10A
4	02	B	L2	L2	10B
5	03	C	L2	L2	10B
6	NULL	NULL	NULL	L3	10C

Nhận xét : Thông tin về lớp 10C vẫn được giữ lại sau phép kết phải

- Tính sĩ số của lớp

2

```
select l.malop, l.tenlop, count(sv.MaLop) as SiSo
from SINHVIEN sv right join LOP l on sv.malop = l.malop
group by l.malop, l.tenlop
```

Kết quả :




	malop	tenlop	SiSo
1	L1	10A	3
2	L2	10B	2
3	L3	10C	0

Nhận xét : Sĩ số của các lớp không có học sinh (10 C) vẫn được xuất ra (vì phép kết không mất thông tin về lớp)

Lưu ý :

- join = inner join
- left join = left outer join
- right join = right outer join

Câu hỏi :

1. Tại sao  và  lại khác nhau ?
2. Tại sao  là `count (sv.MaLop)` chứ không phải là `count (l.MaLop)`

(làm sao mà biết được ,vay cùng hỏi)

### C. Left (Outer) joins (Kết trái)

Phép kết Left (Outer) joins giữa 2 bảng A và B → là một bảng C = {các bộ trong đó mỗi bộ là sự kết hợp của các bộ trong A với các bộ trong B sao cho điều kiện kết được thỏa mãn} + {các bộ còn lại trong A mà không thỏa điều kiện kết với một bộ bất kỳ trong B nào}

### D. Full (Outer) joins

Phép kết Full Outer joins giữa 2 bảng A và B → là một bảng C = {các bộ trong đó mỗi bộ là sự kết hợp của các bộ trong A với các bộ trong B sao cho điều kiện kết được thỏa mãn} + {các bộ còn lại trong A mà không thỏa điều kiện kết với bất kỳ một bộ trong B nào} + {các bộ còn lại trong B mà không thỏa điều kiện kết với bất kỳ một bộ trong A nào}

#### PUBLISHER

	pid	pname	pcity
1	1	Algodata Infosystems 1	MIA
2	2	Algodata Infosystems 2	NYO
3	3	Algodata Infosystems 3	MAN

#### AUTHORS

	auid	firstname	lastname	city
1	1	Reginald	Blotchett-Halls	NYO
2	2	Michel	DeFrance	OAS
3	3	Innes	del Castillo	CAN
4	4	Ann	Dull	LND
5	5	Marjorie	Green	CAL
6	6	Morningstar	Greene	CAL
7	7	Burt	Gringlesby	LOS
8	8	Sheryl	Hunter	NYO

```
select *
from AUTHORS full join PUBLISHERS on city=pcity
```

Kết quả :

	auid	first...	lastname	city	pid	pname	pcity
1	1	Reginald	Blotch...	NYO...	2	Algoda...	NYO
2	2	Michel	DeFrance	OAS...	NULL	NULL	NULL
3	3	Innes	del Ca...	CAN...	NULL	NULL	NULL
4	4	Ann	Dull	LND...	NULL	NULL	NULL
5	5	Marjorie	Green	CAL...	NULL	NULL	NULL
6	6	Morni...	Greene	CAL...	NULL	NULL	NULL
7	7	Burt	Gringlesby	LOS...	NULL	NULL	NULL
8	8	Sheryl	Hunter	NYO...	2	Algoda...	NYO
9	NULL	NULL	NULL	NULL	3	Algoda...	MAN
10	NULL	NULL	NULL	NULL	1	Algoda...	MIA

- Cho biết những tác giả và nhà xuất bản ở cùng thành phố
- Cho biết số lượng tác giả và nhà xuất bản ở cùng thành phố, số lượng tác giả mà không có nhà xuất bản nào ở cùng thành phố và số lượng nhà xuất bản mà không có tác giả nào ở cùng thành phố

```
select  count( case when auid is null then 1 end) as
        [Số tác giả không cùng thành phố với bất kỳ NXB nào],
        count( case when pid is null then 1 end) as
        [Số NXB không cùng thành phố với bất kỳ tác giả nào],
        count(case when city = pcity then 1 end) as
        [Số NXB và Tác giả cùng thành phố]
from    AUTHORS full join PUBLISHERS on city=pcity
```

## IV. Phép chia, phép hội, phép giao và phép trừ

### A. Phép chia

- Tìm nhân viên làm việc tất cả các đề án của công ty
  - **Bước 1** : Tìm các đề án của công ty mà nhân viên '001' chưa làm

Cách 1 : Dùng NOT EXISTS

```
select *
from DEAN da
where not exists (
    select *
    from PHANCONG pc
    where pc.MA_NVIEN = '001' and
          pc.MADA = da.MADA)
```

Cách 2 : Dùng NOT IN

```
select *
from DEAN
where MADA not in (
    select MADA
    from PHANCONG
    where MA_NVIEN='001')
```

- **Bước 2** : Nhận xét rằng : Nếu kết quả trả ra là không có bộ nào → Nhân viên '001' làm tất cả các đề án. Ngược lại, nếu kết quả trả ra là có từ 1 bộ dữ liệu trở lên → có đề án của công ty mà nhân viên '001' chưa làm → nhân viên '001' không làm mọi đề án của công ty.

Do vậy, Tìm các nhân viên làm mọi đề án của công ty tương đương với việc, kiểm tra từng nhân viên, nếu danh sách các đề án của công ty nhân viên đó chưa làm là rỗng (không có bộ nào) → nhân viên làm mọi đề án của công ty, ngược lại thì nhân viên đó không làm mọi đề án của công ty.

Cách 1 : Nếu bước 1 sử dụng NOT EXISTS

```

select *
from NHANVIEN nv
where not exists (
    select *
    from DEAN da
    where not exists (
        select *
        from PHANCONG pc
        where pc.MA_NVIEN = nv.MANV and
              pc.MADA = da.MADA))

```

Cách 2 : Nếu bước 1 sử dụng NOT IN

```

select *
from NHANVIEN nv
where not exists (
    select *
    from DEAN
    where MADA not in (
        select MADA
        from PHANCONG
        where MA_NVIEN=nv.MANV))

```

## B. 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



```
select * from table1
union
select * from table2
```

```
select * from table1
union all
select * from table2
```

**Kết quả:**

a	b
1	2
1	6
2	3
2	7
3	4
4	5

**Kết quả :**

a	b
1	2
3	4
2	3
4	5
2	7
3	4
2	3
1	6

### C. 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

#### D. 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)
```