

Bài thực hành 3: TRUY VẤN DỮ LIỆU

I/ LÝ THUYẾT:

Sau khi chèn được dữ liệu vào các bảng (các bảng này có liên kết với nhau), Ta bắt đầu sử dụng câu lệnh truy vấn.

1/ Cấu trúc câu lệnh truy vấn dữ liệu tổng quát

```
Select [Ten_Bang.]Ten_Cot[,...]
From Ten_Bang1 [<liên kết> Ten_Bang2], ...
Where <Điều kiện>
Group by <các cột gom nhóm>
Having <Điều kiện lọc nhóm>
Order by [Ten_Bang.]Ten_cot [asc/desc,...]
```

Note:

+ Khi chọn cột trong “Select” ta chỉ cần viết tên cột (dù các cột ở các bảng) nếu 2 hoặc nhiều bảng có trùng tên cột (như Masv ở cả bảng Sinhvien và bảng Ketqua) thì ta phải chỉ rõ cột định sử dụng là cột ở bảng nào (ví dụ Sinhvien.Masv).

+ < điều kiện>: là điều kiện đưa vào để chọn lọc dữ liệu, thường gồm:

- Các phép toán so sánh: < , > , ≤ , ≥ , ≠ , =
- Các phép toán logic: AND, OR, và NOT
- Các từ khóa: BETWEEN ... AND, IN, EXISTS, LIKE...

+ **Ví dụ 1:** Cho danh sách sinh viên gồm Mã SV, Họ tên, Ngày sinh của tất cả các sinh viên

```
Select MaSV, TenSV, Ngaysinh
from Sinhvien
```

+ **Ví dụ 2:** Cho danh sách sinh viên nữ thuộc lớp L02

```
Select *
From Sinhvien
Where (GT=N'Nữ') and (Lop='L02')
```

Question: Chạy thử truy vấn dưới, xem kết quả và giải thích?

```
Select *
From Sinhvien
Where (GT='Nữ') and (Lop='L02')
```

+ **Ví dụ 3:** Thêm định danh cho bảng và cho cột (sử dụng từ khóa as)

```
select sv.MaSV, TenSV as 'Tên Sinh Viên'
from Sinhvien as sv /* sau khi định danh cho bảng thì từ đó sử dụng tên định danh*/
```

+ **Ví dụ 4:** Xử lý dữ liệu kiểu chuỗi dùng Like hoặc not like để tìm chuỗi gần đúng

?: Thay thế 1 chuỗi

-: thay thế 1 ký tự

VD 4.1: Cho danh sách các sinh viên có tên là 'Nga'

```
select *  
from sinhvien  
where TenSV like N'% Nga'
```

VD 4.2: Cho danh sách những sinh viên họ Trần

```
select *  
from sinhvien  
where TenSV like N'Trần %'
```

+ **Ví dụ 5:** Sử dụng BETWEEN <GT1> AND <GT2> để lấy những bản ghi thỏa mãn nằm giữa GT1 và GT2.

Cho danh sách gồm mã sinh viên, tên môn học của sinh viên có điểm từ 5 đến 7

```
select MaSV, MaMH  
from ketqua  
where diem between 5 and 7
```

Note: Not Between .. and ... có ý nghĩa ngược lại.

+ **Ví dụ 6:** Sử dụng NULL/NOT NULL cho các trường hợp sau:

- Không biết - không xác định (value unknown)
- Không thể áp dụng (value inapplicable)
- Không tồn tại (value withheld)

VD: Cho danh sách gồm Mã Sinh viên và tên môn học của sinh viên không có điểm thi

```
select MaSV, MaMH  
from ketqua  
where Diem is Null
```

+ **Ví dụ 7: Sử dụng in:** Cho danh sách các sinh viên có quê ở Hà Nội hoặc Phú Thọ

```
Select *  
From Sinhvien  
where que in (N'Hà nội', N'Phú Thọ')
```

+ **Ví dụ 8: Sử dụng các toán tử**

- Distinct : loại bỏ các dòng dữ liệu trùng
- Top n : lấy n dòng đầu tiên

VD 8.1: Liệt kê danh sách các tỉnh có sinh viên theo học tại trường

```
select distinct Que  
from Sinhvien
```

Note: Chạy thử câu lệnh không có distinct để so sánh kết quả. Giải thích.

VD 8.2: Cho thông tin của hai sinh viên đầu tiên trong danh sách sinh viên

```
select top (2) MaSV, TenSV, Que  
from Sinhvien
```

QUESTION: chỉ ra mối liên hệ giữa phép chọn và chiếu trong đại số quan hệ với câu lệnh truy vấn SQL?

2. Kết nối dữ liệu

Ở trên ta đã truy vấn dữ liệu trong 1 bảng theo câu lệnh đơn giản (sử dụng “select *” để lấy hết các cột). Khi dữ liệu nằm trên nhiều bảng ta cần tạo kết nối giữa nhiều bảng đó nhằm tránh dư thừa dữ liệu.

Ví dụ, với yêu cầu “cho danh sách MaSV, TenSV, TenMH và Diem của các sinh viên” ta có thể sử dụng các kiểu kết nối sau:

+ Kết nối sử dụng mệnh đề where

```
select sv.MaSV, TenSV as 'Tên Sinh Viên', TenMH, Diem
from Sinhvien as sv, Monhoc as Mh, Ketqua as kq
where sv.MaSV = kq.MaSV and kq.MaMH = mh.MaMH
```

+ Kết nối dùng JOIN hay là inner join

```
select sv.MaSV, TenSV as 'Tên Sinh Viên', TenMH, Diem
from ((Sinhvien as sv join Ketqua as kq on sv.MaSV = kq.MaSV) join Monhoc as Mh on
kq.MaMH=mh.MaMH)
```

+ Kết nối dùng LEFT JOIN

```
select sv.MaSV, TenSV as 'Tên Sinh Viên', TenMH, Diem
from ((Sinhvien as sv left join Ketqua as kq on sv.MaSV = kq.MaSV) left join
Monhoc as Mh on kq.MaMH=mh.MaMH)
```

+ Kết nối dùng RIGHT JOIN

```
select sv.MaSV, TenSV as 'Tên Sinh Viên', TenMH, Diem
from ((Sinhvien as sv right join Ketqua as kq on sv.MaSV = kq.MaSV) right join
Monhoc as Mh on kq.MaMH=mh.MaMH)
```

+ Kết nối dùng FULL JOIN

```
select sv.MaSV, TenSV as 'Tên Sinh Viên', TenMH, Diem
from ((Sinhvien as sv full join Ketqua as kq on sv.MaSV = kq.MaSV) full join
Monhoc as Mh on kq.MaMH=mh.MaMH)
```

Question: Sự khác nhau khi sử dụng kết nối dữ liệu từ nhiều bảng theo các cách trên (sinh viên tự kết nối theo các cách trên theo 2 bảng hoặc 3 bảng để xem sự khác biệt)

3/ Sử dụng cấu trúc lệnh trong Select

Việc sử dụng cấu trúc lệnh tiện lợi khi muốn hiển thị thêm thông tin trong bảng truy vấn. cấu trúc này sẽ hiển thị thành 1 trường mới, trường này không phải trường ta tạo ban đầu mà là trường truy vấn thể hiện khung nhìn mà người sử dụng quan tâm.

+ Cú pháp:

Select <d/s tên trường> ,

Case

When <dk1> then <chuỗi 1>

When <dk2> then <chuỗi 2>

.....

Else <chuỗi n>

End as <tên trường mới>

From <tên bảng>

+ Ví dụ 1: bổ sung cột thông tin, nếu Gt là nam thì ghi 'Đây là nam', ngược lại thì ghi 'đây là nữ'

```
select Masv, TenSV,  
case  
    when Gioitinh=N'NAM' then N'đây là nam'  
    when Gioitinh =N'Nữ' then N'Đây là nữ'  
    else N'Không Xác định'  
end as 'thông tin'  
from Sinhvien
```

Kết quả:

	MaSV	TenSV	Thông tin
1	1	Trần Bảo Trọng	Đây là nam
2	2	Lê Thùy Dung	Đây là nữ
3	3	Phạm Trung Tính	Đây là nam
4	4	Lê Trường An	Đây là nam
5	5	Phạm Thị Hương Giang	Đây là nữ
6	6	Đoàn Duy Thức	Đây là nam
7	7	Dương Tuấn Thông	Đây là nam
8	8	Lê Thành Đạt	Đây là nam

+ Ví dụ 2: Cho danh sách gồm mã sinh viên, tên SV, tên môn học, điểm và cột xếp loại. Thông tin trên cột xếp loại như sau: điểm \geq 8 xếp loại giỏi, 7= \leq điểm \leq 8: loại khá, 5= \leq điểm \leq 7: loại Trung bình, điểm \leq 5: thi lại.

```
Select Sinhvien.MaSV as 'Mã Sinh viên',TenSV as 'Họ và tên', TenMH as 'Tên môn',Diem as 'Điểm',  
Case  
    when (Diem<5) then N'Thi lại'  
    when (Diem >=5) and (Diem<7) then N'Trung bình'  
    when (Diem>=7) and (Diem<8) then N'Khá'  
    else N'Giỏi'  
end as N'Xếp Loại'  
from Sinhvien, Monhoc, Ketqua  
where (Sinhvien.MaSV = Ketqua.MaSV) and (Monhoc.MaMH = Ketqua.MaMH)
```

4/ Sử dụng phép toán tập hợp trong truy vấn:

Gồm:	UNION	Hợp
	INTERSECT	Giao
	EXCEPT	Trừ

Chú ý: Các trường trong select phải giống nhau mới thực hiện được

+ Ví dụ 1: Cho danh sách mã sinh viên của những sinh viên không có bất kỳ một điểm thi của một môn học nào.

```
select MasV  
from Sinhvien
```

```
Except
select Masv
from ketqua
```

+ Ví dụ 2: Cho danh sách tên của các sinh viên có đủ điểm hai môn với mã môn học là 1 và 3

```
select distinct tensv
from Sinhvien, Ketqua, Monhoc
where Sinhvien.MaSV = Ketqua.MaSV and Ketqua.MaMH = Monhoc.MaMH and Monhoc.MaMH=1
Intersect
select distinct tensv
from Sinhvien, Ketqua, Monhoc
where Sinhvien.MaSV = Ketqua.MaSV and Ketqua.MaMH = Monhoc.MaMH and Monhoc.MaMH=3
```

5/ Sắp xếp theo thứ tự tăng, giảm: Sử dụng ORDER BY <ds cột [asc/desc]>. Mặc định là ASC, thứ tự sắp xếp các cột ưu tiên từ trái qua phải.

+ Ví dụ 1: Cho danh sách sắp xếp theo chiều giảm dần của Điểm thi, nếu điểm thi trùng nhau thì sắp theo chiều tăng dần của mã Sinh viên.

```
select * from Ketqua
order by Diem DESC, MaSV
```

+ Ví dụ 2: Cho danh sách gồm tên SV, Tên môn học và điểm Cho danh sách sắp xếp theo chiều giảm dần của Điểm thi

```
select TenSV, TenMH, Diem
from sinhvien, Monhoc, Ketqua
where Sinhvien.MaSV = Ketqua.MaSV and Ketqua.MaMH = Monhoc.MaMH
order by Diem DESC
```

6/ Sử dụng hàm trong truy vấn: Trong truy vấn cho phép ta sử dụng hàm để lọc hay tìm kiếm dữ liệu. Một số hàm thường dùng:

+ Hàm ngày tháng: dùng để xử lý dữ liệu kiểu ngày/tháng

- **GETDATE()**: Lấy giá trị ngày hiện tại của hệ thống
- **DAY(date), MONTH(date), YEAR(date)**: cho ngày, tháng, năm tương ứng

+ Hàm xử lý chuỗi:

- **LEFT(S, n), RIGHT(S, n)**: Lấy n ký tự bên trái hoặc bên phải chuỗi S
- **LEN(S)**: chiều dài chuỗi S
- **REPLACE(S1, S2, S3)**: thay thế trong chuỗi 1, các chuỗi 2 thành chuỗi 3

+ Hàm thống kê (thường sử dụng trong mệnh đề Group by)

- **SUM(Tên cột)**: Tính tổng
- **AVG(Tên cột)**: Tính trung bình cộng
- **COUNT (Tên cột) | COUNT(*)**: Đếm
- **MAX(Tên cột)**: Tìm giá trị lớn nhất
- **MIN(Tên cột)**: Tìm giá trị nhỏ nhất

+ **Ví dụ 1:** Cho danh sách các sinh viên có năm sinh 1995

```
select *
```

```
from sinhvien
where year(Ngaysinh)=1995
```

+ **Ví dụ 2:** Cho danh sách các sinh viên có tuổi trên 22

```
select *
from sinhvien
where year(getdate())-year(ngaysinh)>22
```

+ **Ví dụ 3:** Cho điểm cao nhất, điểm thấp nhất của cả trường

```
select max(Diem) as 'Điểm cao nhất', min(Diem) as 'Điểm thấp nhất'
from ketqua
```

Question: Cho biết câu lệnh truy vấn sau thực hiện yêu cầu gì?

```
select max(Diem) as 'Điểm cao nhất', min(Diem) as 'Điểm thấp nhất'
from ketqua,sinhvien
where Sinhvien.MaSV= Ketqua.MaSV and Lop='L01'
```

7/ Mệnh đề gom nhóm dữ liệu Group by: dùng để thực hiện gom nhóm các bộ theo một tiêu chí xác định để thực hiện tính toán. Ví dụ: đếm số sinh viên của mỗi lớp, tính điểm trung bình của các lớp, cho danh sách sinh viên có điểm cao nhất của mỗi lớp vv...

■ Cú pháp:

```
SELECT <ds cột gom nhóm>, <hàm tính toán>
FROM <danh sách bảng>
[WHERE <Điều kiện>]
GROUP BY <Ds cột gom nhóm>
[HAVING <Điều kiện tính toán nhóm>]
```

Chú ý:

- Mệnh đề GROUP BY: Các thuộc tính không nằm trong hàm nhóm của mệnh đề SELECT phải được xuất hiện trong mệnh đề GROUP BY
- Mệnh đề HAVING: Chỉ kiểm tra điều kiện trên nhóm, mệnh đề Having không lọc trên từng bản ghi. Điều kiện trên nhóm (ở mệnh đề HAVING) được thực hiện sau khi gom nhóm.
- Thứ tự thực hiện câu truy vấn có mệnh đề GROUP BY và HAVING:
 - Chọn ra những dòng thỏa điều kiện trong mệnh đề WHERE
 - Những dòng này sẽ được gom thành nhiều nhóm tương ứng với mệnh đề GROUP BY
 - Áp dụng các hàm kết hợp cho mỗi nhóm
 - Bỏ qua những nhóm không thỏa điều kiện trong mệnh đề HAVING
 - Rút trích các giá trị của các cột và hàm kết hợp trong mệnh đề SELECT

+ Ví dụ 1: Cho biết số sinh viên của mỗi lớp

```
select Lop, count(Masv) as 'Số SV'
from Sinhvien
group by Lop
```

+ Ví dụ 2: Thống kê số lượng sinh viên nam/nữ của cả trường.

```
Select GT, Count(*) as 'Số SV'  
from Sinhvien  
group by GT
```

+ Ví dụ 3: Thống kê điểm cao nhất, điểm thấp nhất theo từng lớp

```
Select Lop, max(Diem) as 'Điểm cao nhất' , min(Diem) as 'Điểm thấp nhất'  
from Sinhvien, Ketqua, monhoc  
where Sinhvien.MaSV= Ketqua.MaSV and Monhoc.MaMH = Ketqua.MaMH  
group by Lop
```

+ Ví dụ 4: Thống kê số lượng sinh viên phải thi lại của mỗi lớp

```
Select Lop, Count (Sinhvien.MaSV) as 'Số SV thi lại'  
from Sinhvien, Ketqua  
where Sinhvien.MaSV = Ketqua.MaSV and Ketqua.MMh = Monhoc.MaMH and Diem<5  
group by Lop
```

+ Ví dụ 6: Thống kê số lượng sinh viên thi lại theo từng môn

```
select TenMH, count (MaSV) as 'Số SV thi lại'  
from Monhoc, ketqua  
where Monhoc.MaMH = Ketqua.MaMH and Diem<5  
group by TenMH
```

+ Ví dụ 7: Đưa ra danh sách các lớp có số sinh viên trên 2 sinh viên

```
Select Lop, count(Masv) as 'Số Sv'  
from Sinhvien  
group by Lop  
Having count(Masv)>2
```

+ Ví dụ 8: Đưa ra danh sách các lớp có số sinh viên thi lại trên 1 sinh viên


```
Select Lop, count(Sinhvien.MaSV) as 'Số SV'  
from Sinhvien, ketqua, monhoc  
where Sinhvien.MaSV=Ketqua.Masv and Monhoc.Mamh = Ketqua.MaMH and Diem<5  
group by Lop  
having count(Sinhvien.MaSV)>1
```

8/ Truy vấn lồng:

- Là thực hiện lồng ghép nhiều câu lệnh SELECT với nhau
- Câu truy vấn con thường trả về một tập các giá trị
- Các câu truy vấn con trong cùng một mệnh đề WHERE được kết hợp bằng phép nối logic
- Mệnh đề WHERE của câu truy vấn cha
 - <biểu thức> <so sánh tập hợp> <truy vấn con>
 - So sánh tập hợp thường đi cùng với một số toán tử
 - IN, NOT IN
 - ALL
 - ANY hoặc SOME
 - Kiểm tra sự tồn tại
 - EXISTS


- NOT EXISTS

■ Cú pháp truy vấn lồng:

Truy
vấn cha  **SELECT** <danh sách các cột>

FROM <danh sách các bảng>

WHERE <biểu thức> <so sánh tập hợp> (

Truy
vấn con  **SELECT** <danh sách các cột>

FROM <danh sách các bảng>

WHERE <điều kiện>)

■ Phân loại: có 2 loại truy vấn lồng

- **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
- **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 1: Cho danh sách sinh viên có điểm môn học 1 cao nhất

Cách 1:

```
select a.Masv, Tensv, Diem
from sinhvien as a join ketqua as b on a.MaSV=b.Masv
where MaMH=1 and Diem = (select Max(diem) from ketqua where MaMH=1)
```

Cách 2:

```
select a.Masv, Tensv, Diem
from sinhvien as a join ketqua as b on a.MaSV=b.Masv
where MaMH=1 and Diem >= All (select Diem from ketqua where MaMH=1)
```

Question: Cho biết truy vấn sau cho kết quả có gì khác so với truy vấn trước, tại sao?

```
select a.Masv, Tensv, Diem
from sinhvien as a join ketqua as b on a.MaSV=b.Masv
where Diem = (select Max(diem) from ketqua where MaMH=1)
```

+ Ví dụ 2: Cho danh sách sinh viên không bị thi lại môn nào

Cách 1: Dùng phép toán tập hợp

```
Select a.Masv, TenSv
From (sinhvien as a join ketqua as b on a.Masv=b.masv) join monhoc as c on c.Mamh = b.mamh
except
select a.Masv, tenSV
From (sinhvien as a join ketqua as b on a.Masv=b.masv) join monhoc as c on c.Mamh = b.mamh
where diem<5
```

Question 1: Cho biết truy vấn này khác với truy vấn trên chỗ nào? Truy vấn nào sẽ xử lý dữ liệu chính xác hơn?

```
Select a.Masv, TenSv
```



```

From (sinhvien as a join ketqua as b on a.Masv=b.masv) join monhoc as c on c.Mamh = b.mamh
where diem is not Null
except
select a.Masv, tenSV
From (sinhvien as a join ketqua as b on a.Masv=b.masv) join monhoc as c on c.Mamh = b.mamh
where diem<5

```

Question 2: Cho biết truy vấn này sai ở đâu? Tại sao?

```

Select a.Masv, TenSv, Diem
From (sinhvien as a join ketqua as b on a.Masv=b.masv) join monhoc as c on c.Mamh = b.mamh
where diem is not Null
except
select a.Masv, tenSV, Diem
From (sinhvien as a join ketqua as b on a.Masv=b.masv) join monhoc as c on c.Mamh = b.mamh
where diem<5

```

Cách 2: Dùng truy vấn lồng

```

Select a.Masv, TenSv, Que
From sinhvien a, ketqua b, monhoc c
where a.Masv=b.Masv and b.Mamh=c.Mamh and a.Masv not in (Select a.MaSV
From sinhvien a, ketqua b, monhoc c
Where a.Masv=b.Masv and b.Mamh=c.Mamh and Diem<5)

```

II. BÀI TẬP

Bài 1: Cho CSDL QLSV ở trên, sử dụng các truy vấn để thực hiện các yêu cầu sau:

1. Cho biết mã môn học, tên môn học, điểm thi tất cả các môn của sinh viên tên Thức
2. Cho biết mã môn học, tên môn và điểm thi ở những môn mà sinh viên tên Dung phải thi lại (điểm<5)
3. Cho biết mã sinh viên, tên những sinh viên đã thi ít nhất là 1 trong 3 môn Lý thuyết Cơ sở dữ liệu, Tin học đại cương, mạng máy tính.
4. Cho biết mã môn học, tên môn mà sinh viên có mã số 1 chưa có điểm
5. Cho biết điểm cao nhất môn 1 mà các sinh viên đạt được
6. Cho biết mã sinh viên, tên những sinh viên có điểm thi môn 2 không thấp nhất khoa
7. Cho biết mã sinh viên và tên những sinh viên có điểm thi môn 1 lớn hơn điểm thi môn 1 của sinh viên có mã số 3
8. Cho biết số sinh viên phải thi lại môn Toán Cao cấp
9. Đối với mỗi môn, cho biết tên môn và số sinh viên phải thi lại môn đó mà số sinh viên thi lại ≥ 2
10. Cho biết mã sinh viên, tên và lớp của sinh viên đạt điểm cao nhất môn Tin đại cương
11. Đối với mỗi lớp, lập bảng điểm gồm mã sinh viên, tên sinh viên và điểm trung bình chung học tập.
12. Đối với mỗi lớp, cho biết mã sinh viên và tên những sinh viên phải thi lại từ 2 môn trở lên
13. Cho biết mã số và tên của những sinh viên tham gia thi tất cả các môn.
14. Cho biết mã sinh viên và tên của sinh viên có điểm trung bình chung học tập ≥ 6

15. Cho biết mã sinh viên và tên những sinh viên phải thi lại ở ít nhất là những môn mà sinh viên có mã số 3 phải thi lại
16. Cho mã sv và tên của những sinh viên có hơn nửa số điểm ≥ 5 .
17. Cho danh sách tên và mã sinh viên có điểm trung bình chung lớn hơn điểm trung bình của toàn khóa.
18. *Cho danh sách mã sinh viên, tên sinh viên có điểm cao nhất của mỗi lớp.
19. *Cho danh sách tên và mã sinh viên có điểm trung bình chung lớn hơn điểm trung bình của lớp sinh viên đó theo học.

ĐỌC THÊM VỀ KẾT NỐI JOINS

Joins được sử dụng để lấy dữ liệu từ hai hay nhiều bảng, dựa trên một quan hệ logic giữa những bảng đó. Một join chỉ rõ khóa ngoài từ một bảng, và khóa chính liên kết với nó trong một bảng khác. Nó định nghĩa cách thức liên kết hai bảng trong một truy vấn. Để làm việc này, cần phải xác định:

Các cột từ mỗi bảng được sử dụng để join

Một thao tác logic như =, <>, được sử dụng để so sánh các giá trị từ các cột.

Cú pháp:

SELECT <select_list>

FROM Table_A [***AS*** Table_Alias_A]

<join_type>

Table_B [***AS*** Table_Alias_B]

ON <join_condition>

Trong đó,

Table_A, Table_B: Tên của các bảng được join

Table_Alias_A, Table_Alias_B: Tên viết tắt thay cho các bảng trong truy vấn

<join_type>: Chỉ ra cách các bảng được join

<join_condition>: Chỉ ra điều kiện join, tùy thuộc vào kiểu join

Có ba kiểu join:

☐ ☐ Inner

☐ ☐ Outer

☐ ☐ Cross

1. Inner Join

- Inner join kết hợp các bản ghi từ hai bảng và chỉ lấy ra những bản ghi nào thỏa mãn điều kiện sau ON. Khi sử dụng từ khóa JOIN trong một truy vấn, mà không nói rõ kiểu, theo mặc định SQL Server sẽ cho đó là một inner join. Inner Join tương đương với câu lệnh điều kiện Where.

2. Outer Join

Có ba kiểu Outer Join:

Left Outer Join

Left outer join trả lại tất cả các bản ghi từ bảng bên trái, và chỉ những bản ghi phù hợp từ bảng bên phải.

Chú ý: SQL Server hiển thị một giá trị NULL cho tất cả các bản ghi trong bảng thứ hai nếu bảng này không có các bản ghi phù hợp.

☐ ☐ Right Outer Join

Right outer join tương tự như left outer join, ngoại trừ rằng nó trả lại tất cả những bản ghi từ bảng bên phải, và chỉ những bản ghi phù hợp từ bảng bên trái.

Chúng ta chuyển ví dụ truy vấn left outer join thành right outer join. Tất cả những gì chúng ta cần làm là trao đổi các tên bảng, và các bí danh bảng tương ứng.

□ □ **FULL Join**

Full Join, hoặc Full Outer Join, liệt kê tất cả những bản ghi từ hai bảng, bất kể có các bản ghi phù hợp trong các bảng này hay không.

3. Cross Join

Cross join trả lại một tập kết quả là sự kết hợp của tất cả các hàng của hai bảng được join. Số lượng các hàng được trả lại là tích của số lượng các hàng trong hai bảng. Do vậy, kết quả là tích của các bảng được join.

Chú ý: Cross Join chính là phép tích đề các trong tập hợp, trong cross Join không có ***ON <join_condition>***

4. Seft Join

Đôi khi bạn muốn tìm các bản ghi trong một bảng được liên kết với các bản ghi khác trong cùng một bảng. Trong trường hợp như vậy, cần sử dụng *Seft-join*, là một kiểu của *inner join*.