

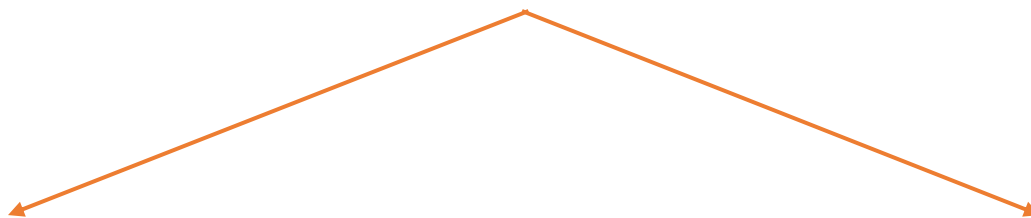


MCI



MAGIC CODE INSTITUTE

BUỔI 3 CHỦ ĐỀ KẾT HỢP BẢNG



Kết hợp bảng theo chiều ngang

(JOIN)

Kết hợp bảng theo chiều dọc

(UNION)



1

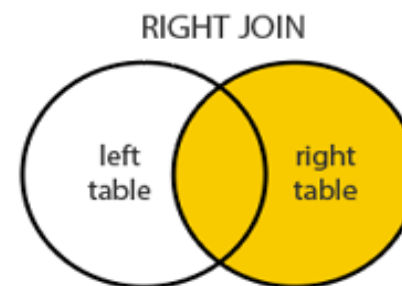
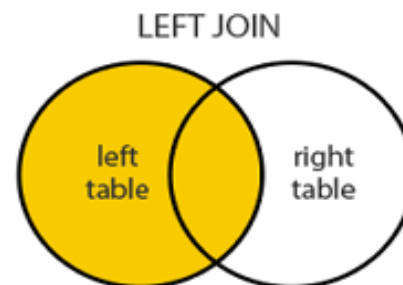
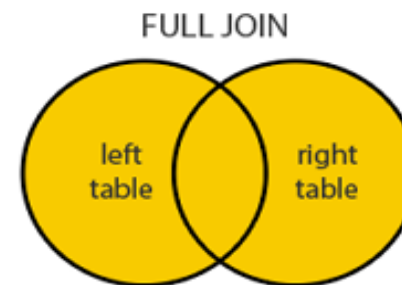
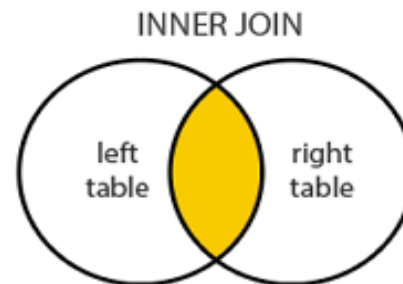
Giới thiệu Join trong SQL

1. Join là gì?

JOIN được sử dụng để kết hợp các dữ liệu từ hai hoặc nhiều bảng dựa trên mối liên quan giữa chúng

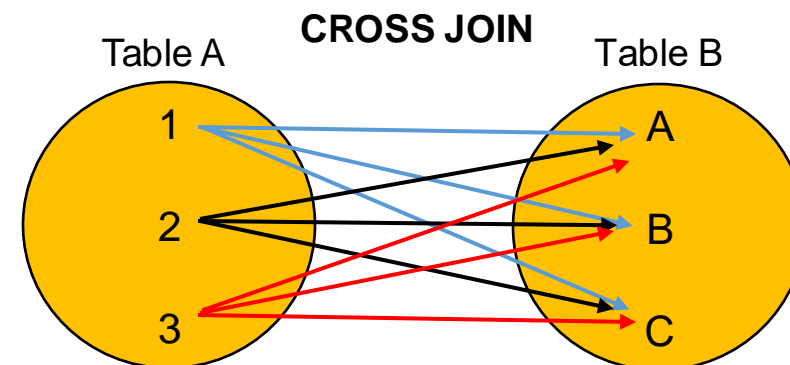
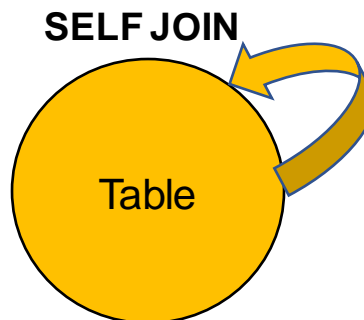
2. Các loại Join thường dùng

- **INNER JOIN** hay JOIN
- **LEFT OUTER JOIN** hay LEFT JOIN
- **RIGHT OUTER JOIN** hay RIGHT JOIN
- **FULL JOIN** hay FULL OUTER JOIN



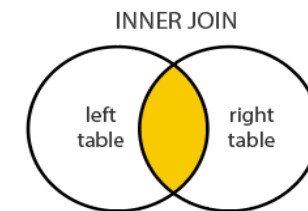
3. Một số loại Join khác

- **SELF JOIN**
- **CROSS JOIN**



2

Cách sử dụng và cấu trúc INNER JOIN



Kết quả:

Trả về các bản ghi có giá trị khớp trong cả hai bảng

Cấu trúc câu lệnh mẫu:

```
SELECT Column_name (s)
```

```
FROM Table_A
```

```
INNER JOIN Table_B
```

```
ON Table_A.Key_column_name = Table_B.Key_column_name
```

Group	Value
A	2
B	3
C	4
D	1

INNER
JOIN

Group	Order
C	6
D	7
E	8
F	9



Group	Value	Order
C	4	6
D	1	7

Ví dụ:

```
SELECT T1.manager_id
```

```
,T1.manager_name
```

```
,T2.region
```

```
FROM Managers as T1
```

```
INNER JOIN Profiles as T2
```

```
ON T1.manager_name=T2.manager
```

Bài tập thực hành:

Từ bảng Orders và Returns chỉ lấy ra những đơn returned, bao gồm những thông tin sau:

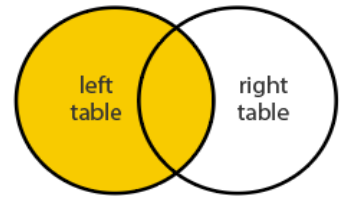
order_id, order_date, value, product_name, returned_date



3

Cách sử dụng và cấu trúc LEFT JOIN

LEFT JOIN



Kết quả:

Trả về tất cả các bản ghi từ bảng bên trái (A) và các bản ghi phù hợp từ bảng bên phải (B).

Kết quả là NULL từ bên phải, nếu không có kết quả phù hợp.

Cấu trúc câu lệnh mẫu:

```
SELECT Column_name (s)
```

```
FROM Table_A
```

```
LEFT JOIN Table_B
```

```
ON Table_A.Key_column_name = Table_B.Key_column_name
```

Group	Value
A	2
B	3
C	4
D	1

LEFT JOIN

Group	Order
C	6
D	7
E	8
F	9



Group	Value	Order
A	2	Null
B	3	Null
C	4	6
D	1	7

Ví dụ:

```
SELECT
```

```
T1.manager_id
```

```
,T1.manager_name
```

```
,T2.region
```

```
FROM Managers as T1
```

```
LEFT JOIN Profiles as T2
```

```
ON T1.manager_name=T2.manager
```

Bài tập thực hành:

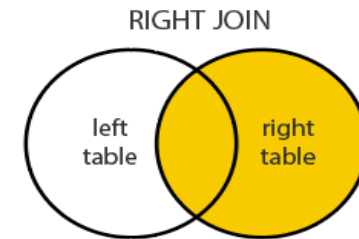
Từ bảng Orders và Returns lấy ra những đơn returns bao gồm những thông tin sau:

order_id, order_date, total_value, total_order_quantity, status, returned_date



4

Cách sử dụng và cấu trúc RIGHT JOIN



Kết quả:

Trả về tất cả các bản ghi từ bảng bên phải (B) và các bản ghi phù hợp từ bảng bên trái (A).

Kết quả là NULL từ bên trái, nếu không có kết quả phù hợp.

Cấu trúc câu lệnh mẫu:

```
SELECT Column_name (s)
```

```
FROM Table_A
```

```
RIGHT JOIN Table_B
```

```
ON Table_A.Key_column_name = Table_B.Key_column_name
```

Ví dụ:

```
SELECT
```

```
T1.manager_id
```

```
,T1.manager_name
```

```
,T2.region
```

```
FROM Managers as T1
```

```
RIGHT JOIN Profiles as T2
```

```
ON T1.manager_name=T2.manager
```

Group	Value
A	2
B	3
C	4
D	1

RIGHT JOIN

Group	Order
C	6
D	7
E	8
F	9



Group	Order	Value
C	6	4
D	7	1
E	8	Null
F	9	Null

Bài tập thực hành:

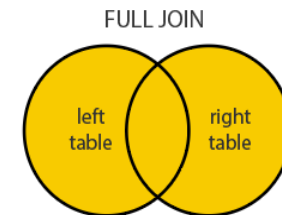
Từ bảng Orders và Returns lấy ra những đơn returns bao gồm những thông tin sau:

order_id, order_date, total_value, total_order_quantity, status, returned_date



5

Cách sử dụng và cấu trúc FULL OUTER JOIN



Kết quả:

Trả về tất cả các bản ghi khi có kết quả khớp trong bản ghi bảng trái (A) hoặc phải (B)

Cấu trúc câu lệnh mẫu:

```
SELECT Column_name (s)
FROM Table_A
```

```
FULL OUTER JOIN Table_B
```

```
ON Table_A.Key_column_name = Table_B.Key_column_name
```

Group	Value
A	2
B	3
C	4
D	1

FULL
OUTER
JOIN

Group	Order
C	6
D	7
E	8
F	9



Group	Value	Order
A	2	Null
B	3	Null
C	4	6
D	1	7
E	Null	8
F	Null	9

Ví dụ:

```
SELECT t1.manager_id, t1.manager_level,
t1.manager_name,
t1.manager_phone, t2.Manager, t2.province
FROM Managers as t1
FULL OUTER JOIN Profiles as t2
ON t1.manager_name = t2.manager
```

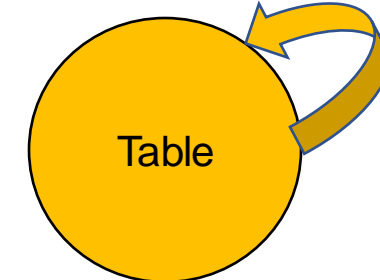
Bài tập thực hành:

Từ bảng Orders và Returns sử dụng Full Outer Join lấy ra thông tin order_id, order_date, total_value, total_order_quantity, status, returned_date

6

Cách sử dụng và cấu trúc SELF JOIN

SELF JOIN



SELF JOIN

SELF JOIN cho phép **join 1 bảng với chính nó**. Nó được sử dụng để truy vấn **dữ liệu phân cấp** hoặc **so sánh** các bản ghi trong cùng 1 bảng.

Cấu trúc câu lệnh mẫu

```
SELECT column_name (s)
FROM table_name AS T1, table_name AS T2
WHERE condition;
```

Ví dụ: Tìm các khách hàng có địa chỉ trong cùng tỉnh

```
SELECT T1.Province, T1.customer_name, T2.customer_name
FROM Orders AS T1, Orders AS T2
WHERE T1.province = T2.province
ORDER BY T1.Province, T1.customer_name, T2.customer_name;
```

id	Student_name	Friend_id	SELF JOIN	id	Student_name	Friend_id
1	A	3		1	A	3
2	B	1		2	B	1
3	C	2		3	C	2

id	Student_name	Friend_name
1	A	C
2	B	A
3	C	B

7

Cách sử dụng và cấu trúc CROSS JOIN

Cách sử dụng:

Mệnh đề **CROSS JOIN** dùng để truy vấn dữ liệu từ hai hay nhiều bảng không liên quan đến nhau.

Kết quả:

CROSS JOIN sẽ nối mỗi bản ghi của bảng đầu tiên (T1) với mỗi bản ghi của bảng thứ hai (T2).

Nói cách khác, **CROSS JOIN** trả về tích đề các bản ghi từ 2 bảng.

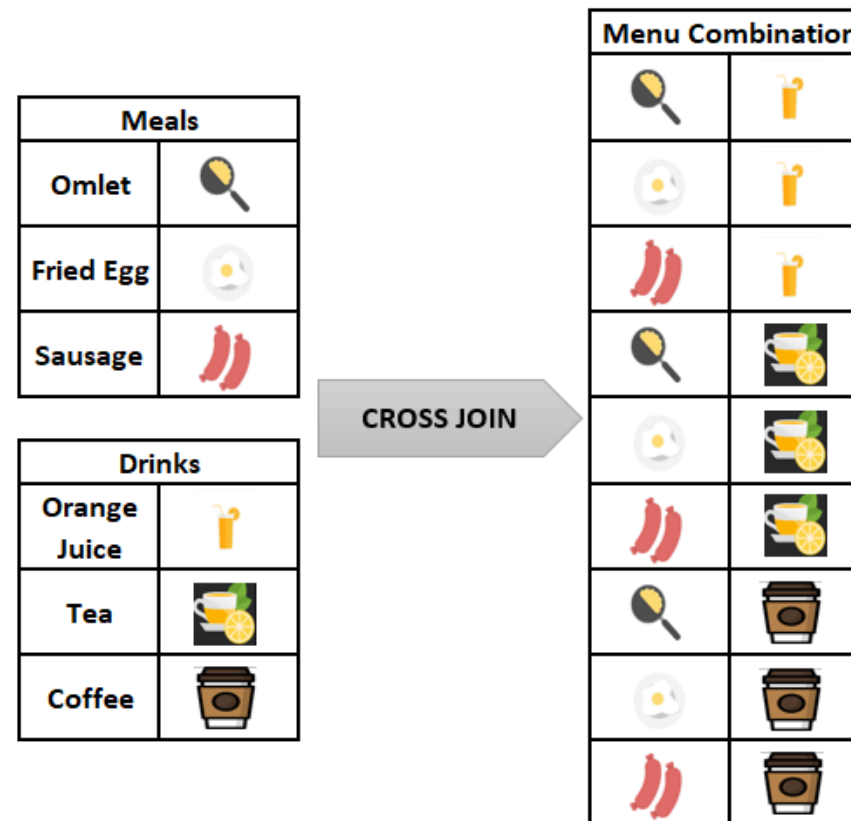
=> Nếu bảng đầu tiên có **n** bản ghi và bảng thứ 2 có **m** bản ghi, **CROSS JOIN** sẽ tạo ra **n*m** bản ghi.

Cấu trúc câu lệnh mẫu:

```
SELECT * FROM [BANG1] CROSS JOIN [BANG2]
```

Hoặc

```
SELECT * FROM [ BANG1] , [BANG2]
```



7

Cách sử dụng và cấu trúc CROSS JOIN

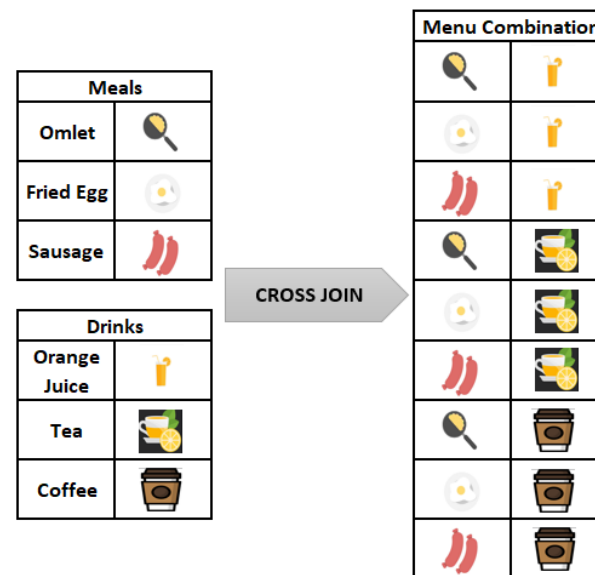
Thực hành:

Sử dụng QUERY sau để tạo bảng Meals và bảng Drinks

```
CREATE TABLE Meals(MealName VARCHAR(100))
CREATE TABLE Drinks(DrinkName VARCHAR(100))
```

```
INSERT INTO Drinks
VALUES('Orange Juice'), ('Tea'), ('Cofee')
INSERT INTO Meals
VALUES('Omlet'), ('Fried Egg'), ('Sausage')
```

```
SELECT *
FROM Meals;
SELECT *
FROM Drinks
```



Yêu cầu 1:

Sử dụng Cross join tạo ra bảng tích hợp gồm 2 cột như hình:

	MealName	DrinkName
1	Omlet	Orange Juice
2	Fried Egg	Orange Juice
3	Sausage	Orange Juice
4	Omlet	Tea
5	Fried Egg	Tea

Yêu cầu 2:

Tạo ra bảng thực đơn bữa sáng MenuList như hình:

	MenuList
1	Omlet-Orange Juice
2	Fried Egg-Orange Juice
3	Sausage-Orange Juice
4	Omlet-Tea
5	Fried Egg-Tea
6	Sausage-Tea
7	Omlet-Coffee
8	Fried Egg-Coffee
9	Sausage-Coffee

TÓM LẠI

Cách sử dụng **CROSS JOIN** hay bất kỳ một loại **JOIN** nào khác là do mong muốn lấy dữ liệu của bạn. Và ta hoàn toàn có thể kết hợp các kỹ thuật khác nhau để có thể lấy được dữ liệu như mong muốn.

Giới thiệu UNION Và UNION ALL

1. Định nghĩa

Union và Union All là phép nối các bảng có **cấu trúc giống nhau**. **Union All** sẽ lấy **tất cả bản ghi** của phép nối còn **Union** sẽ chỉ lấy **các bản ghi duy nhất**

2. Cấu trúc giống nhau là gì?

- Các bảng phải có cùng số cột
- Các cột cũng phải có các loại dữ liệu tương tự
- Các cột trong mỗi bảng phải theo thứ tự

3. Cấu trúc câu lệnh mẫu:

SELECT column1, column2, column3 FROM table1

UNION hoặc **UNION ALL**

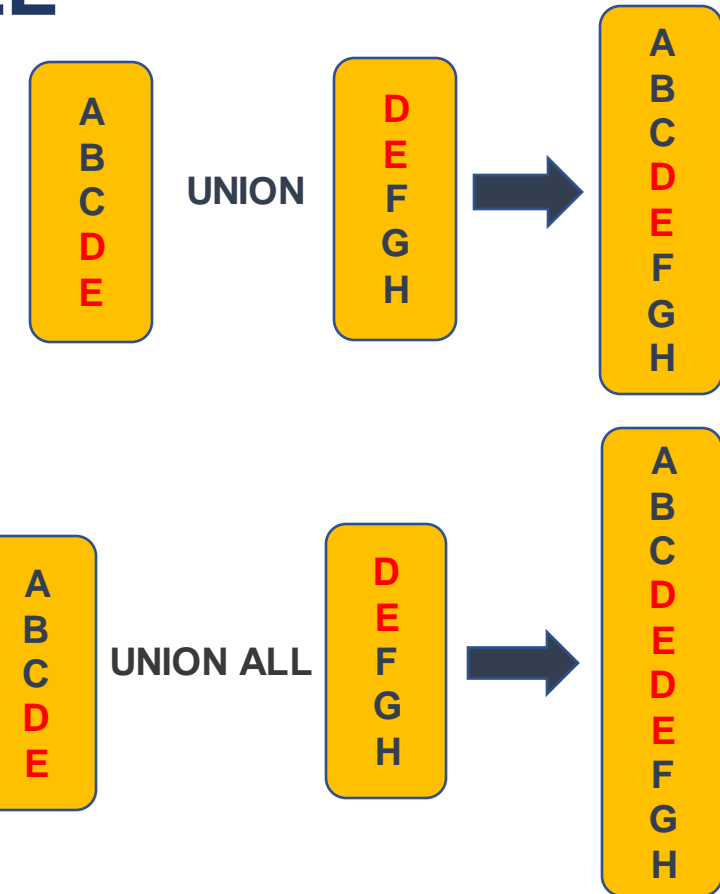
SELECT column1, column2, column3 FROM table2

4. Ví dụ

SELECT t1.province, t1.region FROM Profiles t1

UNION ALL

SELECT t2.province, t2.region FROM Orders t2



Bài tập thực hành:

Dùng Union và Union All nối Profiles với Profiles sau đó rút ra nhận xét so sánh về kết quả số lượng bản ghi từ hai phép nối

8

Bài tập về nhà

BT1.

Từ bảng Orders và Returns tính tổng order_quantity, value, profit của các đơn hàng có trạng thái status = 'Returned' (kết quả chỉ lấy ra đơn hàng có trạng thái returned)

Lưu ý: Làm với 3 loại join (inner join, left join, right join)

Đáp án mẫu: (lấy các trường thông tin như trong bảng kết quả mẫu)

	order_id_return	order_date	total_order_quantity	total_value	total_profit	returned_date
1	45127	2012-12-30 00:00:00.000	10	13.32	7.144	2013-01-13 00:00:00.000
2	42945	2012-12-29 00:00:00.000	45	165.3075	55.305	2013-01-12 00:00:00.000
3	29505	2012-12-27 00:00:00.000	22	37.9456	15.5624	2013-01-11 00:00:00.000
4	30469	2012-12-25 00:00:00.000	46	293.112	137.6924	2013-01-10 00:00:00.000
5	23619	2012-12-21 00:00:00.000	100	4246.1012	1908.7484	2013-01-05 00:00:00.000

BT2.

Sử dụng 2 bảng orders và profiles

Tính total_order_quantity,
total_value, total_profit của từng manager

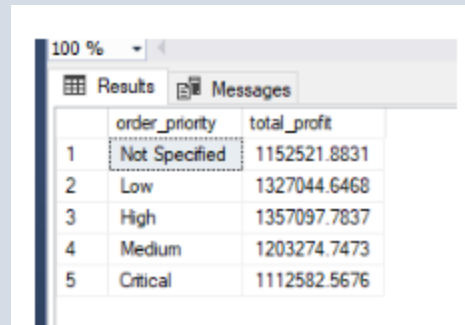
Gợi ý: Mối quan hệ giữa 2 bảng là cột province

	manager	total_order_quantity	total_value	total_profit
1	Erin	51521	3467244.1212	1468952.2242
2	Pat	29311	2266301.2911	955382.580700001
3	Sam	46685	3022359.53560001	1231891.2602
4	Chris	15558	1092511.6531	447856.6088
5	William	71702	4892129.3638	2048438.9546

8

Bài tập về nhà

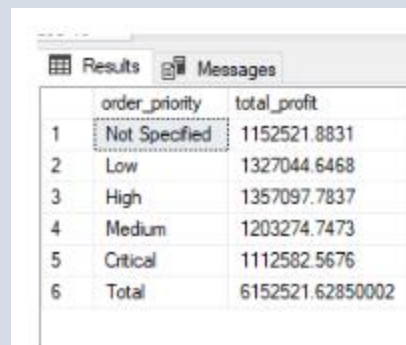
BT4.Hiển thị bảng tính tổng lợi nhuận (total_profit)mỗi mức độ ưu tiên (order_priority) sử dụng union all
Kết quả như ảnh:



	order_priority	total_profit
1	Not Specified	1152521.8831
2	Low	1327044.6468
3	High	1357097.7837
4	Medium	1203274.7473
5	Critical	1112582.5676

BT5.

Hiển thị bảng tính tổng lợi nhuận mỗi mức độ ưu tiên bao gồm dòng total như ảnh:



	order_priority	total_profit
1	Not Specified	1152521.8831
2	Low	1327044.6468
3	High	1357097.7837
4	Medium	1203274.7473
5	Critical	1112582.5676
6	Total	6152521.62850002

8

Hướng dẫn giải bài tập về nhà

BT1.

Dùng inner join

```

Select o.order_id AS order_id_return,
o.order_date,
sum(o.order_quantity) AS total_order_quantity,
sum(o.value) AS total_value,
sum(o.profit) AS total_profit,
r.returned_date
from Orders AS o
inner join Returns AS r
on o.order_id=r.order_id
Group by o.order_id,o.order_date,r.returned_date
Order by o.order_date desc

```

Dùng left join

```

select o.order_id AS order_id_return,
o.order_date,
sum(o.order_quantity) AS total_order_quantity,
sum(o.value) AS total_value,
sum(o.profit) AS total_profit,
r.returned_date
from Returns AS r
left join Orders AS o
on o.order_id=r.order_id
Group by o.order_id,o.order_date,r.returned_date
Order by o.order_date desc

```

Dùng right join

```

select o.order_id AS order_id_return,
o.order_date,
sum(o.order_quantity) AS total_order_quantity,
sum(o.value) AS total_value,
sum(o.profit) AS total_profit,
r.returned_date
from Orders AS o
right join Returns AS r
on o.order_id=r.order_id
Group by o.order_id,o.order_date,r.returned_date
Order by o.order_date desc

```

BT2.

```

select p.manager,
sum(order_quantity) AS total_order_quantity,
sum(value) AS total_value,
sum(profit) AS total_profit
from Orders AS o
left join Profiles AS p
on o.province=p.province
group by p.manager

```



Hướng dẫn giải bài tập về nhà

BT3.

Select order_priority, sum(profit) AS total_profit
From Orders

Where order_priority = 'Not Specified'

Group by order_priority

Union all

Select order_priority, sum(profit) AS total_profit
From Orders

Where order_priority = 'Low'

Group by order_priority

Union all

Select order_priority, sum(profit) AS total_profit
From Orders

Where order_priority = 'High'

Group by order_priority

Union all

Select order_priority, sum(profit) AS total_profit
From Orders

Where order_priority = 'Medium'

Group by order_priority

Union all

Select order_priority, sum(profit) AS total_profit
From Orders

Where order_priority = 'Critical'

Group by order_priority

BT4.

Select order_priority, sum(profit) AS total_profit
From Orders

Group by order_priority

UNION ALL

Select 'Total' AS order_priority, sum(profit)
AS total_profit
From Orders

