

การสืบค้นข้อมูลจากหลาย tables

CUSTOMER		
cid	cname	country
C01	Tom	USA
C02	Mary	CAN
C03	Mike	MEX
C04	Tim	MEX
C05	Steve	USA
C06	Green	CAN
C07	Laura	CAN
C08	Brown	CAN
C09	Bob	USA
C10	Tony	USA

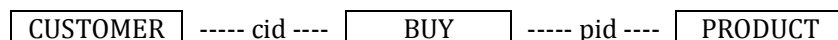
BUY		
cid	pid	quan
C01	P1	100
C01	P1	200
C01	P3	300
C02	P2	100
C03	P1	500
C03	P2	200
C04	P3	400
C05	P3	300
C05	P1	100
C07	P2	200
C08	P3	300
C08	P3	300
C08	P1	200
C09	P2	100
C10	P1	200
C10	P2	300

PRODUCT		
pid	pname	price
P1	Bread	10
P2	Rice	20
P3	Milk	30
P4	Cheese	40

ดังที่ได้กล่าวไว้แล้วว่าการสืบค้นข้อมูลจากหลาย tables เราจะใช้วิธีการ join ตัวอย่างในบทที่แล้วได้ยกตัวอย่างการ join จากสอง tables ในบทนี้จะเป็นการ join มากกว่าสอง โดยมีวัตถุประสงค์

1. สามารถใช้คำสั่งการ join ในกรณีมากกว่าสอง tables
2. เพื่อให้เข้าใจว่าผลลัพธ์ที่เราต้องการต้องนำข้อมูลมาจากรายไหนบ้าง และ
3. สามารถเข้าใจว่า ตารางจากการ join สามารถสั่ง group by อะไรได้บ้าง

การ join มากกว่าสอง tables ใช้หลักการเดียวกันกับสอง tables ดังตัวอย่างจะพบว่าตาราง customer – buy ทำการ join กันด้วย cid และสำหรับ product – buy ทำการ join กันด้วย pid



WHERE customer.cid = buy.cid AND buy.pid = product.pid

ภาพข้างต้นแสดงความสัมพันธ์ระหว่าง tables CUSTOMER, BUY, PRODUCT ระหว่าง CUSTOMER - BUY เชื่อมกันด้วย CID และ BUY – PRODUCT เชื่อมกันด้วย pid นั้นหมายถึงไม่ว่า Query จะต้องการข้อมูลอะไร แต่ WHERE จะเท่ากับ

- WHERE customer.cid = buy.cid กรณี FROM customer, buy
- WHERE buy.pid = product.pid กรณี FROM buy, product
- WHERE customer.cid = buy.cid AND buy.pid = product.pid กรณี FROM customer, buy, product

ความสำคัญของตาราง BUY

หากดูความสัมพันธ์ระหว่างตาราง customer – buy และ product – buy แล้ว ในประเด็นของ referential integrity ตาราง buy จะจัดเป็นตารางรอง ส่วนตาราง customer และ product จะเป็นตารางหลัก หรืออาจเรียก customer และ product ว่าเป็นตาราง master ส่วนตาราง buy เป็น detail แต่หากมองในประเด็นของการเรียกใช้ข้อมูลเกี่ยวกับการขาย เช่นบริษัทขายได้จำนวนกี่ชิ้นเป็นเงินเท่าไร จะพบว่าจะขาด ตาราง buy ไม่ได้เลย เนื่องจากตาราง buy เก็บรายละเอียดการขาย เพราะฉะนั้นทุก query ที่เกี่ยวกับการขายจะต้องใช้ตาราง buy เสมอ

ต้องการ: แสดงรายละเอียดการขาย

```
SELECT *  
FROM buy ;
```

BUY		
cid	pid	quan
C01	P1	100
C01	P1	200
C01	P3	300
C02	P2	100
C03	P1	500
C03	P2	200
C04	P3	400
C05	P3	300
C05	P1	100
C07	P2	200
C08	P3	300
C08	P3	300
C08	P1	200
C09	P2	100
C10	P1	200
C10	P2	300

คำสั่งการจัดกลุ่ม (group by) ที่สามารถใช้กับตาราง buy – หากพิจารณาจากตาราง buy แล้วเราสามารถสั่งจัดกลุ่มด้วย cid และ pid ดังต่อไปนี้

- ต้องการทราบลูกค้าแต่ละคนซื้อสินค้าคนละกี่ชิ้น
SELECT cid, sum(quan)
FROM buy
GROUP BY cid;
- ต้องการทราบว่าสินค้าแต่ละชนิด ขายได้ชิ้นละกี่ชิ้น
SELECT pid, sum(quan)
FROM buy
GROUP BY pid;
- ต้องการทราบว่าลูกค้าแต่ละคนซื้อสินค้าแต่ละชนิด คนละกี่ชิ้น
SELECT cid, pid, sum(quan)
FROM buy
GROUP BY cid, pid;

การ JOIN ระหว่างตาราง BUY - CUSTOMER

ต้องการ: แสดงรายละเอียดการขายพร้อมแสดงชื่อลูกค้า ประเทศของลูกค้า

```
SELECT buy.cid, cname, country, pid, quan
FROM customer, buy
WHERE customer.cid = buy.cid;
```

ประเด็นที่สำคัญ:

- หากต้องการให้ผลลัพธ์แสดงชื่อลูกค้าต้องนำมาจากตาราง customer
- buy กับ customer จะ join กันได้ 16 rows เนื่องจากทุก rows ใน buy สามารถ join กับ customer ได้ (แม้ว่า customer เพียงบาง rows สามารถ join กับ buy ได้ (C06 ใน customer ไม่สามารถ join กับ buy ได้เพราะ C06 ไม่เคยซื้อของ))

BUY – CUSTOMER				
cid	cname	country	pid	quan
C01	Tom	USA	P1	100
C01	Tom	USA	P1	200
C01	Tom	USA	P3	300
C02	Mary	CAN	P2	100
C03	Mike	MEX	P1	500
C03	Mike	MEX	P2	200
C04	Tim	MEX	P3	400
C05	Steve	USA	P3	300
C05	Steve	USA	P1	100
C07	Laura	CAN	P2	200
C08	Brown	CAN	P3	300
C08	Brown	CAN	P3	300
C08	Brown	CAN	P1	200
C09	Bob	USA	P2	100
C10	Tony	USA	P1	200
C10	Tony	USA	P2	300

คำสั่งการจัดกลุ่ม (group by) ที่สามารถใช้กับตาราง buy - customer – หากพิจารณาจากการสั่ง join ระหว่างสอง tables แล้วเราสามารถแสดงชื่อลูกค้า (cname) แทนรหัสลูกค้า (cid) ดังต่อไปนี้

- ต้องการทราบลูกค้าแต่ละคนซื้อสินค้าคนละกี่ชิ้น โดยให้แสดงเป็นชื่อลูกค้า

```
SELECT cname, sum(quan)
FROM customer, buy
WHERE customer.cid = buy.cid
GROUP BY cname;
```
- ต้องการทราบว่าลูกค้าแต่ละคนซื้อสินค้าแต่ละชนิด คนละกี่ชิ้น โดยให้แสดงเป็นชื่อลูกค้า

```
SELECT cname, pid, sum(quan)
FROM customer, buy
WHERE customer.cid = buy.cid
GROUP BY cname, pid;
```

เนื่องจากเราทราบว่าลูกค้าแต่ละคนมาจากประเทศใด เราจึงสามารถสั่ง GROUP BY ประเทศนั้นด้วย

- ต้องการทราบว่า แต่ละประเทศซื้อสินค้าแต่ละชนิด ประเทศละกี่ชิ้น
- ```
SELECT country, pid, sum(quan)
FROM customer, buy
WHERE customer.cid = buy.cid
GROUP BY country, pid;
```

### การ JOIN ระหว่างตาราง BUY - PRODUCT

ต้องการ: แสดงรายละเอียดการขายพร้อมแสดงชื่อสินค้าราคาสินค้า พร้อมทั้งแสดงจำนวน X ราคา

```
SELECT cid, buy.pid, pname, price, quan, price * quan
FROM product, buy
WHERE product.pid = buy.pid;
```

ประเด็นที่สำคัญ:

- หากต้องการชื่อและราคาสินค้าต้องนำมาจากตาราง product
- buy กับ product จะ join กันได้ 16 rows เนื่องจากทุก rows ใน buy สามารถ join กับ product ได้ (แม้ว่า product เพียงบาง rows สามารถ join กับ buy ได้ (P4 ใน product ไม่สามารถ join กับ buy ได้เพราะ p4 ไม่เคยขายได้เลย))

| BUY – PRODUCT |     |       |       |      |            |
|---------------|-----|-------|-------|------|------------|
| cid           | pid | pname | price | quan | price*quan |
| C01           | P1  | Bread | 10    | 100  | 1000       |
| C01           | P1  | Bread | 10    | 200  | 2000       |
| C01           | P3  | Milk  | 30    | 300  | 9000       |
| C02           | P2  | Rice  | 20    | 100  | 2000       |
| C03           | P1  | Bread | 10    | 500  | 5000       |
| C03           | P2  | Rice  | 20    | 200  | 4000       |
| C04           | P3  | Milk  | 20    | 400  | 8000       |
| C05           | P3  | Milk  | 20    | 300  | 6000       |
| C05           | P1  | Bread | 10    | 100  | 1000       |
| C07           | P2  | Rice  | 20    | 200  | 4000       |
| C08           | P3  | Milk  | 20    | 300  | 6000       |
| C08           | P3  | Milk  | 20    | 300  | 6000       |
| C08           | P1  | Bread | 10    | 200  | 2000       |
| C09           | P2  | Rice  | 20    | 100  | 2000       |
| C10           | P1  | Bread | 10    | 200  | 2000       |
| C10           | P2  | Rice  | 20    | 300  | 6000       |

คำสั่งการจัดกลุ่ม (group by) ที่สามารถใช้กับตาราง buy - product – หากพิจารณาจากการสั่ง join กัน เราสามารถแสดงชื่อสินค้า (pname) แทนรหัสสินค้า (pid) ได้ดังนี้

- ต้องการทราบสินค้าแต่ละรายการ ขายได้กี่ชิ้น เป็นเงินเท่าไร โดยให้แสดงเป็นชื่อสินค้า

```
SELECT pname, sum(quan), sum(price*quan)
FROM product, buy
WHERE product.pid = buy.pid
GROUP BY pname;
```

- ต้องการทราบลูกค้าแต่ละราย มียอดซื้อคนละเท่าไร

```
SELECT cid, sum(price*quan)
FROM product, buy
WHERE product.pid = buy.pid
GROUP BY cid;
```

ไม่สามารถแสดงเป็น cname ได้เนื่องจากไม่ได้เรียกตาราง customer มา

### การ JOIN ระหว่างตาราง CUSTOMER - BUY - PRODUCT

**ต้องการ:** แสดงรายละเอียดการขายพร้อมแสดงชื่อลูกค้า ประเทศ ชื่อสินค้าราคาสินค้า พร้อมทั้งแสดงจำนวน X ราคา

```
SELECT customer.cid, cname, country, buy.pid, pname, price, quan, price * quan
FROM product, buy
WHERE product.pid = buy.pid;
```

ประเด็นที่สำคัญ:

- buy, customer, product สามารถ join กันได้ 16 rows เนื่องจากทุก rows ใน buy สามารถ join กับ customer, product ได้

| BUY – PRODUCT |       |         |     |       |       |      |            |
|---------------|-------|---------|-----|-------|-------|------|------------|
| cid           | cname | country | pid | pname | price | quan | price*quan |
| C01           | Tom   | USA     | P1  | Bread | 10    | 100  | 1000       |
| C01           | Tom   | USA     | P1  | Bread | 10    | 200  | 2000       |
| C01           | Tom   | USA     | P3  | Milk  | 30    | 300  | 9000       |
| C02           | Mary  | CAN     | P2  | Rice  | 20    | 100  | 2000       |
| C03           | Mike  | MEX     | P1  | Bread | 10    | 500  | 5000       |
| C03           | Mike  | MEX     | P2  | Rice  | 20    | 200  | 4000       |
| C04           | Tim   | MEX     | P3  | Milk  | 20    | 400  | 8000       |
| C05           | Steve | USA     | P3  | Milk  | 20    | 300  | 6000       |
| C05           | Steve | USA     | P1  | Bread | 10    | 100  | 1000       |
| C07           | Laura | CAN     | P2  | Rice  | 20    | 200  | 4000       |
| C08           | Brown | CAN     | P3  | Milk  | 20    | 300  | 6000       |
| C08           | Brown | CAN     | P3  | Milk  | 20    | 300  | 6000       |
| C08           | Brown | CAN     | P1  | Bread | 10    | 200  | 2000       |
| C09           | Bob   | USA     | P2  | Rice  | 20    | 100  | 2000       |
| C10           | Tony  | USA     | P1  | Bread | 10    | 200  | 2000       |
| C10           | Tony  | USA     | P2  | Rice  | 20    | 300  | 6000       |

คำสั่งการจัดกลุ่ม (group by) ที่สามารถใช้กับตาราง buy – customer - product จะสามารถสั่งได้ดังนี้

- ต้องการทราบลูกค้าแต่ละราย มียอดซื้อ เป็นเงินเท่าไร โดยให้แสดงเป็นชื่อลูกค้า

```
SELECT cname, sum(price*quan)
FROM buy, customer, product
WHERE buy.cid = product.cid AND
 product.pid = buy.pid
GROUP BY cname;
```

- ต้องการทราบแต่ละประเทศ มียอดซื้อ เป็นเงินเท่าไร

```
SELECT country, sum(price*quan)
FROM buy, customer, product
WHERE buy.cid = product.cid AND
 product.pid = buy.pid
GROUP BY country;
```

- ต้องการทราบลูกค้าแต่ละราย ซื้อสินค้าอะไรบ้าง จำนวนกี่ชิ้น เป็นเงินเท่าไร โดยให้แสดงเป็นชื่อลูกค้า

```
SELECT cname, pname, sum(quan), sum(price*quan)
FROM buy, customer, product
WHERE buy.cid = product.cid AND
 product.pid = buy.pid
GROUP BY cname, pname;
```

- ต้องการทราบว่าแต่ละประเทศ ซื้อสินค้าอะไรบ้าง จำนวนกี่ชิ้น เป็นเงินเท่าไร

```
SELECT country, pname, sum(quan), sum(price*quan)
FROM buy, customer, product
WHERE buy.cid = product.cid AND
 product.pid = buy.pid
GROUP BY country, pname;
```

ข้อเสนอแนะ นักศึกษาควรศึกษาตัวอย่างที่อาจารย์ใช้ 4 และ 7 ตารางมาศึกษาเพิ่มเติม โดยแนวคิดในการสืบค้นข้อมูลจะเป็นแนวคิดเดียวกับตัวอย่าง 3 ตารางในเอกสารฉบับนี้

แบบฝึกหัด

แบบฝึกหัดดังกล่าวมิได้มีเฉพาะการ join แต่รวมถึงคำสั่งต่างๆ ที่สามารถกระทำกับข้อมูลปรากฏในหลายตาราง

1. แสดง ชื่อลูกค้า, ชื่อสินค้า, ราคา, จำนวน, เป็นเงิน (ราคา X จำนวน) ของทุกรายการซื้อ
2. แสดง ชื่อลูกค้า, ชื่อสินค้า, ราคา, จำนวน, เป็นเงิน (ราคา X จำนวน) เฉพาะรายการ Bread
3. แสดง ชื่อลูกค้า, ชื่อสินค้า, ราคา, จำนวน, เป็นเงิน (ราคา X จำนวน) เฉพาะรายการที่เป็นเงินสูงกว่า 2,000 เรียงลำดับผลลัพธ์ตามเป็นเงินจากมากไปน้อย
4. แสดงยอดขายทั้งสิ้นของบริษัท
5. แสดงชื่อลูกค้าที่ไม่มีการซื้อของเลย
6. แสดงชื่อสินค้าที่ไม่มีการขายเลย
7. ให้หาว่าสินค้าแต่ละชนิดมียอดขายเป็นเงินเท่าไร
8. ให้หาว่าลูกค้าแต่ละรายมียอดซื้อเป็นเงินเท่าไร
9. ให้หาว่าลูกค้าแต่ละประเทศมียอดสั่งซื้อเป็นเงินเท่าไร
10. ให้หาว่าลูกค้าแต่ละประเทศ สั่งซื้อสินค้าแต่ละชนิด เป็นเงินเท่าไร
11. ให้แสดง สินค้า, ยอดขาย ของสินค้าที่มียอดขายสูงสุด\*
12. ให้แสดง ลูกค้า, ยอดซื้อ ของลูกค้าที่มียอดซื้อสูงสุด\*
13. นักศึกษาคิดว่าเป็นไปได้ไหมที่จะมีคำสั่ง SQL ใดที่จะเรียกใช้ตาราง customer, product (FROM customer, product)

\* โจทย์ Advance

เฉลย

1

```
SELECT pname, cname, price, quan, (price*quan)
FROM product, customer, buy
WHERE product.pid = buy.pid AND customer.cid = buy.cid;
```

2

```
SELECT pname, cname, price, quan, (price*quan)
FROM product, customer, buy
WHERE product.pid = buy.pid AND customer.cid = buy.cid
 AND pname = 'Bread';
```

3

```
SELECT cname, pname, price, quan, price*quan
FROM customer, product, buy
WHERE customer.cid = buy.cid AND buy.pid = product.pid
 AND price*quan > 2000
ORDER BY price*quan DESC;
```

4

```
SELECT SUM(price*quan)
FROM product, buy
WHERE product.pid = buy.pid;
```

5

```
SELECT *
FROM customer
WHERE cid NOT IN
 (SELECT cid
 FROM buy);
```

6

```
SELECT *
FROM product
WHERE pid NOT IN
 (SELECT pid
 FROM buy);
```



7

```
SELECT pname, sum(price*quan)
FROM product, buy
WHERE product.pid = buy.pid
GROUP BY pname;
```

ทำ outer join เพื่อให้สินค้าทุกชนิดแสดงผลลัพธ์ (สินค้าที่ขายไม่ได้แสดงผลลัพธ์)

```
SELECT pname, sum(price*quan)
FROM product left join buy on product.pid = buy.pid
GROUP BY pname;
```

8

```
SELECT cname, SUM(price*quan)
FROM product, buy, customer
WHERE product.pid = buy.pid AND customer.cid = buy.cid
GROUP BY cname;
```

ทำ outer join เพื่อให้ลูกค้าทุกคนแสดงผลลัพธ์ (ลูกค้าที่ไม่ได้ซื้อแสดงผลลัพธ์)

```
SELECT cname, SUM(price*quan)
FROM customer LEFT JOIN (buy JOIN product ON buy.pid = product.pid) ON customer.cid =
buy.cid
GROUP BY cname;
```

9

```
SELECT nationality, SUM(price*quan)
FROM product, customer, buy
WHERE product.pid = buy.pid AND customer.cid = buy.cid
GROUP BY nationality;
```

10

```
SELECT nationality, pname, SUM(price*quan)
FROM product, customer, buy
WHERE product.pid = buy.pid AND customer.cid = buy.cid
GROUP BY nationality, pname
ORDER BY nationality;
```

11

```
SELECT pname, SUM(price*quan)
FROM product, buy
WHERE product.pid = buy.pid
GROUP BY pname
HAVING SUM(price*quan) = (SELECT MAX(SUM(price*quan))
 FROM product, buy
 WHERE product.pid = buy.pid
 GROUP BY pname);
```

12

```
SELECT cname, sum(price*quan)
FROM product, buy, customer
WHERE product.pid = buy.pid AND buy.cid = customer.cid
GROUP BY cname
HAVING SUM(price*quan) = (SELECT MAX(SUM(price*quan))
 FROM product, buy
 WHERE product.pid = buy.pid
 GROUP BY cid);
```

หรือ

```
select top 1 cname, sumx
from
(
select cname, sum(price*quan) as sumx
from customer, buy, product
where customer.cid = buy.cid and buy.pid = product.pid
group by cname
order by sum(price*quan) desc
)
;
```

13

เป็นไปได้จะมีคำสั่ง SQL ใดๆ ที่จะใช้ FROM customer, product เพราะ tables ทั้งสองไม่มีความสัมพันธ์กันโดยตรง หากจะเรียกใช้จะต้อง FROM customer, buy, product เพราะ customer จะมีความสัมพันธ์กับ product โดยผ่าน buy เท่านั้น