

การ SELECT และ UPDATE ข้อมูลจากหลาย tables

CUSTOMERS

CID	CNAME	CITY	CREDIT	C_AMT
1	Peter	Pais	20000	0
2	Rob	Romes	30000	0

ORDERS

OID	ODATE	CID	O_AMT
101	27-Jun_12	1	0
102	27-Jun-12	2	0
103	28-Jun-12	1	0

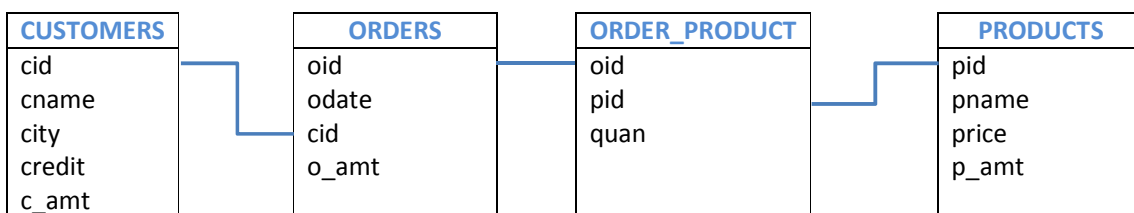
PRODUCTS

PID	PNAME	PRICE	P_AMT
1001	Pen	39	0
1002	Pencil	2	0

ORDER_PRODUCT

OID	PID	QUAN
101	1001	5
101	1002	10
102	1001	15
103	1001	20
103	1002	25

RELATIONSHIP



A หากเราใช้คำสั่งที่ปรากฏด้านขวา จะได้
ผลลัพธ์ดังด้านล่าง

```
SELECT *  
FROM order_product ;
```

OID	PID	QUAN
101	1001	5
101	1002	10
102	1001	15
103	1001	20
103	1002	25

B หากต้องการให้ผลลัพธ์แสดงชื่อสินค้า
ราคา และ ราคา*จำนวน เราต้องนำข้อมูล
ดังกล่าวมาจาก table PRODUCTS

```
SELECT oid, products.pid, quan, pname, price,  
       price*quan  
FROM products, order_product  
WHERE products.pid = order_product.pid ;
```

OID	PID	QUAN	PNAME	PRICE	PRICE*QUAN
101	1001	5	Pen	39	195
101	1002	10	Pencil	2	20
102	1001	15	Pen	39	585
103	1001	20	Pen	39	780
103	1002	25	Pencil	2	50

C ผลลัพธ์จากข้อ (B) หากเราต้องการให้
สรุปว่าสินค้าแต่ละรายการมียอดสั่งซื้อ
เท่าใด โดยให้แสดงผลเป็นชื่อสินค้า
ให้สังเกตว่าโจทย์ข้อนี้ยังคงต้องใช้ข้อมูล
จากทั้งสอง tables

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

PNAME	SUM(PRICE*QUAN)
Pen	1560
Pencil	70

D ผลลัพธ์จากข้อ (B) หากเราต้องการให้
แสดงวันที่ของ order เราจะต้องนำข้อมูล
วันที่มาจาก table ORDERS

```
SELECT orders.oid, odate, products.pid, quan,
       pname, price, price*quan
FROM   products, order_product, orders
WHERE  products.pid = order_product.pid
AND
       orders.oid = order_product.oid ;
```

OID	ODATE	PID	QUAN	PNAME	PRICE	PRICE*QUAN
101	27-Jun-12	1001	5	Pen	39	195
101	27-Jun-12	1002	10	Pencil	2	20
102	27-Jun-12	1001	15	Pen	39	585
103	28-Jun-12	1001	20	Pen	39	780
103	28-Jun-12	1002	25	Pencil	2	50

E จากการ join 3 tables ทำให้เราสามารถ
สรุปยอดขายในแต่ละวันได้

```
SELECT odate, sum( price*quan)
FROM   products, order_product, orders
WHERE  products.pid = order_product.pid
AND
       orders.oid = order_product.oid
GROUP BY odate ;
```

ODATE	PRICE
27-Jun-12	800
28-Jun-12	830

F ผลลัพธ์จากข้อ (D) หากเราต้องการให้
แสดงชื่อลูกค้าเพิ่มเติม เราจะต้องนำข้อมูล
ชื่อลูกค้ามาจาก table CUSTOMERS

```
SELECT cname, orders.oid, odate, products.pid,
       quan, pname, price, price*quan
FROM   products, order_product, orders,
       customers
WHERE  products.pid = order_product.pid
AND
       orders.oid = order_product.oid
AND
       orders.cid = customers.cid ;
```

CNAME	OID	ODATE	PID	QUAN	PNAME	PRICE	PRICE*QUAN
Peter	101	27-Jun-12	1001	5	Pen	39	195
Peter	101	27-Jun-12	1002	10	Pencil	2	20
Rob	102	27-Jun-12	1001	15	Pen	39	585
Peter	103	28-Jun-12	1001	20	Pen	39	780
Peter	103	28-Jun-12	1002	25	Pencil	2	50

G จากการใช้ join 4 tables ทำให้เราสามารถสรุปยอดซื้อของลูกค้าแต่ละคนได้

```
SELECT cname, sum( price*quan)
FROM products, order_product, orders,
      customers
WHERE products.pid = order_product.pid
      AND
      orders.oid = order_product.oid
      AND
      orders.cid = customers.cid
GROUP BY cname ;
```

CNAME	SUM(PRICE*QUAN)
Peter	1045
Rob	585

H จากการใช้ join 4 tables โดยมี table CUSTOMERS เข้าร่วม ทำให้ยังสามารถสรุปยอดซื้อของลูกค้าแต่ละเมืองได้

```
SELECT city, sum( price*quan)
FROM products, order_product, orders,
      customers
WHERE products.pid = order_product.pid
      AND
      orders.oid = order_product.oid
      AND
      orders.cid = customers.cid
GROUP BY city;
```

CITY	SUM(PRICE*QUAN)
Rome	1045
Paris	585

สรุปประเด็นสำคัญในการทำโจทย์การ JOIN

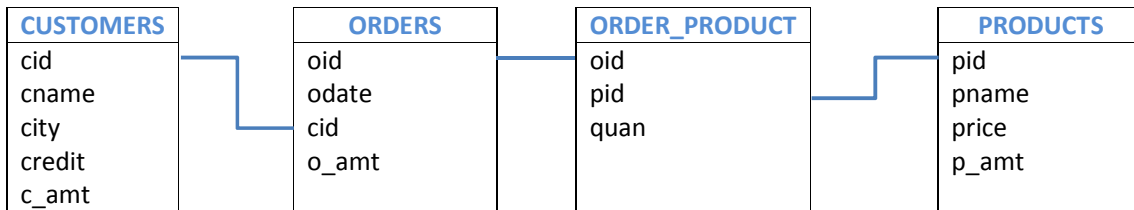


Diagram ที่แสดงความสัมพันธ์ระหว่าง tables ดังกล่าวจะช่วยให้การเขียนคำสั่ง join สองประเด็นด้วยกัน คือ

1. ทำให้เราทราบว่า การสั่ง join ตรง where clause จะต้องสั่งอย่างไร เช่น การ join ระหว่าง order_product และ products จะต้องใช้ pid

```

SELECT *
FROM      products, order_product
WHERE     products.pid = order_product.pid
  
```

2. ทำให้ทราบในการ join ใช้ tables ใหนได้บ้าง จาก diagram ดังกล่าว สรุปได้ดังนี้

customers	ต้อง join กับ	orders
orders	ต้อง join กับ	order_product
order_product	ต้อง join กับ	products

ไม่มีทางที่ orders จะ join กับ products โดยตรง หาก order จะ join กับ products ต้องผ่าน order_product ดังตัวอย่าง D. เช่นเดียวกับ customers ไม่มีทางจะ join กับ products ได้โดยตรงหากต้องการ join ต้อง join ผ่าน orders และ order_product ดังตัวอย่าง F.

การ UPDATE ข้อมูลจากหลาย tables

A ต้องการหายยอดซื้อของแต่ละ orders

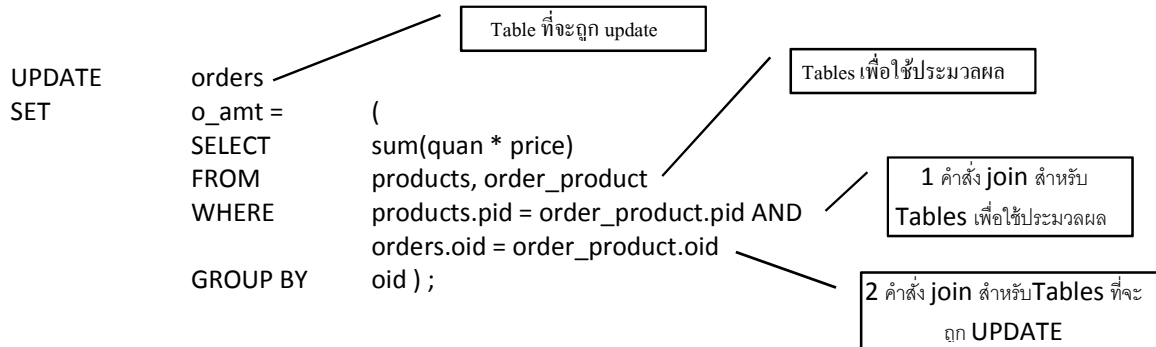
Tables ที่ต้องใช้ทำโจทย์ข้อนี้	orders, products, order_product
Tables ที่จะถูก update	orders (o_amt)
Tables เพื่อใช้ในการประมวลผล	products, order_product

```
UPDATE orders
SET o_amt = (
  SELECT sum(quan * price)
  FROM products, order_product
  WHERE products.pid = order_product.pid AND
        orders.oid = order_product.oid
  GROUP BY oid );
```

คำอธิบายการเขียนคำสั่งการ JOIN ตรง WHERE clause

การ join ตรง WHERE clause ต้อง แบ่งเป็นสองส่วน และ ต้อง มีการ join ของทั้งสองส่วน คือ

1. Join ระหว่าง tables เพื่อใช้ประมวลผล จากตัวอย่าง tables เพื่อใช้ประมวลผลคือ products และ order_product ใช้เพียงสอง tables เราก็จะทราบผลลัพธ์แล้วว่าแต่ละ order มียอดสั่งซื้อเท่าไร (คำอธิบายที่กล่าวมาก็เหมือนกับการใช้คำสั่ง select ทั่วไป ไม่ได้มีอะไรแตกต่าง)
2. Join ระหว่าง tables ที่จะถูก update ค่า (orders) กับ tables ที่ทำการประมวลผลจาก 1. ซึ่ง orders จะ join กับ order_product ด้วย oid



B ต้องการหายอดขายของสินค้าแต่ละชนิด

Tables ที่ต้องใช้ทำโจทย์ข้อนี้	products, order_product
Tables ที่จะถูก update	products (p_amt)
Tables เพื่อใช้ในการประมวลผล	products, order_product

```
UPDATE products B
SET p_amt = (
SELECT sum(quan * price)
FROM products, order_product
WHERE products.pid = order_product.pid AND
B.pid = products.pid
GROUP BY products.pid );
```

โจทย์ข้อนี้จะต่างกับข้อ A คือ table ที่ประมวลผลกับ table ที่ถูก update เป็น table เดียวกันคือ products ดังที่ได้กล่าวไว้แล้วว่าการเขียนคำสั่งเพื่อใช้ในการ join ในส่วน WHERE clause ต้อง มีสองส่วน คือ ระหว่าง tables ที่ใช้ในการประมวลผล และ tables ที่ถูก update

การ join ระหว่าง tables ส่วนประมวลผล (products) กับ table ส่วน update (products) ซึ่งเป็น table เดียวกัน ซึ่งเป็นไปไม่ได้ที่เราจะเขียนคำสั่ง products.pid = products.pid

ด้วยเหตุผลดังกล่าวข้างต้น เราจึงต้องทำให้ table products ณ จุดใดจุดหนึ่งให้เสมือนเป็น table อื่นที่ไม่ใช่ table products โดยการตั้งชื่อเล่น (Alias) ในตัวอย่างข้างต้น เราให้ table products ณ จุดที่จะโดย update โดยตั้งชื่อเล่นว่า B ด้วยคำสั่ง UPDATE products B

C ต้องการหายอดขายของลูกค้าแต่ละราย

Tables ที่ต้องใช้ทำโจทย์ข้อนี้	customers, orders, products, order_product
Tables ที่จะถูก update	customers (c_amt)
Tables เพื่อใช้ในการประมวลผล	orders, products, order_product

```
UPDATE customers
SET c_amt = (
    SELECT sum(quan*price)
    FROM orders, order_product, products
    WHERE orders.oid = order_product.oid and
    order_product.pid = products.pid and
    customers.cid = orders.cid
    GROUP BY cid) ;
```

เราอาจใช้คำสั่ง MERGE ของ Oracle แทนคำสั่ง UPDATE ได้ดังนี้

```
MERGE INTO customers
USING(
    SELECT cid, sum(quan*price) as amtx
    FROM orders, order_product, products
    WHERE orders.oid = order_product.oid and
    products.pid = order_product.pid
    GROUP BY cid
) c
ON (customers.cid = c.cid)
WHEN matched then
UPDATE SET amt = amt + c.amtx;
```