1	การเรียกดูข้อมูลขั้นพื้นฐาน	ปฏิบัติการ

หลังจากที่ได้ศึกษาการใช้งาน MySQL และการจัดการข้อมูลเบื้องต้นไปแล้ว ในครั้งนี้จะทำการเรียนรู้วิธีเรียกดูข้อมูลที่อยู่ใน database ซึ่งได้ใช้ฐานข้อมูลจาก http://www.mysqltutorial.org/mysql-sample-database.aspx รายละเอียดความสัมพันธ์ของแต่ละตารางแนบไว้ส่วนท้ายของเนื้อหา

MySQL: เข้าไปที่เว็บไซต์ http://thepsatri.com/phpMyAdmin/

Username: lab_sql

Password: Lab@sql310

1.1 คำสั่ง SELECT แบบพื้นฐาน

รูปแบบของคำสั่งนี้จะมีส่วนสำคัญพื้นฐาน 2 ส่วนคือ SELECT และ FROM โดยส่วน SELECT จะเป็นการเลือกคอลัมน์ที่ต้องการจากตารางที่ระบุในส่วน FROM มีรูปแบบคำสั่งดังนี้

```
SELECT column_name
FROM table_name;
```

ซึ่งถ้าต้องการมากกว่า 1 คอลัมน์จะต้องใช้เครื่องหมาย “,” คั่น หรือถ้าต้องการทุกคอลัมน์ในตารางให้ใส่เป็นเครื่องหมาย “*”

```
SELECT customerNumber,customerName,city
FROM customers;
```

```
+----------------+-----------------------------------+-------------------+
| customerNumber | customerName                      | city              |
+----------------+-----------------------------------+-------------------+
|            103 | Atelier graphique                 | Nantes            |
|            112 | Signal Gift Stores                | Las Vegas         |
|            114 | Australian Collectors, Co.        | Melbourne         |
|            119 | La Rochelle Gifts                 | Nantes            |
|            121 | Baane Mini Imports                | Stavern           |
|            124 | Mini Gifts Distributors Ltd.      | San Rafael        |
|            125 | Havel & Zbyszek Co                | Warszawa          |
|            128 | Blauer See Auto, Co.              | Frankfurt         |
|            129 | Mini Wheels Co.                   | San Francisco     |
|            …   |            …                      |                 … |
|            …   |            …                      |                 … |
|            489 | Double Decker Gift Stores, Ltd    | London            |
|            495 | Diecast Collectables              | Boston            |
|            496 | Kelly's Gift Shop                 | AucklAND          |
+----------------+-----------------------------------+-------------------+
122 rows in set (0.04 sec)
```

1.2 การจัดรูปแบบข้อมูลที่แสดง

จากรูปแบบพื้นฐานที่ใช้คำสั่ง SELECT ยังมีส่วนคำสั่งประกอบเพิ่มเติมเพื่อให้ข้อมูลมีรูปแบบที่เข้าใจง่ายขึ้น ด้วยคำสั่งเหล่านี้

1.2.1 เปลี่ยนชื่อคอลัมน์

ในคำสั่งที่ผ่านมาจะนำชื่อคอลัมน์มาจากฐานข้อมูล บางครั้งผู้อ่านข้อมูลอาจจะเข้าใจคลาดเคลื่อน MySQL จึงมีคำสั่งเพิ่มในการเปลี่ยนชื่อคอลัมน์ มีรูปแบบดังนี้

```
SELECT column_name as “new column name”
FROM table_name;
```

ให้สังเกตว่าผลลัพธ์ที่ได้จะมีการเปลี่ยนชื่อคอลัมน์ที่แสดงผล แต่การอ้างอิงยังใช้ชื่อคอลัมน์เดิมตามที่มีในฐานข้อมูล ดังตัวอย่างต่อไปนี้

```
SELECT customerName as "Customer 's Name",city as "City"
FROM customers;

+-----------------------------------+--------------------+
| Customer 's Name                  | City               |
+-----------------------------------+--------------------+
| Atelier graphique                 | Nantes             |
| Signal Gift Stores                | Las Vegas          |
| Australian Collectors, Co.        | Melbourne          |
| La Rochelle Gifts                 | Nantes             |
| Baane Mini Imports                | Stavern            |
            …                             …
            …                             …
| Diecast Collectables              | Boston             |
| Kelly's Gift Shop                 | AucklAND           |
+-----------------------------------+--------------------+
122 rows in set (0.01 sec)
```

1.2.2 การจำกัดจำนวนข้อมูลที่นำมาแสดง

จากตัวอย่างที่ผ่านมาจะพบว่า หากมีการเรียกดูข้อมูลจะได้ข้อมูลทั้งหมดที่มีอยู่ในฐานข้อมูล ซึ่งบางครั้งอาจจะเยอะเกินไปทำให้การวิเคราะห์ข้อมูลทำได้ยาก ดังนั้น MySQL จึงได้มีการจำกัดจำนวนที่แสดงในแต่ละครั้ง ด้วยคำสั่งดังนี้

```
SELECT column_name
FROM table_name
[LIMIT Number;]
[LIMIT Begin, Number;]
```

โดยที่ Number คือจำนวนเรคคอร์ดที่จะนำมาแสดง และ Begin คือเรคคอร์ดแรกที่ต้องการนำมาแสดง ซึ่งจะเริ่มนับเรคคอร์ดแรกที่ 0 ให้สังเกตผลลัพธ์ที่ได้จากตัวอย่างต่อไปนี้

```
SELECT customerName as "Customer 's Name",city as "City"
FROM customers
LIMIT 3;
```

```
+---------------------------+-----------+
| Customer 's Name          | City      |
+---------------------------+-----------+
| Atelier graphique         | Nantes    |
| Signal Gift Stores        | Las Vegas |
| Australian Collectors, Co. | Melbourne |
+---------------------------+-----------+
3 rows in set (0.03 sec)
```

```
SELECT customerName as "Customer 's Name",city as "City"
FROM customers
LIMIT 1,2;
```

```
+---------------------------+-----------+
| Customer 's Name          | City      |
+---------------------------+-----------+
| Signal Gift Stores        | Las Vegas |
| Australian Collectors, Co. | Melbourne |
+---------------------------+-----------+
2 rows in set (0.00 sec)
```

1.2.3 การเรียงลำดับข้อมูล

ผู้ใช้สามารถเรียงลำดับข้อมูลตามข้อมูลคอลัมน์หนึ่งๆ ได้ โดยเพิ่มคำสั่ง order by ไปท้ายคำสั่ง SELECT ซึ่งถ้าไม่ได้กำหนดข้อมูลเพิ่มจะเป็นการเรียงตามลำดับน้อยไปมาก แต่ถ้าต้องการเรียงจากมากไปน้อยจะต้องกำหนด desc ไปที่ท้ายคำสั่งด้วย รูปแบบของคำสั่งมีดังนี้

```
SELECT column_name
FROM table_name
order by column_name [desc];
```

ตัวอย่างเช่น

```
SELECT customerName as "Customer 's Name",city as "City"
FROM customers
order by city
LIMIT 5;
```

```
+---------------------------+------------+
| Customer 's Name          | City       |
+---------------------------+------------+
| Warburg Exchange          | Aachen     |
| Diecast Classics Inc.     | Allentown  |
| Schuyler Imports          | Amsterdam  |
| Kelly's Gift Shop         | AucklAND   |
| Down Under Souveniers, Inc | AucklAND   |
+---------------------------+------------+
5 rows in set (0.00 sec)
```

```
SELECT customerName as "Customer 's Name",city as "City"
FROM customers
order by city desc
LIMIT 5;

+------------------------------+--------------+
| Customer 's Name             | City         |
+------------------------------+--------------+
| Heintze Collectables         | ลrhus        |
| Mini Classics                | White Plains |
| Extreme Desk Decorations, Ltd | Wellington   |
| Havel & Zbyszek Co           | Warszawa     |
| Auto Assocs & Cie.           | Versailles   |
+------------------------------+--------------+
5 rows in set (0.00 sec)
```

1.2.4 การแสดงข้อมูลที่ไม่ซ้ำกัน

ให้นักศึกษาลองเรียกใช้คำสั่งต่อไปนี้

```
SELECT city FROM customers order by city;
```

จะพบว่าข้อมูลที่แสดงมีการซ้ำกัน ทำให้ไม่สามารถนับจำนวนที่แท้จริงได้ว่ามี city อยู่จำนวนเท่าไร ซึ่ง MySQL ได้เตรียมคำสั่งในการแสดงข้อมูลที่ไม่ซ้ำกัน ดังนี้

```
SELECT distinct column_name
FROM table_name
```

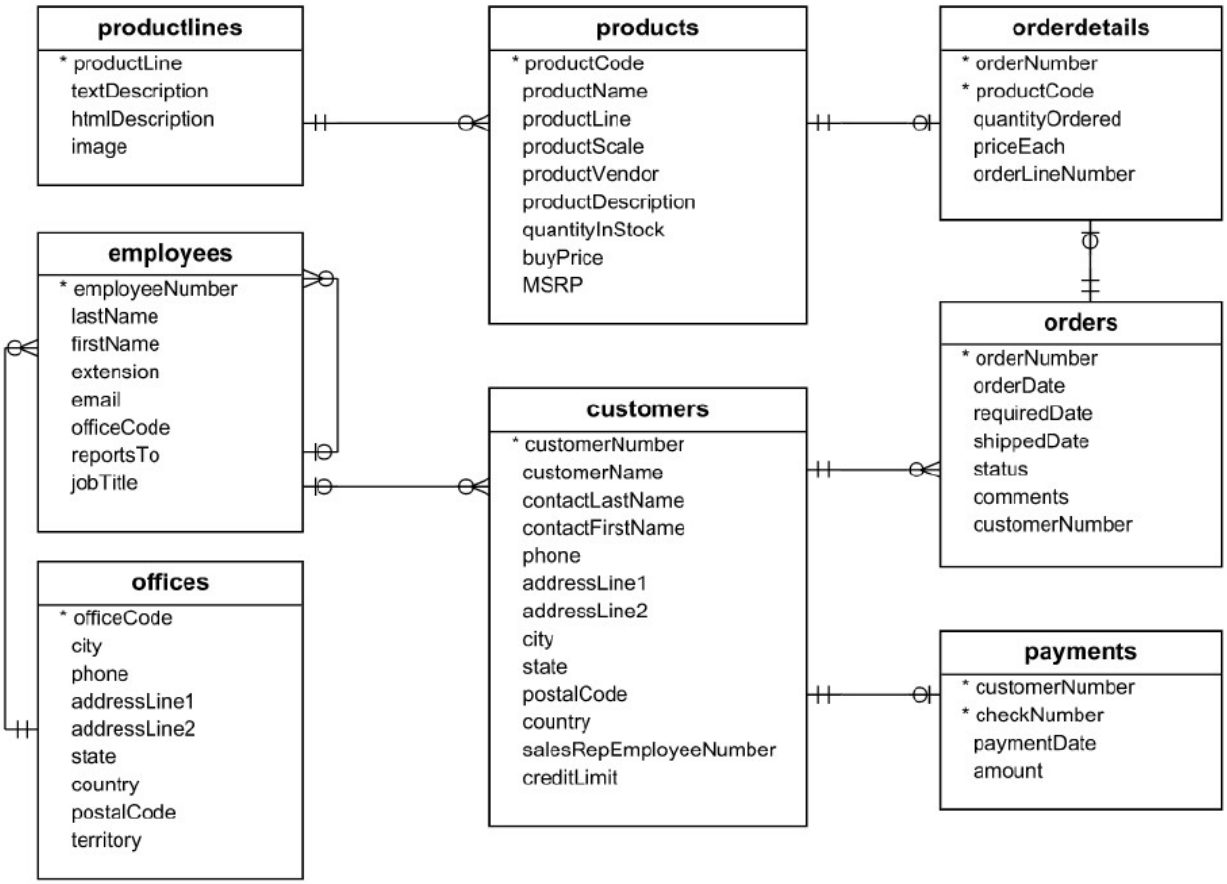
ตัวอย่างเช่น

```
SELECT city FROM customers order by city;

+--------------------+
| city               |
+--------------------+
| Aachen             |
| Allentown          |
| Amsterdam          |
| AucklAND           |
| AucklAND           |
| AucklAND           |
| Barcelona          |
| Bergamo            |
      …
| Wellington         |
| White Plains       |
| ลrhus              |
+--------------------+
122 rows in set (0.01 sec)
```

```
SELECT distinct city FROM customers order by city;

+--------------------+
| city               |
+--------------------+
| Aachen             |
| Allentown          |
| Amsterdam          |
| AucklAND           |
| Barcelona          |
| Bergamo            |
      …
| Wellington         |
| White Plains       |
| ลrhus              |
+--------------------+
95 rows in set (0.07 sec)
```


productlines
* productLine
textDescription
htmlDescription
image
products
* productCode
productName
productLine
productScale
productVendor
productDescription
quantityInStock
buyPrice
MSRP
orderdetails
* orderNumber
* productCode
quantityOrdered
priceEach
orderLineNumber
employees
* employeeNumber
lastName
firstName
extension
email
officeCode
reportsTo
jobTitle
orders
* orderNumber
orderDate
requiredDate
shippedDate
status
comments
customerNumber
customers
* customerNumber
customerName
contactLastName
contactFirstName
phone
addressLine1
addressLine2
city
state
postalCode
country
salesRepEmployeeNumber
creditLimit
offices
* officeCode
city
phone
addressLine1
addressLine2
state
country
postalCode
territory
payments
* customerNumber
* checkNumber
paymentDate
amount

2	การเรียกดูข้อมูลแบบมีเงื่อนไข	ปฏิบัติการ

จากการเรียนรู้คำสั่ง SELECT แบบพื้นฐานไปแล้วนั้น จะพบว่า MySQL จะทำการแสดงข้อมูลทั้งหมดที่มีอยู่ในตาราง ซึ่งบางครั้งผู้ใช้งานไม่ต้องการดูข้อมูลทั้งหมดแต่ต้องการเลือกดูเพียงแค่บางแถวเท่านั้น ดังนั้นจึงต้องมีการกำหนดเงื่อนไขลงไปในคำสั่ง SELECT เพื่อให้ได้ข้อมูลตามที่ต้องการ

2.1 คำสั่ง SELECT แบบมีเงื่อนไข

รูปแบบของคำสั่งนี้จะเพิ่มส่วน WHERE เข้าไปหลัง FROM เพื่อใช้ในการกำหนดเงื่อนไขในการค้นหาข้อมูลในตาราง โดยมีรูปแบบดังนี้

```
SELECT column_name
FROM table_name
WHERE condition;
```

โดยตัวดำเนินการที่ใช้ในการเปรียบเทียบค่าของข้อมูลได้แก่

=	คือ เท่ากับ	!=	คือ ไม่เท่ากับ
>	คือ มากกว่า	<	คือ น้อยกว่า
>=	คือ มากกว่าหรือเท่ากับ	<=	คือ น้อยกว่าหรือเท่ากับ

เช่น ต้องการหารายชื่อลูกค้าที่อยู่เมือง NYC จะสามารถทำได้ด้วยคำสั่ง

```
SELECT customerName, city
FROM customers
WHERE city = 'NYC';
```

```
+----------------------+------+
| customerName         | city |
+----------------------+------+
| LAND of Toys Inc.    | NYC  |
| Muscle Machine Inc   | NYC  |
| Vitachrome Inc.      | NYC  |
| Classic Legends Inc. | NYC  |
| Microscale Inc.      | NYC  |
+----------------------+------+
5 rows in set (0.01 sec)
```

โดยจะเห็นได้ว่า NYC ที่เป็นชื่อเมืองจะอยู่ในเครื่องหมาย ' ' เนื่องจากข้อมูลที่เปรียบเทียบเป็นแบบ char แต่ถ้าเป็นแบบ integer ไม่ต้องใส่ในเครื่องหมาย ดังตัวอย่างต่อไปนี้

```
SELECT employeeNumber, firstName, lastName
FROM employees
WHERE employeeNumber > 1500;
```

```
+----------------+-----------+----------+
| employeeNumber | firstName | lastName |
+----------------+-----------+----------+
|           1501 | Larry     | Bott     |
|           1504 | Barry     | Jones    |
|           1611 | ANDy      | Fixter   |
|           1612 | Peter     | Marsh    |
|           1619 | Tom       | King     |
|           1621 | Mami      | Nishi    |
|           1625 | Yoshimi   | Kato     |
|           1702 | Martin    | Gerard   |
+----------------+-----------+----------+
8 rows in set (0.01 sec)
```

ในการระบุเงื่อนไขสามารถทำได้มากกว่า 1 เงื่อนไข ด้วยการเชื่อมด้วย AND, or โดยผลของค่าความจริงจะเป็นไปตามหลักตรรกศาสตร์ เช่น ต้องการหารายชื่อลูกค้าที่อยู่ในประเทศ USA และมี creditLIMIT มากกว่า 200,000 สามารถเขียนคำสั่งได้เป็น

```
SELECT customerName, creditLIMIT, country
FROM customers
WHERE country = 'USA'
AND creditLIMIT > 200000;
```

```
+------------------------------+-------------+---------+
| customerName                 | creditLIMIT | country |
+------------------------------+-------------+---------+
| Mini Gifts Distributors Ltd. |      210500 | USA     |
+------------------------------+-------------+---------+
1 row in set (0.01 sec)
```

2.2 การใช้ SELECT กับชุดข้อมูลที่ต้องการ

หากต้องการข้อมูลที่เป็นเซต สามารถใช้คำสงวน “in” เพื่อกำหนดเงื่อนไขในคำสั่งย่อย WHERE โดยมีรูปแบบ คือ

SELECT column_name
FROM table_name
WHERE column_name [**not**] **in** (data set);

เช่น ถ้าต้องการข้อมูลรายชื่อลูกค้าที่อยู่ในเมือง Milan, London หรือ NYC นอกจากจะสามารถใช้ “or” ช่วยในการเลือกข้อมูลที่ต้องการได้แล้ว ยังเขียนคำสั่งเพื่อให้กระชับด้วยคำสงวน “in” ดังตัวอย่าง

```
SELECT customerName, city, country
FROM customers
WHERE city in ('Milan','London','NYC');
+--------------------------------+--------+---------+
| customerName                   | city   | country |
+--------------------------------+--------+---------+
| LAND of Toys Inc.              | NYC    | USA     |
| Muscle Machine Inc             | NYC    | USA     |
| Vitachrome Inc.                | NYC    | USA     |
| Stylish Desk Decors, Co.       | London | UK      |
| Classic Legends Inc.           | NYC    | USA     |
| Microscale Inc.                | NYC    | USA     |
| Frau da Collezione             | Milan  | Italy   |
| Double Decker Gift Stores, Ltd | London | UK      |
+--------------------------------+--------+---------+
8 rows in set (0.02 sec)
```

หรืออาจจะหาข้อมูลที่อยู่ภายนอกเซตที่ต้องการด้วยคำสงวน “not” เช่น ต้องการหาชื่อลูกค้าที่อยู่ในประเทศอังกฤษ และไม่ได้อยู่ในเมือง London, Liverpool สามารถทำได้ดังนี้

```
SELECT customerName, city
FROM customers
WHERE country = 'UK'
AND city not in ('London','Liverpool');
+-------------------+------------+
| customerName      | city       |
+-------------------+------------+
| AV Stores, Co.    | Manchester |
| giftsbymail.co.uk | Cowes      |
+-------------------+------------+
2 rows in set (0.01 sec)
```

2.3 การใช้ SELECT กับข้อมูลที่อยู่ในช่วงที่ต้องการ

หากต้องการแสดงข้อมูลที่เป็นลักษณะช่วงของค่าหนึ่งๆ สามารถเพิ่มคำสงวนไว้ในคำสั่งย่อย WHERE โดยมีรูปแบบเป็นดังนี้

```
SELECT column_name
FROM table_name
WHERE column_name [not] BETWEEN value_1 AND value_2;
```

เช่น ให้แสดงชื่อของลูกค้าที่มีวงเครดิตระหว่าง 10,000 ถึง 30,000 สามารถเขียนคำสั่งได้ 2 รูปแบบ ทั้งที่เป็นการนำเงื่อนไข 2 ส่วนมา AND กัน หรือจะใช้ between ก็ได้ ดังตัวอย่าง

```
SELECT customerName,creditLIMIT, city
FROM customers
WHERE creditLIMIT >= 10000
AND creditLIMIT <= 30000;
+-------------------------+-------------+------------+
| customerName            | creditLIMIT | city       |
+-------------------------+-------------+------------+
| Atelier graphique       |       21000 | Nantes     |
| Auto-Moto Classics Inc. |       23000 | Brickhaven |
| Boards & Toys Co.       |       11000 | Glendale   |
| Royale Belge            |       23500 | Charleroi  |
+-------------------------+-------------+------------+
4 rows in set (0.03 sec)
```

ให้ลองเปรียบเทียบกับคำสั่งที่เป็นคำสงวน between ดังนี้

```
SELECT customerName, creditLIMIT, city
FROM customers
WHERE creditLIMIT BETWEEN 10000 AND 30000;
+-------------------------+-------------+------------+
| customerName            | creditLIMIT | city       |
+-------------------------+-------------+------------+
| Atelier graphique       |       21000 | Nantes     |
| Auto-Moto Classics Inc. |       23000 | Brickhaven |
| Boards & Toys Co.       |       11000 | Glendale   |
| Royale Belge            |       23500 | Charleroi  |
+-------------------------+-------------+------------+
4 rows in set (0.01 sec)
```

2.4 การใช้ SELECT กับข้อมูลที่ตรงกับรูปแบบที่ต้องการ

ในกรณีที่ต้องการหาข้อมูลที่มีรูปแบบตามต้องการ เช่น ชื่อลูกค้าที่ขึ้นต้นด้วย ‘S’ หรือขึ้นต้นด้วย ‘S’ แล้วตามด้วยอักษรอีก 3 ตัว สามารถนำคำสงวน ‘LIKE’ มาเพิ่มในเงื่อนไขได้ดังนี้

```
SELECT column_name
FROM table_name
WHERE column_name LIKE ‘%_’;
```

โดยที่ ‘%’ จะแทนอักษรใดๆ ก็ได้ไม่ระบุจำนวนตัว ในขณะที่ ‘_’ จะแทนอักษรใดๆ เพียง 1 ตัว เช่น ต้องการหาชื่อลูกค้าที่ขึ้นต้นด้วย ‘Toy’ จะสามารถทำได้ดังนี้

```
SELECT customerName, city
FROM customers
WHERE customerName LIKE 'Toy%';
+----------------------+----------+
| customerName         | city     |
+----------------------+----------+
| Toys of FinlAND, Co. | Helsinki |
| Toys4GrownUps.com    | Pasadena |
+----------------------+----------+
2 rows in set (0.03 sec)
```

หรืออาจจะต้องการหาชื่อพนักงานที่ขึ้นต้นด้วย ‘M’ แล้วตามด้วยอักษรอีก 3 ตัว สามารถเขียนคำสั่งได้ดังนี้

```
mysql> SELECT firstName, lastName
    -> FROM employees
    -> WHERE firstName LIKE 'M___';
+-----------+-----------+
| firstName | lastName  |
+-----------+-----------+
| Mary      | Patterson |
| Mami      | Nishi     |
+-----------+-----------+
2 rows in set (0.01 sec)
```

2.5 กิจกรรมท้ายบทเรียน

ให้นักศึกษาเขียนคำสั่ง SQL เพื่อแสดงข้อมูลที่ต้องการดังต่อไปนี้

2.5.1 ให้แสดงรายชื่อลูกค้าที่อยู่ในประเทศ Germany โดยประกอบไปด้วย customerName, city, country และเรียงลำดับตามชื่อลูกค้าจาก a-z

2.5.2 ให้แสดงรายชื่อสินค้าที่มีราคาระหว่าง 30.00 ถึง 50.00

2.5.3 ให้แสดงรายชื่อสินค้าที่มี stock ต่ำกว่า 500 ชิ้น

3	การเรียกดูข้อมูลแบบซับซ้อน	ปฏิบัติการ

เราได้เรียนรู้ SELECT ทั้งแบบพื้นฐาน และแบบมีเงื่อนไขไปแล้วนั้น จะสังเกตได้ว่ามีการเรียกดูข้อมูลจากเพียงตารางเดียว ในขณะที่ฐานข้อมูลที่ใช้อยู่มีตารางอยู่มากมาย ซึ่งในภาคทฤษฎีได้เรียนรู้การออกแบบ และความสัมพันธ์ของคีย์ต่างๆ ในตารางไปแล้ว ในหัวข้อนี้จึงเป็นการเรียกดูข้อมูลที่ซับซ้อนขึ้น

3.1 การเรียกดูข้อมูลจากตารางมากกว่า 1 ตาราง

คำสั่งนี้จะใช้ SELECT เหมือนในหัวข้อก่อนหน้า เพียงแต่อ้างอิงถึงตารางมากกกว่า 1 ตารางเท่านั้นเอง เช่น ต้องการดูรายละเอียดข้อมูลของลูกค้า โดยแสดงพนักงานที่ดูแลลูกค้ารายนั้นๆ ด้วย จากความต้องการรายละเอียดข้างต้นจำเป็นต้องใช้ตาราง 2 ตาราง คือ Customers และ Employees โดยเริ่มจากข้อมูลของ Customers ก่อน

```
SELECT customerName,salesRepEmployeeNumber
FROM Customers
LIMIT 5;
+---------------------------+------------------------+
| customerName              | salesRepEmployeeNumber |
+---------------------------+------------------------+
| Atelier graphique         |                   1370 |
| Signal Gift Stores        |                   1166 |
| Australian Collectors, Co. |                   1611 |
| La Rochelle Gifts         |                   1370 |
| Baane Mini Imports        |                   1504 |
+---------------------------+------------------------+
5 rows in set (0.00 sec)
```

ถ้าใช้เพียงตาราง Customers จะทำให้ไม่สามารถได้ชื่อพนักงานได้ จึงต้องทำการเชื่อมโยงไปยังตาราง Employees ที่มีรหัส และชื่อ ซึ่งสามารถนำรหัสที่ได้จากผลลัพธ์ด้านบนไปหารายชื่อพนักงานได้ดังนี้

```
SELECT employeeNumber,firstName,lastName
FROM employees
WHERE employeeNumber IN (1370,1166,1611,1370,1504);
+----------------+-----------+-----------+
| employeeNumber | firstName | lastName  |
+----------------+-----------+-----------+
|           1166 | Leslie    | Thompson  |
|           1370 | Gerard    | HernANDez |
|           1504 | Barry     | Jones     |
|           1611 | ANDy      | Fixter    |
+----------------+-----------+-----------+
4 rows in set (0.00 sec)
```

ถ้าต้องการเขียนคำสั่งเพียงครั้งเดียวจะต้องรวมทั้ง 2 ตาราง ซึ่งจะมีการเรียกทั้ง Customers และ Employees และการกำหนดเงื่อนไขบางอย่างเข้าช่วย

```
SELECT customers.customerName,customers.salesRepEmployeeNumber,
employees.firstName,employees.lastName
FROM customers,employees
WHERE customers.salesRepEmployeeNumber=employees.employeeNumber
LIMIT 5;
+---------------------------+-----------------------+-----------+-----------+
| customerName              | salesRepEmployeeNumber | firstName | lastName  |
+---------------------------+-----------------------+-----------+-----------+
| Atelier graphique         |                  1370 | Gerard    | HernANDez |
| Signal Gift Stores        |                  1166 | Leslie    | Thompson  |
| Australian Collectors, Co. |                  1611 | ANDy      | Fixter    |
| La Rochelle Gifts         |                  1370 | Gerard    | HernANDez |
| Baane Mini Imports        |                  1504 | Barry     | Jones     |
+---------------------------+-----------------------+-----------+-----------+
5 rows in set (0.08 sec)
```

จากคำสั่งข้างต้น จะเห็นว่ามีการเรียกข้อมูลจาก 2 ตาราง (สังเกตได้จาก FROM customers,employees) และในคำสั่ง SELECT จะต้องเพิ่มชื่อตารางก่อนหน้าชื่อคอลัมน์ด้วย เพื่อให้ MySQL สามารถรู้ว่าเป็นการนำข้อมูลมาจากตารางใด นอกจากนั้นยังต้องกำหนดเงื่อนไขว่าต้องให้ค่าทั้งสองที่มีค่าเท่ากัน คือ salesRepEmployeeNumber และ employeeNumber เพื่อให้ตารางทั้งสองมาเชื่อมโยงกันอย่างถูกต้อง

เราสามารถย่อชื่อตารางเพื่อให้สะดวกในการเรียกใช้ โดยทำการย่อชื่อตารางในส่วนคำสั่ง FROM ดังนั้น จากคำสั่งด้านบนจะเขียนได้อีกแบบดังนี้

```
SELECT c.customerName,c.salesRepEmployeeNumber,
e.firstName,e.lastName
FROM customers as c, employees as e
WHERE c.salesRepEmployeeNumber=e.employeeNumber
LIMIT 5;
+---------------------------+-----------------------+-----------+-----------+
| customerName              | salesRepEmployeeNumber | firstName | lastName  |
+---------------------------+-----------------------+-----------+-----------+
| Atelier graphique         |                  1370 | Gerard    | HernANDez |
| Signal Gift Stores        |                  1166 | Leslie    | Thompson  |
| Australian Collectors, Co. |                  1611 | ANDy      | Fixter    |
| La Rochelle Gifts         |                  1370 | Gerard    | HernANDez |
| Baane Mini Imports        |                  1504 | Barry     | Jones     |
+---------------------------+-----------------------+-----------+-----------+
5 rows in set (0.00 sec)
```

จากคำสั่งด้านบน เป็นการกำหนดชื่อใหม่ให้สั้นลงด้วยคำสงวน “as” ซึ่งจะทำให้เราเรียกตารางแต่ละอันด้วยชื่อใหม่ที่สั้นลง ทำให้สะดวกขึ้นมาก หรือในกรณีที่ ต้องการเรียกดูข้อมูลพนักงานที่ทำหน้าที่ดูแลลูกค้าที่อยู่ในประเทศญี่ปุ่น เราจะใช้การเรียกข้อมูลเพียงแค่ตารางเดียวจะไม่เพียงพอ เพราะถ้าต้องการลูกค้าที่อยู่ประเทศญี่ปุ่นต้องค้นหาในตาราง Customers ในขณะที่ข้อมูลพนักงานจะต้องเรียกดูที่ตาราง Employees ดังนี้

```
SELECT customername, salesRepEmployeeNumber,country
FROM customers
WHERE country='Japan';
+-------------------------+-----------------------+---------+
| customername            | salesRepEmployeeNumber | country |
+-------------------------+-----------------------+---------+
| Osaka Souveniers Co.    |                  1621 | Japan   |
| Tokyo Collectables, Ltd |                  1621 | Japan   |
+-------------------------+-----------------------+---------+
2 rows in set (0.00 sec)
```

หลังจากนั้นจะได้รหัสพนักงานไปเรียกดูข้อมูลในตาราง Employees เพื่อหาชื่อ และนามสกุลของพนักงานคนนั้น จึงต้องทำการเรียกใช้คำสั่งอีกครั้งดังนี้

```
SELECT employeeNumber,firstName,lastName
FROM employees
WHERE employeeNumber = 1621;
+----------------+-----------+----------+
| employeeNumber | firstName | lastName |
+----------------+-----------+----------+
|           1621 | Mami      | Nishi    |
+----------------+-----------+----------+
1 row in set (0.00 sec)
```

จะเห็นได้ว่าเป็นการเรียกดูข้อมูล 2 ครั้ง เราสามารถรวมได้เป็นดังนี้

```
SELECT c.customerName,c.country,e.firstName,e.lastName
FROM customers as c, employees as e
WHERE c.salesRepEmployeeNumber=e.employeeNumber
AND c.country='Japan';
+-------------------------+---------+-----------+----------+
| customerName            | country | firstName | lastName |
+-------------------------+---------+-----------+----------+
| Osaka Souveniers Co.    | Japan   | Mami      | Nishi    |
| Tokyo Collectables, Ltd | Japan   | Mami      | Nishi    |
+-------------------------+---------+-----------+----------+
2 rows in set (0.00 sec)
```

3.2 การเรียกดูข้อมูลแบบซ้อนกัน

หากต้องการข้อมูลที่ละเอียดมากขึ้น การเรียกดูข้อมูลก็จะซับซ้อนเพิ่มขึ้นตามไปด้วย โดยรูปแบบหนึ่งที่สามารถทำได้คือ การเรียกดูข้อมูลแบบซ้อนกัน หรือ Subquery ซึ่งจะทำการประมวลผลในส่วนของการเรียกดูข้อมูลย่อยก่อน รูปแบบทั่วไปแสดงได้ดังนี้

```
SELECT column_name                                  Main Query
FROM table_name
WHERE condition
      (SELECT column_name                           Sub Query
       FROM table_name
       WHERE condition);
```

เช่น อยากรู้ว่ามีบริษัทลูกค้าอะไรบ้าง ที่อยู่เมืองเดียวกับบริษัท Vitachrome Inc. โดยอาจจะแบ่งการเรียกดูข้อมูลออกเป็น 2 ขั้นตอนดังนี้

ขั้นตอนแรก หาว่า Vitachrome Inc. อยู่ที่เมืองอะไร

```
SELECT city
FROM customers
WHERE customerName='Vitachrome Inc.';
+------+
| city |
+------+
| NYC  |
+------+
1 row in set (0.00 sec)
```

ขั้นที่สอง นำเมืองที่ได้จากขั้นตอนแรกมาหาว่ามีบริษัทอะไรอยู่ที่เมืองนี้บ้าง

```
SELECT customerName
FROM customers
WHERE city = 'NYC';
+----------------------+
| customerName         |
+----------------------+
| LAND of Toys Inc.    |
| Muscle Machine Inc   |
| Vitachrome Inc.      |
| Classic Legends Inc. |
| Microscale Inc.      |
+----------------------+
5 rows in set (0.00 sec)
```

แต่เราสามารถเรียกคำสั่ง SQL ซ้อนกันได้ เพื่อทำงานเพียงครั้งเดียว ดังนี้

```
SELECT customerName
FROM customers
WHERE city =
      (SELECT city
      FROM customers
      WHERE customerName='Vitachrome Inc.');
+----------------------+
| customerName         |
+----------------------+
| LAND of Toys Inc.    |
| Muscle Machine Inc   |
| Vitachrome Inc.      |
| Classic Legends Inc. |
| Microscale Inc.      |
+----------------------+
5 rows in set (0.00 sec)
```

โดยการทำงานจะคล้ายกับการแยกเป็น 2 ชุดคำสั่ง ซึ่งจะเริ่มทำที่คำสั่งย่อยก่อน คือ คำสั่งในวงเล็บ หลังจากนั้นจะส่งผลลัพธ์ไปให้อีกชุดคำสั่งเพื่อทำการหาผลลัพธ์ต่อไป ในคำสั่งย่อยนั้นจะต้องมีคอลัมน์ที่ต้องการเพียงคอลัมน์เดียว และประเภทของข้อมูลต้องสอดคล้องกับเงื่อนไขกับคำสั่งหลัก นอกจากนี้หากคำสั่งเรียกดูข้อมูลย่อยมีหลายค่าควรใช้เงื่อนไข ‘in’ แทน ‘=’

3.3 กิจกรรมท้ายบทเรียน

ให้นักศึกษาเขียนคำสั่ง SQL เพื่อแสดงข้อมูลที่ต้องการดังต่อไปนี้

3.3.1 ให้แสดงรายละเอียด ชื่อ-นามสกุลพนักงาน และรหัส office รวมไปถึงเมืองที่ตั้ง office ด้วย

3.3.2 ให้แสดงรายละเอียดชื่อลูกค้า ชื่อ-นามสกุลพนักงานที่ดูแล และหมายเลขการสั่งซื้อ

3.3.3 ให้แสดงหมายเลขคำสั่งซื้อ วันที่สั่ง และสถานการณ์สั่งซื้อ ของบริษัทลูกค้าในเมือง NYC

3.3.4 ให้แสดงรายชื่อ office ทั้งหมดที่มีลูกค้าอยู่ในประเทศ USA

3.3.5 ให้แสดงรายชื่อ-นามสกุลพนักงานที่ลูกค้ามียอดชำระเงินเกิน 100,000 เหรียญ