

04.CRUD-Query

SQL UNION Operator

- คำสั่ง UNION เป็นการรวมข้อมูลจากตาราง 2 ตารางเข้าด้วยกัน โดยมีเงื่อนไขดังต่อไปนี้
 - ตารางทั้งสอง จะต้องมีความคอลัมน์ที่เท่ากัน
 - ชนิดข้อมูลของแต่ละคอลัมน์ของตารางทั้งสองต้องเหมือนกัน
 - ลำดับของคอลัมน์ที่กำหนดในคำสั่ง SELECT จะต้องเรียงลำดับเหมือนกัน

รูปแบบคำสั่ง

```
SELECT column_name(s) FROM table1  
UNION (ALL)
```

```
SELECT column_name(s) FROM table2;
```

ตัวอย่าง

```
SELECT City FROM Customers  
UNION  
SELECT City FROM Suppliers  
ORDER BY City;
```

```
SELECT City, Country FROM Customers  
WHERE Country='Germany'  
UNION ALL  
SELECT City, Country FROM Suppliers  
WHERE Country='Germany'  
ORDER BY City;
```

SQL LIKE Operator

- Like เป็นตัวดำเนินการเปรียบเทียบค่าข้อความสตริงในรูปแบบคำคล้ายคลึงกัน
 - % แทนตัวอักษรอะไรก็ได้ไม่จำกัดจำนวนตัวอักษร หรือไม่มีตัวอักษรเลย
 - _ แทนตัวอักษรหนึ่งตัว

SELECT column1, column2, ...

FROM table_name WHERE column LIKE pattern;

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	หาข้อความที่ขึ้นต้นหน้าด้วยตัวอักษร "a"
WHERE CustomerName LIKE '%a'	หาข้อความที่มีลงท้ายด้วยตัวอักษร "a"
WHERE CustomerName LIKE '%or%'	หาข้อความที่มีตัวอักษร "or" ในตำแหน่งไหนก็ได้
WHERE CustomerName LIKE '_r%'	หาข้อความที่มีตัวอักษร "r" ในตำแหน่งตัวอักษรที่สอง
WHERE CustomerName LIKE 'a_%'	หาข้อความที่ขึ้นต้นหน้าด้วยตัวอักษร "a" และมีความยาวข้อความอย่างน้อย 2 ตัวอักษรขึ้นไป
WHERE CustomerName LIKE 'a__%'	หาข้อความที่ขึ้นต้นหน้าด้วยตัวอักษร "a" และมีความยาวข้อความอย่างน้อย 3 ตัวอักษรขึ้นไป
WHERE ContactName LIKE 'a%o'	หาข้อความที่ขึ้นต้นหน้าด้วยตัวอักษร "a" และลงท้ายด้วยตัวอักษร "o"

SQL SELECT INTO Statement

- The SELECT INTO เป็นคำสั่งในคัดลอกข้อมูลจากตารางเก่า (oldtable) ไปยังตารางใหม่ (newtable)

All Columns	Some Columns
SELECT * INTO newtable [IN externaldb] FROM oldtable WHERE condition;	SELECT column1, column2, ... INTO newtable [IN externaldb] FROM oldtable WHERE condition;
SELECT * INTO CustomersBackup2017 FROM Customers;	SELECT CustomerName, ContactName INTO CustomersBackup2017 FROM Customers;

```
SELECT * INTO CustomersGermany  
FROM Customers  
WHERE Country = 'Germany';
```

```
SELECT Customers.CustomerName, Orders.OrderID  
INTO CustomersOrderBackup2017  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

SQL INSERT INTO SELECT Statement

- The INSERT INTO SELECT เพิ่มข้อมูลในตาราง table2 จากการคัดลอกข้อมูลจากอีกตาราง table1

All Columns	Some Columns
INSERT INTO table2 SELECT * FROM table1 WHERE condition;	INSERT INTO table2 (column1, column2, ...) SELECT column1, column2, ... FROM table1 WHERE condition;

```
INSERT INTO Customers (CustomerName, City, Country)  
SELECT SupplierName, City, Country FROM Suppliers;
```

```
INSERT INTO Customers (CustomerName, City, Country)  
SELECT SupplierName, City, Country FROM Suppliers  
WHERE Country='Germany' ;
```

SQL MIN() and MAX() Functions

- ฟังก์ชัน MIN แสดงค่าต่ำสุดของคอลัมน์

```
SELECT MIN(column_name)  
FROM table_name  
WHERE condition;
```

เช่น **SELECT** MIN(Price) AS SmallestPrice **FROM** Products;

- ฟังก์ชัน MAX แสดงค่าสูงสุดของคอลัมน์

```
SELECT MAX(column_name)  
FROM table_name  
WHERE condition;
```

เช่น **SELECT** MAX(Price) AS LargestPrice **FROM** Products

SQL COUNT(), AVG() and SUM() Functions

- ฟังก์ชัน COUNT แสดงนับจำนวนแถวที่ตรงกับเงื่อนไขที่กำหนด

`SELECT COUNT(column_name) FROM table_name WHERE condition;`

เช่น `SELECT COUNT(ProductID) FROM Products;`

- ฟังก์ชัน AVG แสดงค่าเฉลี่ยของคอลัมน์ที่เป็นตัวเลขเท่านั้น

`SELECT AVG(column_name) FROM table_name WHERE condition;`

เช่น `SELECT AVG(Price) FROM Products`

- ฟังก์ชัน SUM แสดงค่าผลรวมของคอลัมน์ที่เป็นตัวเลขเท่านั้น

`SELECT SUM(column_name) FROM table_name WHERE condition;`

เช่น `SELECT SUM(Unit) FROM Products`

PRODUCTS TABLE

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10	18
2	Chang	1	1	24	19
3	Aniseed Syrup	1	2	12	10
4	Chef Anton's Cajun Seasoning	2	2	48	22
5	Chef Anton's Gumbo Mix	2	2	36	21

SQL Command	Result
SELECT MIN(Price) AS SmallestPrice FROM Products;	
SELECT MAX(Price) AS LargestPrice FROM Products	
SELECT COUNT(ProductID) FROM Products	
SELECT AVG(Price) FROM Products	
SELECT SUM(unit) FROM Products	
SELECT COUNT(ProductID) FROM Products WHERE Price > 15;	
SELECT AVG(Price) FROM Products WHERE CategoryID = 2;	
SELECT SUM(unit) FROM Products WHERE SupplierID = 1;	

PRODUCTS TABLE

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10	18
2	Chang	1	1	24	19
3	Aniseed Syrup	1	2	12	10
4	Chef Anton's Cajun Seasoning	2	2	48	22
5	Chef Anton's Gumbo Mix	2	2	36	21

SQL Command	Result
SELECT CategoryID, COUNT(CustomerID) FROM Products GROUP BY CategoryID;	
SELECT SupplierID, AVG(Price) FROM Products GROUP BY SupplierID ORDER BY COUNT(CustomerID) DESC;	
SELECT CategoryID, SupplierID, SUM(UNIT) FROM Products GROUP BY CategoryID, SupplierID;	https://www.w3schools.com/sql/

SQL GROUP BY Statement

- คำสั่ง GROUP BY เป็นการรวมแถวที่มีค่าเหมือนกันเพื่อใช้ในการคำนวณค่ารวมต่าง ๆ เช่น COUNT, SUM, AVG เป็นต้น เช่น
 - จำนวนสินค้าในแต่ละประเภท
 - ค่าเฉลี่ยคะแนนของนิสิตในแต่ละสาขาวิชา
 - ผลรวมผู้ป่วยโควิด 19 ในแต่ละเดือน

```
SELECT column_name(s)
FROM table_name
GROUP BY column_name(s)
ORDER BY column_name(s);
```

เช่น

```
SELECT CategoryID, COUNT(CustomerID) FROM Products GROUP BY CategoryID;
SELECT SupplierID, AVG(Price) FROM Products GROUP BY SupplierID
ORDER BY AVG(Price) DESC;
```