

# SQL

Structured Query Language

SELECT : เลือกข้อมูลทั้งหมด

SELECT ชื่อ column FROM ชื่อตาราง ;

ตัวเล็กตัวใหญ่ ใช้เนื่องจากกัน

เลือกมากกว่า 1 column : firstname, lastname

AS : เปลี่ยนชื่อ column ตามประสงค์ในนะ

SELECT FirstName as FN, ชื่อใหม่  
LastName as LN  
FROM customers;



/\* .... \*/ : comment

WHERE : กำหนดเงื่อนไข ใช้กับ Aggregate Function ไม่ได้

SELECT \* FROM customers  
WHERE CustomerID = 1; เงื่อนไข ID = 1

หากกล่าว 1 เงื่อนไข ก็ AND/OR :

SELECT \* FROM customers  
WHERE Country = 'Canada' AND State = 'ON';

ORDER BY : เรียงลำดับข้อมูล

SELECT \* FROM customers  
WHERE Country = 'Canada' AND state = 'ON';  
ORDER BY FirstName; เรียงตามตัวอักษรจากหลังไป前往

เรียงจากมากไปน้อย ใช้ DESC (Descending Order)

SELECT \* FROM customers  
WHERE Country = 'Canada' AND state = 'ON';  
ORDER BY FirstName DESC;

LIMIT : จำกัดจำนวนข้อมูล

SELECT \* FROM customers  
WHERE Country = 'Canada' AND state = 'ON';  
ORDER BY FirstName DESC;  
LIMIT 5; แสดงแค่ 5 บรรทัด

INSERT INTO : เพิ่มข้อมูล

INSERT INTO customers ไม่ต้องระบุ  
VALUES ('60', 'Soon', 'Thanakorn', ..., NULL); ใส่เชิงตรรกะด้วย field!

แบบระบุ field  
INSERT INTO customers(FirstName, LastName, Email)  
VALUES ('Soon', 'Thanakorn', 'tfp@gmail.com');

UPDATE, SET, WHERE : ការផ្តល់ឯកសារ

```
UPDATE customers ទីតាំង
SET FirstName = "Amy", LastName = "Tanamas" តើមីនាមីន
WHERE Customer Id = 61; នៅនៅ
```

DELETE : លើកសារ

```
DELETE FROM customers ទីតាំង
WHERE Customer Id = 61; នៅនៅ
```

## Aggregate Functions (ដែលសង SELECT)

COUNT : បញ្ជីចំណេះចំណេះ តិច តារាង

```
SELECT COUNT(FirstName) FROM Customers;
```

Avg : គ្រាប់លើខ្លួន តិច field តិច តារាង

```
SELECT AVG(Total) FROM invoices;
```

SUM : សរុបតុលាឌ តិច field តិច តារាង

```
SELECT SUM(Total) FROM invoices;
```

MAX : សាកល្បែង តិច field តិច តារាង

```
SELECT MAX(Total) FROM invoices;
```

MIN : នូវតុលាឌ តិច field តិច តារាង

```
SELECT MIN(Total) FROM invoices;
```

GROUP BY : ការចែងក្នុងរបៀបរាយ value ក្នុង field

```
SELECT COUNT(Customer Id), City
FROM customers
GROUP BY City
```

5 Bangkok  
3 Berlin  
7 Soul

HAVING : នឹងពេល WHERE នៃការណើ Aggregate Function

```
SELECT COUNT(Customer Id) AS People, Country
FROM customers
GROUP BY Country
HAVING People >= 2 នៅនៅ
ORDER BY People DESC
```

```
SELECT
    column_name(s)
FROM
    table_name
WHERE
    condition(s)
GROUP BY
    column_name(s)
HAVING
    condition(s)
ORDER BY
    column_name(s)
LIMIT
    number of rows;
```

## Operator

NOT : ไม่เท่ากับ

```
SELECT * FROM employees
WHERE NOT Title = "General Manager"; หัก掉หัวข้อ
```

IN : เอกทุกอย่างใน()

```
SELECT * FROM employees
WHERE Title IN ("Sales Support Agent", "Sales Manager", "General Manager");
```

แมลงขี้มูด 3 อย่าง

LIKE : หาข้อมูลที่มีส่วนต่อไปนี้ %

% คือ ตัวกำหนด Keyword ในภาษาค้นหา

```
SELECT * FROM employees
WHERE FirstName LIKE "%a"; ชื่อลงทะเบียน a
      "a%" ชื่อพื้นทั่วๆ a
      "%a%" ห้าม a อยู่ระหว่างชื่อ
```

BETWEEN : ช่องจังหวะช่วง

```
SELECT * FROM employees
WHERE CustomerId BETWEEN 10 AND 35; ช่วง 10 - 35
```

IS NULL : ไม่มีข้อมูล / ว่าง

```
SELECT * FROM employees
WHERE Fax IS NULL; ห้ามกรอกข้อมูล Fax (ว่าง)
```

Join : เอกข้อมูล 2 \_join กับอีกข้อมูลหนึ่ง

(INNER) JOIN

**เข้าใจความแตกต่างการ JOIN สำหรับ SQL** สรุปความแตกต่าง รูปแบบการ **JOIN TABLE** สำหรับ

**Structured Query Language: SQL**

**(INNER) JOIN**  
ทำการรักเกิร์นกลับเฉพาะข้อมูลที่มีการจับคู่กันได้เท่านั้น  
(ล้วนๆ หรือ Intersection กันหมด)

Table : Customers		Table : Orders	
CustomerID	CustomerName	OrderID	CustomerID
1	Alfreds Futterkiste	10248	2
2	Ana Trujillo Emparedados y helados	10249	1
3	Antonio Moreno Taquería	10250	4
4	Around the Horn	...	...

มาลองทำการ **INNER JOIN** กัน!

```
SELECT Orders.OrderID, Customers.CustomerName
FROM Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

ผลลัพธ์ที่ได้คือ สอง join รับ Order

OrderID	CustomerName
10248	Ana Trujillo Emparedados y helados
10249	Alfreds Futterkiste
10250	Around the Horn

บุ๊ค ดู ผลลัพธ์ความต้องเห็น แบบ ... แท็บ ที่กรุณา ดาวน์โหลด ส่งใบ Messenger

## LEFT (OUTER) JOIN

สรุปความแตกต่างการ JOIN ส่าหรับ SQL รูปแบบการ JOIN TABLE สำหรับ Structured Query Language: SQL



### LEFT (OUTER) JOIN

ทำการรีเทิร์นกลับทั้งหมดของตารางผึ้งช้าย และ ส่วนที่จับกู่กันได้กับตารางผึ้งขาว

Table : Customers		Table : Orders					
CustomerID	CustomerName	Address	OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
1	Alfreds Futterkiste	Obere Str. 57	10248	2	5	1996-07- 3	3
2	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	10249	8	1	1996-07- 1	4
3	Antonio Moreno Taquería	Mataderos 2312	10250	4	4	1996-07- 2	5
4	Around the Horn	120 Hanover Sq.	..	..	..	1996-07- 08	..
..	..	..	..	..	..	..	..

มาลองทำการ LEFT (OUTER) JOIN กัน !

```
SELECT Customers.CustomerName, Orders.OrderID FROM Customers
LEFT JOIN Orders ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

ผลลัพธ์ที่ได้จะเรียงตามัวอักษรของชื่อสู่ก้าว แล้วแสดงผลด้านล่าง..

CustomerName	OrderID
Alfreds Futterkiste	null
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365

ปุ่ม กู้ไข้ ค ทดสอบความถูกต้อง แก้ไข .. แม่ข่ายภาษา ด่วนเลือก ส่งใบ Messenger

## RIGHT (OUTER) JOIN

สรุปความแตกต่างการ JOIN ส่าหรับ SQL รูปแบบการ JOIN TABLE สำหรับ Structured Query Language: SQL



### RIGHT (OUTER) JOIN

ทำการรีเทิร์นกลับทั้งหมดของตารางผึ้งขาว และ ส่วนที่จับกู่กันได้กับตารางผึ้งช้าย

Table : Employees				Table : Orders				
EmployeeID	LastName	FirstName	BirthDate	OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
1	Davolio	Nancy	12/8/1968	10248	2	5	1996-07- 3	3
2	Fuller	Andrew	2/19/1952	10249	1	1	1996-07- 1	4
3	Leverling	Janet	8/30/1963	10250	4	4	1996-07- 2	5
4	Peacock	Margaret	9/19/1958	..	..	..	1996-07- 08	..
..	..	..	..	..	..	..	..	..

มาลองทำการ RIGHT (OUTER) JOIN กัน !

```
SELECT Orders.OrderID, Employees.LastName, Employees.FirstName
FROM Orders RIGHT JOIN Employees
ON Orders.EmployeeID = Employees.EmployeeID ORDER BY Orders.OrderID;
```

ผลลัพธ์ที่ได้จะเรียงตามหมายเลข OrderID และแสดงผลด้านล่าง..

OrderId	LastName	FirstName
10248	West	Adam
10249	Buchanan	Steven
10250	Suyama	Michael
10251	Peacock	Margaret
10251	Leverling	Janet

ปุ่ม กู้ไข้ ค ทดสอบความถูกต้อง แก้ไข .. แม่ข่ายภาษา ด่วนเลือก ส่งใบ Messenger

## FULL (OUTER) JOIN

สอนความแยกต่างกันการ JOIN สำหรับ SQL รูปแบบการ **JOIN TABLE** สำหรับ **BorntoDev**

### Structured Query Language: SQL

**FULL (OUTER) JOIN**  
ทำการรีเทิร์นกลับเหมือน LEFT และ RIGHT แต่เป็นการศึกษาค่าของตารางที่ไม่ได้จับคู่ของทั้งสองตารางด้วย

**Table : Customers**

CustomerID	CustomerName	Address
1	Alfreds Futterkiste	Obere Str. 57
2	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222
3	Antonio Moreno Taquería	Mataderos 2312

**Table : Orders**

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10308	2	7	1996-09- 3	18
10309	37	3	1996-09- 1	19
10310	77	8	1996-09- 2	20

มาลองทำ Full (Outer) JOIN กัน!

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers FULL OUTER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID ORDER BY Customers.CustomerName;;
```

ผลลัพธ์ที่ได้คือ ..

OrderID	CustomerName
Alfreds Futterkiste	10308
Ana Trujillo Emparedados y helados	10365
Antonio Moreno Taquería	10382
	10351

## CASE :

```
SELECT InvoiceId, Total,
CASE
    WHEN Total >= 10 THEN "AAA"
    WHEN Total < 10 THEN "AA"
    ELSE "NONE"
END AS Result;
FROM invoices
```

## SubQueries

```
SELECT column-names
FROM table-name1
WHERE value IN (SELECT column-name
                  FROM table-name2
                  WHERE condition)
```

```
SELECT ProductName
FROM Product
WHERE Id IN (SELECT ProductId
              FROM OrderItem
              WHERE Quantity > 100)
```

PRODUCT	ORDERITEM
id	id
ProductName	OrderId
Supplier	ProductId
UnitPrice	UnitPrice
Quantity	Quantity

## การ Comment

```
SELECT * FROM Customers -- Select All
```