



ความรู้พื้นฐานเกี่ยวกับการเขียนชุดคำสั่ง

นาย ธนพล สวรักษ์

นักวิชาการคอมพิวเตอร์ กองแผนงาน

ภาษา SQL คืออะไร

- ภาษา SQL (Standard Query Language) คือ ภาษาคอมพิวเตอร์ ที่ใช้ในการจัดการฐานข้อมูลโดยเฉพาะ เช่น สร้างฐานข้อมูล เพิ่มข้อมูล แก้ไขข้อมูล ลบข้อมูล เป็นต้น
- ภาษา SQL เป็นภาษามาตรฐานสำหรับระบบฐานข้อมูล พัฒนาขึ้นมาโดยบริษัท IBM
- ภาษา SQL เป็นส่วนหนึ่งของระบบฐานข้อมูลที่ได้รับความนิยม ง่ายต่อความเข้าใจ โดยใช้ภาษาอังกฤษในการสื่อในเวลาเขียนคำสั่ง
- ภาษา SQL จะพบในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) ที่ได้รับความนิยมในปัจจุบัน เช่น MS Access , MS SQL Server , Oracle , MySQL เป็นต้น



ภาษา SQL แบ่งออกเป็น ๓ ส่วน คือ

- ภาษาสำหรับนิยามข้อมูล (Data Definition Language-DDL)
- ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language: DML)
- ภาษาควบคุมข้อมูล (Data Control Language : DCL)



ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL)

เป็นคำสั่งที่ใช้ในการกำหนดโครงสร้างของฐานข้อมูล ประกอบด้วย

- **CREATE DATABASE** : สร้างฐานข้อมูลใหม่
- **DROP DATABASE** : ลบฐานข้อมูล
- **CREATE TABLE** : สร้างตารางข้อมูลใหม่ในฐานข้อมูล
- **DROP TABLE** : ลบตารางข้อมูลในฐานข้อมูล
- **ALTER TABLE** : แก้ไขโครงสร้างตารางข้อมูล
- **CREATE INDEX** : สร้างดัชนีในตาราง
- **DROP INDEX** : ลบดัชนีในตาราง
- **CREATE VIEW** : สร้างวิว
- **DROP VIEW** : ลบวิว



ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language: DML)

เป็นคำสั่งที่ใช้ในการจัดการข้อมูลในตารางข้อมูล ประกอบด้วย

- **INSERT INTO** : เพิ่มรายการข้อมูลใหม่ในตาราง
- **UPDATE** : แก้ไขรายการข้อมูลในตาราง
- **DELETE** : ลบรายการข้อมูลในตาราง
- **SELECT** : เรียกดูหรือสืบค้นข้อมูลจากตาราง



ภาษาควบคุมข้อมูล

(Data Control Language : DCL)

เป็นคำสั่งที่ใช้ในการกำหนดสิทธิการใช้งานฐานข้อมูลให้แก่ผู้ใช้ ประกอบด้วย

- **GRANT** : กำหนดสิทธิในการใช้งานวัตถุต่าง ๆ ในฐานข้อมูลให้แก่ผู้ใช้
- **REVOKE** : ยกเลิกสิทธิในการใช้งานวัตถุต่าง ๆ ในฐานข้อมูลจากผู้ใช้



คำสั่งค้นหาข้อมูล (Query Statement)

คำสั่ง **SELECT** เป็นคำสั่งการเรียกดูข้อมูล หรือ ค้นหาข้อมูล ตามเงื่อนไขที่ระบุ มีรูปแบบการใช้งานที่ง่ายเพื่อช่วยในการค้นหาข้อมูลที่ซับซ้อน ดังรูปแบบ

SELECT <ชื่อคอลัมน์ที่ต้องการให้แสดง, ... , ... หรือ * เพื่อแสดงทุกคอลัมน์>
FROM <ชื่อตารางที่ต้องการสืบค้นข้อมูล, ... , ...>
WHERE <เงื่อนไขในการแสดงรายการข้อมูล>
GROUP BY <ชื่อคอลัมน์ที่ใช้ในการจัดกลุ่มข้อมูล , ... , ...>
HAVING <เงื่อนไขในการจัดกลุ่มข้อมูลเพื่อแสดงผล>
ORDER BY <ชื่อคอลัมน์ที่ใช้ในการเรียงลำดับข้อมูล (ASC || DESC)>



คำสั่งค้นหาข้อมูล (Query Statement)

ตัวอย่าง ๑ จงแสดงข้อมูลนักศึกษาจากตาราง Students ทุกรายการ

```
SELECT * FROM Students
```

--จะได้ผลลัพธ์คือข้อมูลในตาราง Students ทุกแถว ทุกคอลัมน์--

std_id	std_fname	std_lname	std_address	std_major	std_gpa	fac_id
1234567801	Bob	Norbert	USA	IT	3	Sc
1234567821	William	Pilgrim	AUS	CS	2.5	Sc
1234567822	Luke	Brazzi	USA	PHY	3.5	Sc
1234567840	Homer	Wells	AUS	CS	2.5	Sc
2345678910	Candy	Kendall	NZ	FIN	2.75	Msc
2345678920	Roberto	Morales	AUS	FIN	3.25	Msc
2345789015	Tess	Dodge	CAN	ECON	3.75	Msc
4567890122	Joe	Estrada	CAN	Art	3.25	En
4567890133	Cristopher	Colan	USA	MUS	2.75	En
5678901235	Wally	Kendall	NZ	GSc	3.5	Ed
5678901240	Mariah	Dodge	CAN	Gsc	2.5	Ed

ตัวอย่าง ๒ จงแสดงข้อมูลรหัสนักศึกษา, ชื่อ, นามสกุล และสาขาวิชาของทุกคน

```
SELECT std_id, std_fname, std_lname, std_major  
FROM Students
```

--ผลลัพธ์--

std_id	std_fname	std_lname	std_major
1234567801	Bob	Norbert	IT
1234567821	William	Pilgrim	CS
1234567822	Luke	Brazzi	PHY
1234567840	Homer	Wells	CS
2345678910	Candy	Kendall	FIN
2345678920	Roberto	Morales	FIN
2345789015	Tess	Dodge	ECON
4567890122	Joe	Estrada	Art
4567890133	Cristopher	Colan	MUS
5678901235	Wally	Kendall	GSc
5678901240	Mariah	Dodge	Gsc

คำสั่งค้นหาข้อมูล (Query Statement) (SELECT DISTINCT)

คำสั่ง **SELECT DISTINCT** ใช้สำหรับขจัดรายการข้อมูลที่ซ้ำกัน โดยจะแสดงแถวข้อมูลที่ซ้ำกันเพียงรายการเดียว

ตัวอย่าง ๓ จงแสดงว่ามีข้อมูลตำแหน่งวิชาใดบ้างที่อาจารย์ได้รับ

SELECT ins_rank FROM Instructor

ins_rank
ASSC
PROF
ASSC
PROF
ASST
ASST

--ผลลัพธ์--



SELECT DISTINCT ins_rank FROM Instructor

ins_rank
ASSC
PROF
ASST

คำสั่งค้นหาข้อมูล (Query Statement) (คำสั่ง WHERE)

คำสั่ง **WHERE** ใช้ในการกำหนดเงื่อนไขในการแสดงผล มีรูปแบบดังนี้

```
SELECT <column_name,... | *>  
FROM <table_name,...>  
WHERE column_name operator value
```

โดย

Column_name คือ แอททริบิวต์ที่ใช้เป็นเงื่อนไข

Operator คือ ตัวดำเนินการต่าง ๆ เช่น = , <> , > , < , >= , <= , LIKE , BETWEEN , IN

Value คือ ค่าที่นำมาใช้เปรียบเทียบ

คำสั่งค้นหาข้อมูล (Query Statement) (Operator AND)

คำสั่ง **AND** ใช้เชื่อมเงื่อนไข ต้องให้ผลลัพธ์ที่เป็นจริงทุกเงื่อนไขจึงจะแสดงรายการข้อมูลนั้น

ตัวอย่าง ๔ จงแสดงข้อมูลนักศึกษาที่อยู่ประเทศแคนาดา(CAN) และ ได้เกรดเฉลี่ยมากกว่า ๓.๐๐

```
SELECT * FROM Students  
WHERE std_address = 'CAN' AND std_gpa > ๓.๐
```

--ผลลัพธ์--

std_id	std_fname	std_lname	std_address	std_major	std_gpa	fac_id
2345789015	Tess	Dodge	CAN	ECON	3.75	Msc
4567890122	Joe	Estrada	CAN	Art	3.25	En

คำสั่งค้นหาข้อมูล (Query Statement) (Operator OR)

คำสั่ง **OR** ใช้เชื่อมเงื่อนไข จะแสดงรายการข้อมูลที่สอดคล้องเมื่อเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริงหรือทุกเงื่อนไขเป็นจริง ถ้าเงื่อนไขเป็นเท็จทุกกรณีจะไม่แสดงข้อมูล

ตัวอย่าง ๕ จงแสดงข้อมูลนักศึกษา ที่สังกัดคณะรหัส Sc หรือ นักศึกษาที่อยู่ประเทศ USA

```
SELECT * FROM Students  
WHERE fac_id = 'Sc' OR std_address = 'USA'
```

--ผลลัพธ์--

std_id	std_fname	std_lname	std_address	std_major	std_gpa	fac_id
1234567801	Bob	Norbert	USA	IT	3	Sc
1234567821	Wiliam	Pilgrim	AUS	CS	2.5	Sc
1234567822	Luke	Brazzi	USA	PHY	3.5	Sc
1234567840	Homer	Wells	AUS	CS	2.5	Sc
4567890133	Cristopher	Colan	USA	MUS	2.75	En

คำสั่งค้นหาข้อมูล (Query Statement) (Operator NOT)

คำสั่ง **NOT** ใช้กลับค่าให้เป็นค่าตรงข้าม เช่น จริงกลับค่าเป็น เท็จ หรือ เท็จกลับค่าเป็นจริง

ตัวอย่าง ๒ จงแสดงข้อมูลนักศึกษา ที่ไม่สังกัดคณะรหัส Sc

```
SELECT * FROM Students  
WHERE NOT (fac_id = 'Sc')
```

--ผลลัพธ์--

std_id	std_fname	std_lname	std_address	std_major	std_gpa	fac_id
2345678910	Candy	Kendall	NZ	FIN	2.75	Msc
2345678920	Roberto	Morales	AUS	FIN	3.25	Msc
2345789015	Tess	Dodge	CAN	ECON	3.75	Msc
4567890122	Joe	Estrada	CAN	Art	3.25	En
4567890133	Cristopher	Colan	USA	MUS	2.75	En
5678901235	Wally	Kendall	NZ	GSc	3.5	Ed
5678901240	Mariah	Dodge	CAN	Gsc	2.5	Ed

คำสั่งค้นหาข้อมูล (Query Statement) (Operator BETWEEN ...AND...)

คำสั่ง **BETWEEN ... AND ...** ใช้สำหรับกรองข้อมูลในช่วงข้อมูลระหว่างค่าสองค่า เช่น ราคาตั้งแต่ ๑๐๐ จนถึง ๕๐๐ , วันที่สั่งซื้อ ตั้งแต่ ๐๑/๐๑/๒๐๑๐ ถึง ๓๐/๐๖/๒๐๑๐

ตัวอย่าง ๗ จงแสดงข้อมูลนักศึกษา ที่ได้เกรดตั้งแต่ ๓.๐ ถึง ๓.๕

```
SELECT std_id, std_fname, std_lname, std_gpa
FROM Students
WHERE std_gpa BETWEEN ๓.๐ AND ๓.๕
```

--ผลลัพธ์--

std_id	std_fname	std_lname	std_gpa
1234567801	Bob	Norbert	3.0
1234567822	Luke	Brazzi	3.5
2345678920	Roberto	Morales	3.25
4567890122	Joe	Estrada	3.25
5678901235	Wally	Kendall	3.5

*** ตัวดำเนินการ **BETWEEN ... AND ...** ใช้เปรียบเทียบข้อมูลที่ เป็น ตัวเลข , ข้อความ และ วัน/เวลา ***

คำสั่งค้นหาข้อมูล (Query Statement) (Operator IN)

คำสั่ง **IN** ใช้สำหรับกรองข้อมูลแบบหลายค่าตามรายการที่ระบุในตัวดำเนินการ **IN**

ตัวอย่าง ๘ จงแสดงข้อมูลนักศึกษา ที่อยู่ในประเทศ อเมริกา(USA) หรือ แคนาดา (CAN)

```
SELECT * FROM Students  
WHERE std_address IN ('USA' , 'CAN');
```

--ผลลัพธ์--

std_id	std_fname	std_lname	std_address	std_major	std_gpa	fac_id
1234567801	Bob	Norbert	USA	IT	3	Sc
1234567822	Luke	Brazzi	USA	PHY	3.5	Sc
2345789015	Tess	Dodge	CAN	ECON	3.75	Msc
4567890122	Joe	Estrada	CAN	Art	3.25	En
4567890133	Cristopher	Colan	USA	MUS	2.75	En
5678901240	Mariah	Dodge	CAN	Gsc	2.5	Ed

*** ข้อสังเกต คำสั่ง **IN** เหมือนกับ การใช้ตัวดำเนินการ **OR** ที่เปรียบเทียบค่าในคอลัมน์เดียวกัน เช่น **WHERE std_address = 'USA' OR std_address = 'CAN'**

คำสั่งค้นหาข้อมูล (Query Statement) (Operator LIKE)

คำสั่ง **LIKE** ใช้สำหรับกรองข้อมูลที่มีข้อความตามรูปแบบที่กำหนด

ตัวอย่าง ๙ จงแสดงข้อมูลนักศึกษาที่มีชื่อขึ้นต้นด้วย ตัวอักษร C

```
SELECT * FROM Students  
WHERE std_fname LIKE 'C%'
```

--ผลลัพธ์--

std_id	std_fname	std_lname	std_address	std_major	std_gpa	fac_id
2345678910	Candy	Kendall	NZ	FIN	2.75	Msc
4567890133	Cristopher	Colan	USA	MUS	2.75	En

ตัวอย่าง ๑๐ จงแสดงข้อมูลชื่อ , นามสกุลของนักศึกษาที่ภายในนามสกุลมีรูปแบบ M_r_les

```
SELECT std_fname, std_lname FROM Students  
WHERE std_lname LIKE 'M_r_les'
```

--ผลลัพธ์--

std_fname	std_lname
Roberto	Morales

คำสั่งค้นหาข้อมูล (Query Statement) (Operator LIKE)

ตัวอย่าง ๑๑ จงแสดงข้อมูลชื่อ , นามสกุลของนักศึกษาที่มีชื่อขึ้นต้นด้วยตัวอักษร B หรือ C หรือ H

```
SELECT std_fname, std_lname  
FROM Students  
WHERE std_fname LIKE '[BCH]%'
```

--ผลลัพธ์--

std_fname	std_lname
Bob	Norbert
Homer	Wells
Candy	Kendall
Cristopher	Colan

คำสั่งค้นหาข้อมูล (Query Statement) (Operator TOP)

คำสั่ง **TOP** ใช้ในการกำหนดจำนวนรายการที่ต้องการแสดง

ตัวอย่าง ๑๓ จงแสดงข้อมูลนักศึกษา ๓ รายการแรก

```
SELECT TOP ๓ * FROM Students
```

—ผลลัพธ์—

std_id	std_fname	std_lname	std_address	std_major	std_gpa	fac_id
1234567801	Bob	Norbert	USA	IT	3	Sc
1234567821	William	Pilgrim	AUS	CS	2.5	Sc
1234567822	Luke	Brazzi	USA	PHY	3.5	Sc

คำสั่งค้นหาข้อมูล (Query Statement) (Alias)

คำสั่ง **AS** ใช้ในการกำหนดชื่อนามแฝงให้กับคอลัมน์ หรือ ตาราง ในการแสดงข้อมูลที่ได้จากสืบค้น

ตัวอย่าง_๓๓ กำหนดนามแฝงให้กับแอททริบิวต์ ชื่อ และ นามสกุล

```
SELECT std_fname AS ชื่อ , std_lname AS นามสกุล  
FROM Students  
WHERE std_address = 'USA'
```

--ผลลัพธ์--

ชื่อ	นามสกุล
Bob	Norbert
Luke	Brazzi
Cristopher	Colan

คำสั่งค้นหาข้อมูล (Query Statement) (ORDER BY)

คำสั่ง **ORDER BY** ใช้สำหรับจัดเรียงผลลัพธ์จากการสืบค้นข้อมูล

ตัวอย่าง ๑๔ จงแสดงข้อมูลนักศึกษาจัดเรียงข้อมูลจากเกรดเฉลี่ยจากมากไปหาน้อย

```
SELECT * FROM Students  
ORDER BY std_gpa DESC
```

--ผลลัพธ์--

std_id	std_fname	std_lname	std_address	std_major	std_gpa	fac_id
2345789015	Tess	Dodge	CAN	ECON	3.75	Msc
5678901235	Wally	Kendall	NZ	GSc	3.5	Ed
1234567822	Luke	Brazzi	USA	PHY	3.5	Sc
4567890122	Joe	Estrada	CAN	Art	3.25	En
2345678920	Roberto	Morales	AUS	FIN	3.25	Msc
1234567801	Bob	Norbert	USA	IT	3	Sc
4567890133	Cristopher	Colan	USA	MUS	2.75	En
2345678910	Candy	Kendall	NZ	FIN	2.75	Msc
5678901240	Mariah	Dodge	CAN	Gsc	2.5	Ed
1234567840	Homer	Wells	AUS	CS	2.5	Sc
1234567821	William	Pilgrim	AUS	CS	2.5	Sc

โดย

ASC (Ascending) เป็นการจัดเรียงข้อมูลจากน้อยไปหามาก

DESC (Descending) เป็นการจัดเรียงข้อมูลจากมากไปหาน้อย

*** หากมีคำสั่ง ORDER BY แต่ไม่กำหนดชนิดการจัดเรียงหลัง column name จะถือว่าเป็นการเรียงแบบ ASC ***

คำสั่งค้นหาข้อมูล (Query Statement) (GROUP BY)

คำสั่ง **GROUP BY** ใช้สำหรับจัดกลุ่มข้อมูลตามคอลัมน์ที่กำหนด ส่วนใหญ่จะใช้คำสั่ง **GROUP BY** ร่วมกับการใช้ฟังก์ชันการรวม (aggregate function) เช่น ต้องการแสดงข้อมูลจากหลาย ๆ คอลัมน์ และมีคอลัมน์ที่ใช้ฟังก์ชันการรวมอยู่ภายในคำสั่ง **SELECT** จะต้องจัดกลุ่มข้อมูลตามคอลัมน์ต่าง ๆ ทั้งหมด ยกเว้นคอลัมน์ที่ใช้ฟังก์ชันการรวม

ตัวอย่าง ๑๕ จะแสดงรหัสคณะและจำนวนนักศึกษาที่สังกัดในแต่ละคณะ จัดเรียงข้อมูลตามจำนวนนักศึกษาจากมากไปหาน้อย

```
SELECT fac_id, COUNT(*) AS Count_of_students
FROM Students
GROUP BY fac_id
ORDER BY COUNT(*) DESC
```

ในคำสั่ง **GROUP BY** จะต้องระบุคอลัมน์ปกติทุกคอลัมน์ ยกเว้น คอลัมน์ที่มีการใช้ฟังก์ชันการรวม

--ผลลัพธ์--

fac_id	Count_Of_students
Sc	4
Msc	3
En	2
Ed	2



ฟังก์ชันรวมของภาษา SQL

ภาษา SQL ตามมาตรฐาน ISO นิยามฟังก์ชันการรวม (Aggregate function) ไว้ ๕ ฟังก์ชัน ดังนี้

COUNT: จำนวนข้อมูลของคอลัมน์ที่ระบุ

MAX: ค่าข้อมูลที่สูงสุดของคอลัมน์ที่กำหนด

MIN: ค่าข้อมูลที่ต่ำสุดของคอลัมน์ที่กำหนด

SUM: ค่าผลรวมของคอลัมน์ที่กำหนด

AVG: ค่าเฉลี่ยของคอลัมน์ที่กำหนด

ฟังก์ชันของภาษา SQL (COUNT)

ฟังก์ชัน **COUNT** ใช้ในการแสดงจำนวนรายการข้อมูลที่สอดคล้องกับเงื่อนไข

Std_id	Std_fname	Std_lnam	fac_id
2345789015	Bob	Brazzi	Msc
5678901235	Norbert	Mariah	Ed
1234567822	Candy	Dodge	Sc
4567890122	Kendall	Roberto	En
2345678920	Cristopher	Morales	Msc
1234567801	Colan	Tess	Sc
4567890133	Homer	Dodge	En
2345678910	Wells	Wally	Msc
5678901240	Joe	Kendall	En
1234567840	Estrada	William	Sc
1234567821	Luke	Pilgrim	Sc

ตัวอย่าง ๑๖ จงแสดงจำนวนนักศึกษาทั้งหมด

```
SELECT COUNT (*) AS All_students  
FROM Students ;
```

--ผลลัพธ์--

All_Students
11

ตัวอย่าง ๑๗ จงแสดงจำนวนนักศึกษาที่สังกัดคณะ Sc

```
SELECT COUNT (*) AS Science_students  
FROM Students  
Where fac_id = 'Sc'
```

--ผลลัพธ์--

Science_Students
4

ฟังก์ชันของภาษา SQL (MAX)

ฟังก์ชัน **MAX** ใช้ในการหาค่าสูงสุดของคอลัมน์ที่กำหนด

ตัวอย่าง ๑๘ จงแสดงข้อมูลค่าเงินเดือนที่สูงสุดของอาจารย์

T_id	T_fname	T_lname	fac_id	Salary
2345789015	Bob	Brazzi	Msc	10,142
5678901235	Norbert	Mariah	Ed	14,324
1234567822	Candy	Dodge	Sc	42,442
4567890122	Kendall	Roberto	En	41,135
2345678920	Cristopher	Morales	Msc	12,113
1234567801	Colan	Tess	Sc	13,213
4567890133	Homer	Dodge	En	44,123
2345678910	Wells	Wally	Msc	32,121
5678901240	Joe	Kendall	En	13,214
1234567840	Estrada	William	Sc	23,142
1234567821	Luke	Pilgrim	Sc	15,321

```
SELECT MAX (Salary) AS MAX_of_Salary  
FROM Instructor ;
```

—ผลลัพธ์—



MAX_of_Salary
44,123

ฟังก์ชันของภาษา SQL (MIN)

ฟังก์ชัน **MIN** ใช้ในการหาค่าต่ำสุดของคอลัมน์ที่กำหนด

ตัวอย่าง ๑๗ จงแสดงข้อมูลค่าเงินเดือนที่น้อยที่สุดของอาจารย์

T_id	T_fname	T_lname	fac_id	Salary
2345789015	Bob	Brazzi	Msc	10,142
5678901235	Norbert	Mariah	Ed	14,324
1234567822	Candy	Dodge	Sc	42,442
4567890122	Kendall	Roberto	En	41,135
2345678920	Cristopher	Morales	Msc	12,113
1234567801	Colan	Tess	Sc	13,213
4567890133	Homer	Dodge	En	44,123
2345678910	Wells	Wally	Msc	32,121
5678901240	Joe	Kendall	En	13,214
1234567840	Estrada	William	Sc	23,142
1234567821	Luke	Pilgrim	Sc	15,321

```
SELECT MIN(salary) AS MIN_of_Salary  
FROM Instructor
```

--ผลลัพธ์--



MIN_of_Salary
10,142

ฟังก์ชันของภาษา SQL (SUM)

ฟังก์ชัน **SUM** ใช้ในการหาค่าผลรวมของคอลัมน์ที่เป็นตัวเลข

ตัวอย่าง ๒๑ จงแสดงข้อมูลเงินเดือนรวมของอาจารย์ทุกคน

T_id	T_fname	T_lname	fac_id	Salary
2345789015	Bob	Brazzi	Msc	10,142
5678901235	Norbert	Mariah	Ed	14,324
1234567822	Candy	Dodge	Sc	42,442
4567890122	Kendall	Roberto	En	41,135
2345678920	Cristopher	Morales	Msc	12,113
1234567801	Colan	Tess	Sc	13,213
4567890133	Homer	Dodge	En	44,123
2345678910	Wells	Wally	Msc	32,121
5678901240	Joe	Kendall	En	13,214
1234567840	Estrada	Wiliam	Sc	23,142
1234567821	Luke	Pilgrim	Sc	15,321

```
SELECT SUM(salary) AS Sum_of_Salary  
FROM Instructor
```

--ผลลัพธ์--



MAX_of_Salary
261,290

ฟังก์ชันของภาษา SQL (AVG)

ฟังก์ชัน **AVG** ใช้ในการหาค่าเฉลี่ยของคอลัมน์ที่เป็นตัวเลข

ตัวอย่าง ๒๑ จงแสดงข้อมูลค่าเฉลี่ยเงินเดือนของอาจารย์ทุกคน

T_id	T_fname	T_lname	fac_id	Salary
2345789015	Bob	Brazzi	Msc	10,142
5678901235	Norbert	Mariah	Ed	14,324
1234567822	Candy	Dodge	Sc	42,442
4567890122	Kendall	Roberto	En	41,135
2345678920	Cristopher	Morales	Msc	12,113
1234567801	Colan	Tess	Sc	13,213
4567890133	Homer	Dodge	En	44,123
2345678910	Wells	Wally	Msc	32,121
5678901240	Joe	Kendall	En	13,214
1234567840	Estrada	Wiliam	Sc	23,142
1234567821	Luke	Pilgrim	Sc	15,321

```
SELECT AVG(salary) AS Average_of_Salary  
FROM Instructor
```

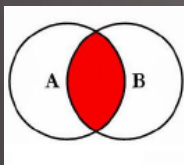
—ผลลัพธ์—



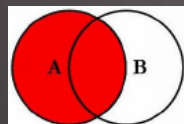
Average_of_Salary
23,753.64

(การเชื่อมโยงโดยใช้คำสั่ง JOIN)

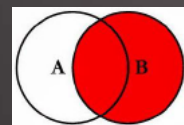
คำสั่งการ JOIN



INNER JOIN : ส่งค่าแถวกลับเมื่อมีอย่างน้อยหนึ่งแถวที่ตรงกับตารางทั้งสอง



LEFT JOIN : ส่งค่ากลับทุกแถวของตารางทางด้านซ้ายของตัวดำเนินการ JOIN แม้ว่าจะไม่ตรงกับค่าในตารางทางด้านขวาของตัวดำเนินการ



RIGHT JOIN : ส่งค่ากลับทุกแถวของตารางทางด้านขวาของตัวดำเนินการ JOIN แม้ว่าจะไม่ตรงกับค่าในตารางทางด้านซ้ายของตัวดำเนินการ

การเชื่อมโยงโดยใช้ตัวดำเนินการ JOIN (INNER JOIN)

ตาราง Categories

cate_id	cate_name
1	เครื่องใช้ไฟฟ้า
2	เครื่องใช้สำนักงาน
3	ยานพาหนะ
5	เครื่องมือสื่อสาร

```
SELECT categories.cate_name, products.prod_id, products.prod_name  
FROM categories INNER JOIN products ON categories.cate_id = products.cate_id
```

ตาราง Products

prod_id	prod_name	cate_id
10	รถยนต์	3
11	รถบรรทุก	3
12	โต๊ะ	2
15	คอมพิวเตอร์	1
16	เก้าอี้	2
20	ถ่าน AAA	4

--ผลลัพธ์--

cate_name	prod_name	cate_id
เครื่องใช้ไฟฟ้า	15	คอมพิวเตอร์
เครื่องใช้สำนักงาน	12	โต๊ะ
เครื่องใช้สำนักงาน	16	เก้าอี้
ยานพาหนะ	10	รถยนต์
ยานพาหนะ	11	รถบรรทุก

การเชื่อมโยงโดยใช้ตัวดำเนินการ JOIN (LEFT JOIN)

ตาราง Categories	
cate_id	cate_name
1	เครื่องใช้ไฟฟ้า
2	เครื่องใช้สำนักงาน
3	ยานพาหนะ
5	เครื่องมือสื่อสาร

```
SELECT categories.cate_name, products.prod_id, products.prod_name  
FROM categories LEFT JOIN products ON categories.cate_id = products.cate_id
```

ตาราง Products		
prod_id	prod_name	cate_id
10	รถยนต์	3
11	รถบรรทุก	3
12	โต๊ะ	2
15	คอมพิวเตอร์	1
16	เก้าอี้	2
20	ร้าน AAA	4

--ผลลัพธ์--

cate_name	prod_name	cate_id
เครื่องใช้ไฟฟ้า	15	คอมพิวเตอร์
เครื่องใช้สำนักงาน	12	โต๊ะ
เครื่องใช้สำนักงาน	16	เก้าอี้
ยานพาหนะ	10	รถยนต์
ยานพาหนะ	11	รถบรรทุก
เครื่องมือสื่อสาร	5	ของใช้

การเชื่อมโยงโดยใช้ตัวดำเนินการ JOIN (RIGHT JOIN)

ตาราง Categories	
cate_id	cate_name
1	เครื่องใช้ไฟฟ้า
2	เครื่องใช้สำนักงาน
3	ยานพาหนะ
5	เครื่องมือสื่อสาร

```
SELECT categories.cate_name, products.prod_id, products.prod_name  
FROM categories RIGHT JOIN products ON categories.cate_id = products.cate_id
```

ตาราง Products		
prod_id	prod_name	cate_id
10	รถยนต์	3
11	รถบรรทุก	3
12	โต๊ะ	2
15	คอมพิวเตอร์	1
16	เก้าอี้	2
20	ร้าน AAA	4

--ผลลัพธ์--

cate_name	prod_name	cate_id
เครื่องใช้ไฟฟ้า	15	คอมพิวเตอร์
เครื่องใช้สำนักงาน	12	โต๊ะ
เครื่องใช้สำนักงาน	16	เก้าอี้
ยานพาหนะ	10	รถยนต์
ยานพาหนะ	11	รถบรรทุก
	20	ร้าน AAA



แหล่งเรียนรู้เพิ่มเติม

<https://www.thaicreate.com/tutorial/sql.html>

