



มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University





มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

บทที่ 5 ฐานข้อมูล Mysql และ ภาษา SQL

รายวิชา 3603610 การพัฒนาเว็บแอปพลิเคชันสำหรับสตาร์ทอัพ

ผศ.ดร.เดช รสรมศิริ



MySQL คืออะไร

- MySQL เป็นระบบจัดการฐานข้อมูลระบบสัมพันธ์ (RDBMS) แบบ Open-Source ที่ได้รับความนิยม มีความสามารถในการจัดการข้อมูลได้อย่างมีประสิทธิภาพ โดยใช้ภาษา SQL ในการจัดการกับฐานข้อมูลนั้นๆ
- การใช้งาน MySQL ในการจัดเก็บข้อมูลในเนื้อหา นี้ จะใช้การจัดการผ่าน phpmyadmin



ข้อดีของฐานข้อมูล MySQL

ฐานข้อมูล MySQL มีข้อดีหลายประการ ทำให้เป็นตัวเลือกยอดนิยมในหมู่ผู้พัฒนาและองค์กรต่างๆ ต่อไปนี้คือประโยชน์หลักบางประการของการใช้ MySQL:

1. โอเพ่นซอร์สและคุ่มต้นทุน

MySQL เป็น ระบบจัดการฐานข้อมูลโอเพ่นซอร์สซึ่งหมายความว่าสามารถดาวน์โหลด ใช้งาน และแจกจ่ายได้ฟรี ซึ่งทำให้ MySQL เป็นตัวเลือกที่มีประสิทธิภาพด้านต้นทุนสำหรับองค์กร เนื่องจากสามารถหลีกเลี่ยงค่าธรรมเนียมใบอนุญาตที่เกี่ยวข้องกับระบบฐานข้อมูลที่เป็นกรรมสิทธิ์ได้

2. ความสามารถในการปรับขนาดและประสิทธิภาพ

MySQL ได้รับการออกแบบมาให้ปรับขนาดได้อย่างมีประสิทธิภาพเพื่อรองรับปริมาณงานที่เพิ่มขึ้นและชุดข้อมูลขนาดใหญ่ รองรับฟีเจอร์ต่างๆ เช่น การจำลอง การจัดกลุ่ม และการแบ่งพาร์ติชัน ช่วยให้องค์กรปรับขนาดฐานข้อมูลในแนวนอนและแนวตั้งเพื่อตอบสนองความต้องการด้านประสิทธิภาพ

3. ความปลอดภัยสูง

MySQL มอบความปลอดภัยที่ดีขึ้นด้วยการใช้เลเยอร์การรักษาความปลอดภัยข้อมูลที่แข็งแกร่งเพื่อจัดเก็บข้อมูลที่ละเอียดอ่อน ดังนั้นข้อมูล เช่น รหัสผ่าน รายละเอียดบัตรเครดิต และข้อมูลส่วนตัวอื่นๆ จะได้รับการเข้ารหัสและปลอดภัยจากการแฮ็กหรือการเข้าถึงจากภายนอก



ข้อดีของฐานข้อมูล MySQL

4. ความสะดวกในการใช้งาน

เนื่องจาก MySQL ได้รับการยอมรับอย่างกว้างขวาง จึงสามารถนำไปใช้งานได้อย่างง่ายดายในแอปพลิเคชันต่างๆ และกับแพลตฟอร์มหรือระบบใดๆ ความเรียบง่ายทำให้ผู้พัฒนาและผู้ดูแลระบบทุกระดับทักษะเข้าถึงได้ จึงช่วยลดขั้นตอนการเรียนรู้ที่เกี่ยวข้องกับการดูแลฐานข้อมูล

5. ความเข้ากันได้ข้ามแพลตฟอร์ม

นอกจากจะได้รับความนิยมอย่างกว้างขวางแล้ว MySQL ยังทำงานร่วมกับระบบปฏิบัติการและแพลตฟอร์มต่างๆ ได้หลากหลาย ดังนั้น คุณจึงสามารถใช้ MySQL กับฮาร์ดแวร์และซอฟต์แวร์ต่างๆ ได้โดยไม่มีปัญหาเรื่องความเข้ากันได้ และทำงานได้อย่างมีประสิทธิภาพบนระบบต่างๆ เช่น Windows, Linux, macOS และ Unix

6. การสนับสนุนและทรัพยากรชุมชนที่แข็งแกร่ง

MySQL มีชุมชนนักพัฒนา ผู้ใช้ และผู้สนับสนุนที่กระตือรือร้นและกระตือรือร้น ซึ่งให้การสนับสนุน เอกสารประกอบ บทช่วยสอน และทรัพยากร องค์กรต่างๆ สามารถใช้ประโยชน์จากระบบนิเวศที่ขับเคลื่อนโดยชุมชนนี้เพื่อแก้ไขปัญหา เรียนรู้แนวทางปฏิบัติที่ดีที่สุด และคอยติดตามการพัฒนาล่าสุดใน MySQL

การจัดการฐานข้อมูล Mysql ผ่าน phpMyAdmin

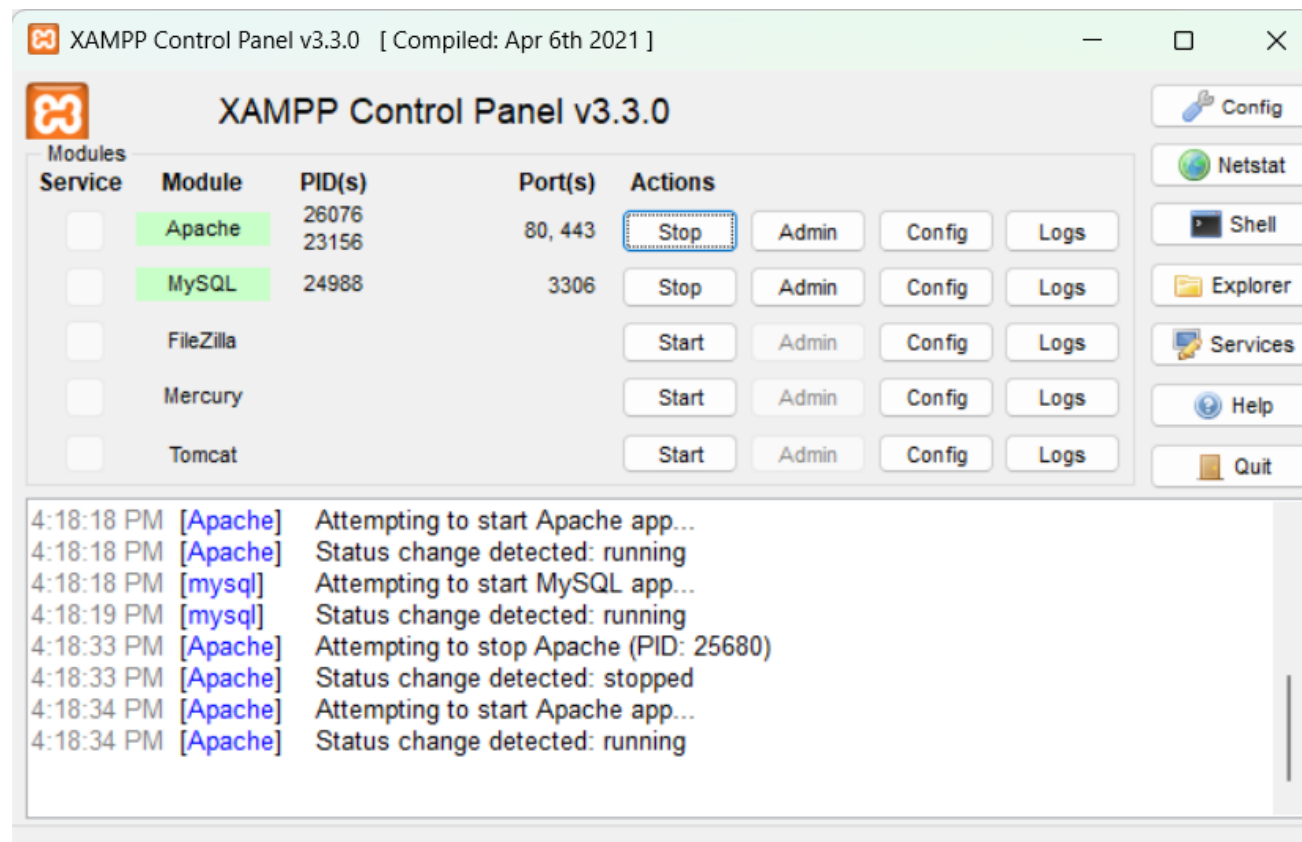
phpMyAdmin คือโปรแกรมที่ถูกพัฒนาโดยใช้ภาษา PHP เพื่อใช้ในการบริหารจัดการฐานข้อมูล Mysql แทนการคีย์คำสั่ง เนื่องจากถ้าเราจะใช้ฐานข้อมูลที่เป็น MySQL บางครั้งจะมีความลำบากและยุ่งยากในการใช้งาน ดังนั้นจึงมีเครื่องมือในการจัดการฐานข้อมูล MySQL ขึ้นมาเพื่อให้สามารถจัดการ ตัวDBMS ที่เป็น MySQL ได้ง่ายและสะดวกยิ่งขึ้น โดย phpMyAdmin ก็ถือเป็นเครื่องมือชนิดหนึ่งในการจัดการ

phpMyAdmin เป็นโปรแกรมประเภท MySQL Client ตัวหนึ่งที่ใช้ในการจัดการข้อมูล MySQL ผ่านweb browser ได้โดยตรง phpMyAdmin ตัวนี้จะทำงานบน Web server เป็น PHP Application ที่ใช้ควบคุมจัดการ MySQL Server ความสามารถของ phpMyAdmin คือ

1. สร้างและลบ Database
2. สร้างและจัดการ Table เช่น แทรก record, ลบ record, แก้ไข record, ลบ Table, แก้ไข field
3. โหลดเท็กซไฟล์เข้าไปเก็บเป็นข้อมูลในตารางได้
4. หาผลสรุป (Query) ด้วยคำสั่ง SQL

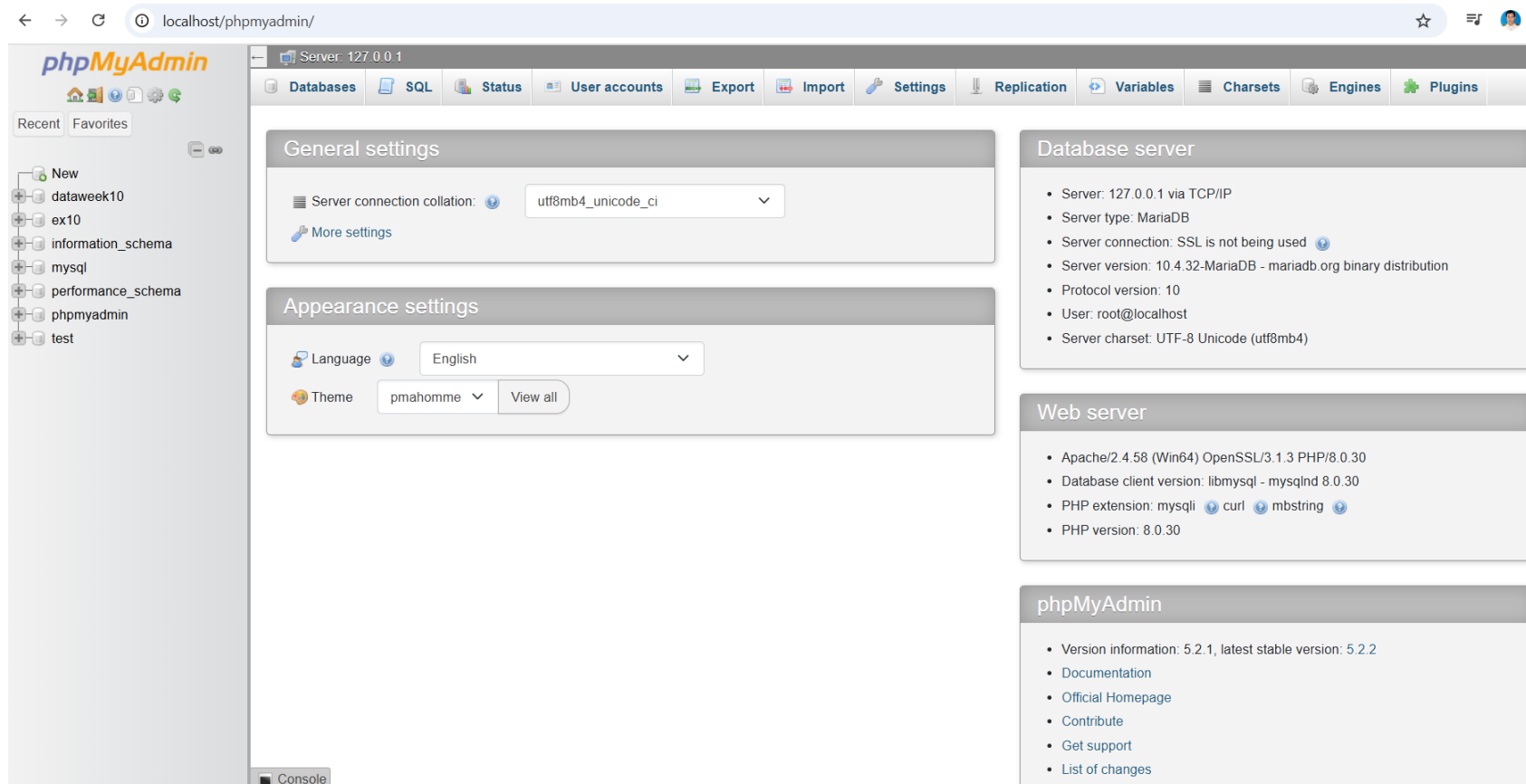
การใช้งาน phpMyAdmin

- ต้องทำการ start apache และ Mysql ใน Xampp ก่อน



การใช้งาน phpMyAdmin

- จากนั้นเปิด browser เข้าผ่าน link `http://localhost/phpmyadmin/`





SQL คืออะไร?

- SQL (Structured Query Language) คือ ภาษาโปรแกรมมิ่งที่ใช้ติดต่อสื่อสารและจัดการกับข้อมูลแบบเชิงสัมพันธ์ (Relational Database) สำหรับ DBMS (Database Management System) ที่เรารู้จักกันดี เช่น MySQL, PostgreSQL, Oracle, MS Server SQL, SQLite เป็นต้น ถึงแม้ว่าแต่ละ DBMS จะมีโครงสร้างหรือฟีเจอร์แตกต่างกันบางส่วน แต่ไวยากรณ์โดยทั่วไปแล้วจะมีความคล้ายคลึงกันครับ ดังนั้นเรียนในบทเรียนนี้ก็ถือว่าไปต่อยอดได้สบายไร้ปัญหา

คำสั่ง Sql เบื้องต้น

- ภาษา SQL ถูกแบ่งออกมาเป็นส่วนประกอบต่างๆ เราเรียกส่วนเหล่านี้ตามรูปแบบเช่น
- Clauses คือองค์ประกอบหนึ่งของ statement และ query (ส่วนนี้เป็น Optional)
- Expressions คือการสร้างผลลัพธ์ออกมาในรูป table ที่ประกอบด้วย column และ row จากข้อมูล
- Predicates คือรูปแบบเงื่อนไขที่มีผลลัพธ์เป็น true/false/unknown หรือก็คือ Boolean นั่นเอง
- Queries คือการดึงข้อมูลตามเงื่อนไข (clause) เป็นส่วนสำคัญใน SQL
- Statements คือสามารถมีผลต่อโครงสร้างข้อมูล, จัดการข้อมูล transactions, program flow, session หรือแม้กระทั่ง วิเคราะห์ปัญหา โดยจำเป็นต้องจบด้วย semicolon (;) ซึ่งเป็นสิ่งจำเป็นที่ต้องมีทุกครั้งสำหรับ SQL

```
UPDATE clause [UPDATE country
SET clause   [SET population = population + 1
WHERE clause [WHERE name = 'USA';
```

Diagram illustrating the structure of an SQL statement:

- The **UPDATE clause** is shown as `[UPDATE country`.
- The **SET clause** is shown as `[SET population = population + 1`. The expression `population + 1` is labeled as an **Expression**.
- The **WHERE clause** is shown as `[WHERE name = 'USA';`. The expression `name = 'USA'` is labeled as an **Expression**, and the entire `WHERE name = 'USA'` is labeled as a **Predicate**.
- A large bracket on the right groups the three clauses under the label **Statement**.



คำสั่ง SQL พื้นฐาน

Database SQL- Create database

รูปแบบ (Syntax)

SQL Create database statement โดยชื่อ database ในระบบจะต้องไม่ซ้ำกัน
ภายใน RDBMS ที่ใช้งาน

```
CREATE DATABASE <DatabaseName>;
```

ตัวอย่าง

```
CREATE DATABASE ex1;
```

ประเภทข้อมูลใน MySQL ข้อมูลประเภท Integer ใช้เก็บตัวเลขประเภทจำนวนเต็ม

ประเภท	ความหมาย
TINYINT	มีขนาด 1 byte เก็บค่าได้ 2^8 จำนวน
SMALLINT	มีขนาด 2 byte เก็บค่าได้ 2^{16} จำนวน
MEDIUMINT	มีขนาด 3 byte เก็บค่าได้ 2^{24} จำนวน
INT	มีขนาด 4 byte เก็บค่าได้ 2^{32} จำนวน
BIGINT	มีขนาด 8 byte เก็บค่าได้ 2^{64} จำนวน

ประเภทที่ 2 คือ Float และ Double ใช้เก็บตัวเลขที่เป็นทศนิยม

ประเภท	ความหมาย
FLOAT	จำนวนทศนิยม ระหว่าง 1.18×10^{-38} – 3.40×10^{38}
DOUBLE	จำนวนทศนิยม ระหว่าง 2.23×10^{-308} – 1.80×10^{308}

ประเภทที่ 3 คือ String และ Text ใช้สำหรับเก็บข้อความและตัวอักษร

ประเภท	ความหมาย
CHAR	ใช้เก็บข้อความแบบกำหนดขนาดคงที่ มีขนาด 255 ตัวอักษร
VARCHAR	ใช้เก็บข้อความแบบกำหนดขนาดได้ มีขนาดสูงสุด 255 ตัวอักษร
TINYTEXT	ใช้เก็บตัวอักษรได้ 255 ตัวอักษร
TEXT	ใช้เก็บตัวอักษรได้มากถึง 65,535 ตัวอักษร
MEDIUMTEXT	ใช้เก็บตัวอักษรได้มากถึง 16,777,215 ตัวอักษร
LONGTEXT	ใช้เก็บตัวอักษรได้มากถึง 4,294,967,295 ตัวอักษร
BLOB	ใช้เก็บข้อมูลที่มีลักษณะเป็น ไฟล์รูปภาพ ไฟล์เสียง ใช้พื้นที่ 64 KB
TINYBLOB	ใช้เก็บข้อมูลที่มีลักษณะเป็น ไฟล์รูปภาพ ไฟล์เสียง ใช้พื้นที่ 1 byte
MEDIUMBLOB	ใช้เก็บข้อมูลที่มีลักษณะเป็น ไฟล์รูปภาพ ไฟล์เสียง ใช้พื้นที่ 16 MB
LOB	ใช้เก็บข้อมูลที่มีลักษณะเป็น ไฟล์รูปภาพ ไฟล์เสียง ใช้พื้นที่ 4 GB
ENUM	ใช้เก็บข้อมูลที่เป็นตัวเลือก เช่น เก็บเพศชายก็เก็บเป็น 'm' เก็บเพศหญิงก็เก็บเป็น 'f'
SET	ใช้เก็บข้อมูลที่เป็นประเภท CheckBox ซึ่งเก็บได้หลายค่า เช่น รายการผลไม้ที่ชอบ มะม่วง ทุเรียน เงาะ ซึ่งลูกค้าได้เลือกหลายรายการ ที่ส่งมาจากหน้าฟอร์ม

ประเภทที่ 4 คือ ประเภท วัน เดือน ปี ต่างๆ

ประเภท	ความหมาย
DATE	ใช้สำหรับเก็บวันที่ในรูปแบบ YYYY-MM-DD (ปี-เดือน-วัน)
TIME	ใช้สำหรับเก็บเวลาในรูปแบบ HH:MM:SS (ชั่วโมง:นาที:วินาที)
DATETIME	ใช้สำหรับเก็บวันที่และเวลาในรูปแบบ YYYY-MM-DD HH:MM:SS
YEAR	ใช้สำหรับเก็บปี ซึ่งจะเก็บ 4 หลัก คือ 2019 หรือ เก็บ 2 หลัก คือ 19
TIMESTAMP	ใช้สำหรับเก็บวันและเวลา ที่มีการเพิ่มข้อมูล กรณีไม่ถ้ามีการเพิ่มข้อมูลเข้ามาจะเป็นค่า null
TIMESTAMP(14)	หมายถึงเก็บในรูปแบบ YYYYMMDDHHMMSS
TIMESTAMP(12)	หมายถึงเก็บในรูปแบบ YYMMDDHHMMSS
TIMESTAMP(10)	หมายถึงเก็บในรูปแบบ YYMMDDHHMM
TIMESTAMP(8)	หมายถึงเก็บในรูปแบบ YYYYMMDD
TIMESTAMP(6)	หมายถึงเก็บในรูปแบบ YYMMDD
TIMESTAMP(6)	หมายถึงเก็บในรูปแบบ YYMM

Database SQL – Create table

รูปแบบ (Syntax)

SQL Create table statement ข้อมูลที่ต้องระบุคือ table name, column name, data type

```
CREATE TABLE table_name(  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    .....  
    columnN datatype,  
    PRIMARY KEY( one or more columns )  
);
```

คำสั่ง CREATE TABLE เป็น keyword เพื่อบอกกับ database ว่าเราต้องการทำอะไร ในที่นี้คือสร้างตารางใหม่ ที่ชื่อต้องไม่ซ้ำกับตารางเดิมที่มีอยู่ในฐานข้อมูล โดยต้องใช้รูปแบบตามตัวอย่างที่กำหนดไว้ หลังจากนั้นในส่วนบของวงเล็บที่ตามมา จะเป็นการกำหนดข้อมูลแต่ละ column ของ table ว่า data type ที่จะใช้คือข้อมูลประเภทใด

ตัวอย่างการสร้าง table

SQL statement block ต่อไปนี้คือ ตัวอย่างการสร้าง table CUSTOMERS ที่ใช้ ID เป็น primary key ซึ่งต้องมีค่าไม่ซ้ำกัน และไม่มีค่าว่าง (NOT NULL)

```
SQL> CREATE TABLE CUSTOMERS(  
    ID INT NOT NULL,  
    NAME VARCHAR (20) NOT NULL,  
    AGE INT NOT NULL,  
    ADDRESS CHAR (25) ,  
    SALARY DECIMAL (18, 2),  
    PRIMARY KEY (ID)  
);
```

ตัวอย่างคำสั่งในเบื้องต้นที่ถูกใช้บ่อยใน SQL

คำสั่ง	คำอธิบาย
CREATE	สร้างตารางขึ้นมาใหม่
ALTER	เปลี่ยนแปลงข้อมูลในตาราง เช่น เปลี่ยนชื่อฟิลด์ (คอลัมน์) เป็นต้น
DROP	ลบตาราง
SELECT	เลือกดึงข้อมูลจากตารางที่ต้องการ
INSERT	บันทึกข้อมูลเข้าไปใน record (แถว) ตาราง
UPDATE	แก้ไขหรืออัปเดตข้อมูลใน record
DELETE	ลบข้อมูลใน record

คำสั่ง Select

- รูปแบบ (Syntax) SQL Select statement จำเป็นต้องใช้ข้อมูล column name และ table name ในการระบุชุดของข้อมูลที่จะแสดงเช่น
 - * แบบที่ 1 การ select แบบระบุคอลัมน์
 - SELECT column1, column2, columnN
 - FROM table_name;
 - Select คอลัมน์ที่ 1,คอลัมน์ที่ 2 ,และคอลัมน์ต่อ ๆ ไป อย่าลืมคั่นคอลัมน์ด้วย , เพื่อแบ่งแต่ละคอลัมน์
 - From ชื่อตาราง
-
- เช่น SELECT std_id,std_name,std_surname FROM student;

คำสั่ง Select แบบที่ 2

แบบที่ 2 การ select แบบไม่ระบุคอลัมน์แต่เลือกมาทั้งหมด ทุกคอลัมน์ในตาราง ใช้ * แทนชื่อ คอลัมน์ทั้งหมด

```
SELECT * FROM table_name
```

ตัวอย่างคำสั่ง

```
Select * from student
```



รายละเอียดของคำสั่ง Select

ส่วนคำสั่ง SQL	หน้าที่	จำเป็น ไหม
SELECT	แสดงเขตข้อมูลที่มีข้อมูลที่สนใจ	ใช่
FROM	แสดงตารางที่มีเขตข้อมูลปรากฏอยู่ในส่วนคำสั่ง SELECT	ใช่
WHERE	ระบุเงื่อนไขเขตข้อมูลที่ใช้เลือก ระเบียนที่จะรวมอยู่ในผลลัพธ์	ไม่ใช่
ORDER BY	ระบุวิธีเรียงลำดับผลลัพธ์	ไม่ใช่
GROUP BY	ในคำสั่ง SQL ที่มีฟังก์ชันการรวม จะแสดงเขตข้อมูลที่ไม่ถูกรวบรวมอยู่ในส่วนคำสั่ง SELECT	เฉพาะเมื่อมีเขตข้อมูลเหล่านี้เท่านั้น
HAVING	ในคำสั่ง SQL ที่มีฟังก์ชันการรวม จะระบุเงื่อนไขซึ่งใช้กับเขตข้อมูลที่ถูกสรุปรวมอยู่ในคำสั่ง SELECT	ไม่ใช่

คำสั่ง SQL SELECT WHERE

- คำสั่ง SELECT สามารถมี WHERE ข้อจำกัดเพิ่มเติมได้ ข้อจำกัด WHERE ดังกล่าวช่วยให้เราสามารถดึงข้อมูลมาจากตารางฐานข้อมูลที่ตรงกับเงื่อนไขที่ระบุ ตัวอย่างเช่น

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT *
FROM Customers
WHERE last_name = 'Doe';

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
5	Betty	Doe	28	UAE



คำสั่ง SQL Insert

Insert คือ คำสั่งที่ใช้สำหรับ "เพิ่ม" ข้อมูลในแถวใหม่ของตาราง นี่คือตัวอย่างไวยากรณ์ของคำสั่ง โดยจะใช้คีย์เวิร์ด INSERT

```
INSERT INTO table_name (column1, column2)
VALUES (value1, value2);
```

ตัวอย่างคำสั่งการ insert ใน SQL

```
INSERT INTO Students (name, age)
VALUES ('Dech', 47);
```

แล้วถ้าอยาก insert ข้อมูลครั้งเดียวได้หลาย ๆ แถวล่ะ?

```
INSERT INTO Students (name, age)
VALUES ('Dech', 7), ('Supoj', 49), ('Phayung', 50);
```

โดยการ insert ข้อมูลครั้งเดียวได้หลายแถวแบบนี้ เราจะใช้ comma "," คั่นระหว่างชุดข้อมูลที่อยู่ในวงเล็บ (Parenthesis)

คำสั่ง SQL Update

Update คือ คำสั่งที่ใช้สำหรับ "แก้ไข" ข้อมูลในแถว (record) ของตาราง นี่คือตัวอย่างไวยากรณ์ของคำสั่ง โดยจะใช้คีย์เวิร์ด UPDATE

```
UPDATE table_name
```

```
SET column1 = value1, column2 = value2
```

```
WHERE condition;
```

จากตัวอย่างด้านบน ต่อไปจะเป็นการทดสอบการ Update ข้อมูลใน SQL

```
UPDATE Students
```

```
SET name = "Dech", age = 40
```

```
WHERE id = 1 ;
```

คำสั่ง SQL Delete

Delete คือ คำสั่งที่ใช้สำหรับ "ลบ" ข้อมูลในแถว (record) ของตาราง นี่คือตัวอย่างไวยากรณ์ของคำสั่ง โดยจะใช้คีย์เวิร์ด DELETE

```
DELETE FROM table_name  
WHERE condition;
```

ตัวอย่างการใช้คำสั่ง Delete ใน SQL

```
DELETE FROM Students  
WHERE id = 2;
```



ทดลองปฏิบัติกัน



มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

ขอบคุณครับ

