

PHP เชื่อมต่อฐานข้อมูล



342 236 Web Application Programming

Monlica Wattana, Ph.D

Department of Computer Science, Khon Kaen University

Outline



- คำสั่ง SQL: Select
- ระบบฐานข้อมูล MySQL
- ฟังก์ชัน PHP สำหรับการเชื่อมต่อฐานข้อมูล

คำสั่ง SQL



- คำสั่ง SQL แบ่งได้ 3 รูปแบบ
 - ภาษาสำหรับการนิยามข้อมูล (Data Definition Language : DDL)
 - ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language : DML)
 - ภาษาควบคุม (Data Control Language : DCL)

คำสั่ง SELECT



- คำสั่ง select เป็นคำสั่งในการเลือกข้อมูลจากฐานข้อมูลประกอบด้วย
 - Select
 - From
 - Where (optional)
 - Order By (optional)
 - Group By (optional)

คำสั่ง SELECT



คำสั่ง SELECT ใช้ดึงข้อมูลจาก database

SELECT **column_name1, column_name2, ...**

FROM **table_name**;

* คือดึงข้อมูลจากทุก fields ในตาราง **SELECT * FROM Student;**

Student Table

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12

คำสั่ง SELECT



SELECT **column_name,column_name**

FROM **table_name**;

SELECT **std_name, std_gpa** FROM **Student**;

std_name	std_gpa
Sammy	3.00
Colly	3.20
Tony	3.42
Simon	3.12

WHERE - Optional



- ใช้ในการ join ตาราง และได้ข้อมูลกลับมาที่เฉพาะเจาะจง

SQL Operators ของ WHERE

- Logical Operators

AND , OR

- Comparison Operators

=

Equality

!= or <>

Inequality

> , >=

Greater than , Greater than or equal

< , <=

Less than , Less than or equal

Like

String search

In

List of values

คำสั่ง SELECT - Equals



```
SELECT *  
FROM Student  
WHERE std_dept = "Computer";
```

Student Table

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12



Result

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20

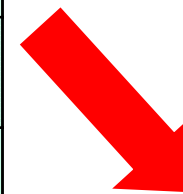
คำสั่ง SELECT – not Equals



```
SELECT *  
FROM Student  
WHERE std_dept <> "Computer";
```

Student Table

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12



Result

std_id	std_name	std_dept	std_gpa
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12

คำสั่ง SELECT – Greater than or equal



```
SELECT *  
FROM Student  
WHERE std_gpa >= 3.20;
```

Student Table

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
56033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12



Result

std_id	std_name	std_dept	std_gpa
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42

คำสั่ง SELECT – LIKE



```
SELECT *  
FROM Student  
WHERE std_name LIKE 'S%';
```

Student Table

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12



Result

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59044444-4	Simon	Math	3.12

คำสั่ง SELECT – IN



```
SELECT *  
FROM Student  
WHERE std_dept IN ('Math', 'Physics');
```

Student Table

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12



Result

std_id	std_name	std_dept	std_gpa
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12

คำสั่ง SELECT – Multiple Conditions Using AND



```
SELECT *  
FROM Student  
WHERE std_dept = "Computer" AND std_gpa > 3.15;
```

Student Table

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12



Result

std_id	std_name	std_dept	std_gpa
59022222-2	Colly	Computer	3.20

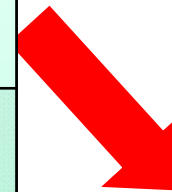
คำสั่ง SELECT – Multiple Conditions Using OR



```
SELECT *  
FROM Student  
WHERE std_dept = "Computer" OR std_gpa > 3.15;
```

Student Table

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12



Result

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42

ระบบฐานข้อมูล MySQL



- มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูลที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่นอย่างบูรณาการ
- เป็นระบบงานที่รองรับ ความต้องการของผู้ใช้ เช่น ทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) เช่น ภาษา php ภาษา asp.net หรือภาษาเจเอสพี เป็นต้น หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) เช่น ภาษาวิซวลเบสิกคอตเน็ต ภาษาจาวา หรือภาษาซีชาร์ป เป็นต้น
- โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลาย



ระบบฐานข้อมูล MySQL



- เป็นระบบฐานข้อมูลโอเพนทซอร์ท (Open Source) ที่ถูกนำไปใช้งานมากที่สุด
- ปัจจุบันบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems, Inc.) เข้าซื้อกิจการของ MySQL AB ดังนั้นผลิตภัณฑ์ภายใต้ MySQL AB ทั้งหมดจะตกเป็นของ บริษัทซันไมโครซิสเต็มส์

ความสามารถและการทำงานของโปรแกรม MySQL



- MySQL ถือเป็นระบบจัดการฐานข้อมูล (DataBase Management System (DBMS)) ฐานข้อมูลมีลักษณะเป็นโครงสร้างของการเก็บรวบรวมข้อมูล การที่จะเพิ่ม เข้าถึง หรือประมวลผลข้อมูลที่เก็บในฐานข้อมูลจำเป็นจะต้องอาศัยระบบจัดการ ฐานข้อมูล ซึ่งจะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูล MySQL ทำหน้าที่เป็นทั้งตัวฐานข้อมูลและระบบจัดการฐานข้อมูล
- MySQL เป็นระบบจัดการฐานข้อมูลแบบ Relational ฐานข้อมูลแบบ relational จะทำการเก็บข้อมูลทั้งหมดในรูปแบบของตารางแทนการเก็บข้อมูลทั้งหมดลงในไฟล์ เพียงไฟล์เดียว ทำให้ทำงานได้รวดเร็วและมีความยืดหยุ่น โดยอาศัยภาษา SQL ที่เป็นส่วนหนึ่งของโปรแกรม MySQL ซึ่งเป็นภาษามาตรฐานในการเข้าถึงฐานข้อมูล

ความสามารถและการทำงานของโปรแกรม MySQL



- MySQL แจกจ่ายให้ใช้งานแบบ Open Source นั่นคือ ผู้ใช้งาน MySQL ทุกคนสามารถใช้งานและปรับแต่งการทำงานได้ตามต้องการ สามารถดาวน์โหลดโปรแกรม MySQL ได้จากอินเทอร์เน็ตและนำมาใช้งานโดยไม่มีค่าใช้จ่ายใดๆ

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



ฟังก์ชัน `mysqli_connect()`: การเชื่อมต่อกับฐานข้อมูล

- ในการติดต่อกับฐานข้อมูลจะต้องทำการเปิดการติดต่อ Database Server ก่อน โดยมีรูปแบบการใช้งานดังนี้

```
mysqli_connect(hostname, username, password, database name);
```

- `hostname` คือ ชื่อของ Database Server ในการที่ติดตั้ง MySQL ไว้ในเครื่องเดียวกับเว็บเซิร์ฟเวอร์ ก็สามารถระบุเป็น `localhost` แทนชื่อจริงได้เลย
- `username` คือ ชื่อผู้ใช้ที่ถูกกำหนดให้สามารถทำงานกับ MySQL ได้
- `password` คือ รหัสผ่านของผู้ใช้
- `database name` คือ ชื่อของฐานข้อมูล

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



- ค่าที่คืนออกมาจากการเรียกใช้ฟังก์ชันนี้จะได้ค่าเป็นจริง หากสามารถติดต่อกับ MySQL ได้สำเร็จ แต่ถ้าไม่สามารถติดต่อได้หรือติดต่อไม่สำเร็จจะมีค่าเป็นเท็จ ตัวอย่าง

```
<?php
```

```
$host = "localhost";
```

```
$user = "root";
```

```
$password = "";
```

```
$database = "mydatabase";
```

```
$conn = mysqli_connect($host, $user, $password, $database);
```

```
if(!$conn) { echo "เกิดข้อผิดพลาดไม่สามารถเชื่อมต่อฐานข้อมูลได้";
```

```
    exit;
```

```
    }else{ echo "สามารถเชื่อมต่อฐานข้อมูลได้ ...";    }
```

```
?>
```

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



ฟังก์ชัน `mysqli_close` : การยกเลิกการเชื่อมต่อ

- ฟังก์ชันที่ใช้ในการยกเลิกหรือปิดการติดต่อ Database Server

```
mysqli_close(database_connect);
```

- โดยผลลัพธ์ที่คืนออกมาจากฟังก์ชันนี้ ถ้าปิดการติดต่อกับ MySQL ได้สำเร็จก็จะมีค่าเป็นจริง ถ้าไม่สำเร็จจะมีค่าเป็นเท็จ
ตัวอย่าง

```
mysqli_close($conn);
```

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



การนำภาษา SQL มาใช้ในฐานข้อมูล MySQL ดังนี้

ฟังก์ชัน `mysqli_query()`

- เป็นฟังก์ชันสำหรับสั่งงาน MySQL ด้วยภาษา SQL เพื่อจัดการกับข้อมูลในฐานข้อมูล เช่น การเพิ่ม การลบ เป็นต้น ต้องใช้กับฟังก์ชัน `mysql_select_db()`

```
mysqli_query(database_connect, string query);
```

- query หมายถึง คำสั่ง SQL ที่เรียกดึงข้อมูลใช้ฐานข้อมูล
 - database_connect หมายถึง ตัวแปรที่ใช้เชื่อมต่อกับฐานข้อมูล
- ตัวอย่าง

```
$sql = "SELECT * FROM student";  
$result = mysqli_query($conn,$sql);
```

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



ฟังก์ชัน `mysqli_free_result()`

- เป็นฟังก์ชันสำหรับคืนหน่วยความจำให้กับระบบ เพื่อใช้หน่วยความจำให้มีประสิทธิภาพมากที่สุด ถ้ามีการใช้ตัวแปรมากๆ แล้วไม่มีการคืนหน่วยความจำจะส่งผลให้หน่วยความจำเต็มและมีผลต่อการทำงานของระบบได้

```
mysqli_free_result(result);
```

- `result` หมายถึง ค่าที่ได้จากการใช้คำสั่งคิวรี
ตัวอย่าง

```
mysqli_free_result($result);
```

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



ฟังก์ชัน `mysqli_fetch_array()`

- เป็นฟังก์ชันที่ใช้สำหรับดึงค่าผลลัพธ์ของฐานข้อมูลเก็บไว้ในอาร์เรย์ ผลลัพธ์ที่คืนออกมาจากฟังก์ชันนี้ จะเป็นข้อมูลอาร์เรย์ที่มีสมาชิกเท่ากับจำนวนคอลัมน์ของตาราง

```
mysqli_fetch_array(result);
```

- `result` คือค่าที่ได้จากฟังก์ชัน `mysqli_query()`
จากการใช้ฟังก์ชันนี้ จะเป็นการอ่านค่าและส่งค่าลงตัวแปรอาร์เรย์ทีละ 1 รายการ
ถ้าต้องการแสดงค่าของข้อมูลครบทุกรายการที่มีในตารางผลลัพธ์ ก็จะต้องกำหนดคำสั่งให้วนรอบการทำงานของฟังก์ชัน ตัวอย่าง

```
while ($rs = mysqli_fetch_array($result)) {  
    echo $rs["std_id"] . " | ";  
    echo $rs["std_name"] . "<br />";  
}
```

ผลลัพธ์

59011111-1	Sammy
59022222-2	Colly
59033333-3	Tony
59044444-4	Simon

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



ฟังก์ชัน `mysqli_fetch_row()`

- เป็นฟังก์ชันที่ใช้สำหรับเลื่อนตำแหน่งของตัวชี้ข้อมูลไปยังเรคอร์ดถัดไป

```
mysqli_fetch_row(result);
```

- `result` คือค่าที่ได้จากฟังก์ชัน `mysqli_query()`
ตัวอย่าง

```
$row = mysqli_fetch_row($result);  
    echo $row[0];  
    echo $row[1]."<br>";  
  
$row = mysqli_fetch_row($result);  
    echo $row[0];  
    echo $row[1]."<br>";
```

ผลลัพธ์

```
59011111-1Sammy  
59022222-2Colly
```

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



ฟังก์ชัน `mysqli_num_fields()`

- เป็นฟังก์ชันที่ใช้ในการหาจำนวนคอลัมน์ที่มีทั้งหมด

```
mysqli_num_fields(result);
```

- `result` คือค่าที่ได้จากฟังก์ชัน `mysqli_query()`
ผลลัพธ์ที่คืนออกมาจากฟังก์ชันนี้ เป็นชนิดตัวเลข ได้แก่ จำนวนคอลัมน์ทั้งหมดของตาราง ตัวอย่าง

```
$numfield = mysqli_num_fields($result);  
while($row = mysqli_fetch_array($result))  
{  
    for($i = 0; $i < $numfield; $i++)  
        { echo $row[$i]. " | "; }  
    echo "<br>";  
}
```

ผลลัพธ์

59011111-1		Sammy		Computer		3.00	
59022222-2		Colly		Computer		3.20	
59033333-3		Tony		Physics		3.42	
59044444-4		Simon		Math		3.12	

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



ฟังก์ชัน `mysqli_num_rows()`

- เป็นฟังก์ชันที่ใช้สำหรับคำนวณหาจำนวนแถวหรือจำนวนรายการทั้งหมด

```
mysqli_num_rows(result);
```

- `result` คือค่าที่ได้จากฟังก์ชัน `mysqli_query()`
ผลลัพธ์ที่คืนออกมาจากฟังก์ชันนี้ เป็นข้อมูลชนิดตัวเลข ได้แก่ จำนวนรายการทั้งหมดของตารางผลลัพธ์

```
$num_row= mysqli_num_rows($result);  
echo "ข้อมูลมีจำนวนแถวทั้งหมด : " . $num_row . " แถว";
```

ผลลัพธ์

```
ข้อมูลมีจำนวนแถวทั้งหมด : 4 แถว
```

ฟังก์ชันในการจัดการฐานข้อมูลใน MySQL



ฟังก์ชัน `mysqli_data_seek()`

- เป็นฟังก์ชันที่ใช้สำหรับการเลื่อนพอยน์เตอร์ไปชี้ที่ลำดับที่ระบุ

```
mysqli_data_seek(result, index);
```

- `result` คือค่าที่ได้จากฟังก์ชัน `mysqli_query()`
- `index` คือลำดับแถวใน `result`

ตัวอย่าง

```
mysqli_data_seek($result, 3);  
$row = mysqli_fetch_row($result);  
echo $row[0]. " | ";  
echo $row[1]. "<br>";
```

std_id	std_name	std_dept	std_gpa
59011111-1	Sammy	Computer	3.00
59022222-2	Colly	Computer	3.20
59033333-3	Tony	Physics	3.42
59044444-4	Simon	Math	3.12

ผลลัพธ์

```
59044444-4 | Simon
```

ขั้นตอนการเชื่อมต่อฐานข้อมูล



ประกอบด้วยขั้นตอน 4 ขั้นตอน

1. เริ่มติดต่อกับฐานข้อมูล และ เลือกฐานข้อมูลที่ต้องการ
2. กำหนดคำสั่ง SQL และส่งให้ทำงาน เช่น เก็บข้อมูลลงแปร Array, นับจำนวน Record
3. แสดงผลบราวเซอร์
4. ปิดฐานข้อมูล MySQL

ตัวอย่างการเขียนคำสั่งเชื่อมต่อฐานข้อมูล



```
<?php
```

```
$host = "localhost";
```

```
$user = "root";
```

```
$password = "";
```

```
$database = "mydatabase";
```

```
$conn = mysqli_connect($host, $user, $password, $database);
```

```
if(!$conn){
```

```
    echo "เกิดข้อผิดพลาดไม่สามารถเชื่อมต่อฐานข้อมูลได้";
```

```
    exit;
```

```
    }else{
```

```
        echo "สามารถเชื่อมต่อฐานข้อมูลได้ <br>";
```

```
    }
```

สามารถเชื่อมต่อฐานข้อมูลได้

แสดงผลข้อมูลตาราง student

59011111-1 Sammy เกรดเฉลี่ย: 3.00

59022222-2 Colly เกรดเฉลี่ย: 3.20

59033333-3 Tony เกรดเฉลี่ย: 3.42

59044444-4 Simon เกรดเฉลี่ย: 3.12

ตัวอย่างการเขียนคำสั่งเชื่อมต่อฐานข้อมูล (ต่อ)



```
$sql = "SELECT * FROM student";
```

```
$result = mysqli_query($conn,$sql);
```

```
echo "แสดงผลข้อมูลตาราง student <br>";
```

```
while ($row = mysqli_fetch_row($result)) {
```

```
    echo $row[0]. "  ";
```

```
    echo $row[1]. "  ";
```

```
    echo " เกรดเฉลี่ย: ".$row[3]. "<br>";
```

```
}
```

```
mysqli_close($conn);
```

```
?>
```

สามารถเชื่อมต่อฐานข้อมูลได้
แสดงผลข้อมูลตาราง student

59011111-1	Sammy	เกรดเฉลี่ย: 3.00
59022222-2	Colly	เกรดเฉลี่ย: 3.20
59033333-3	Tony	เกรดเฉลี่ย: 3.42
59044444-4	Simon	เกรดเฉลี่ย: 3.12

ลองเขียน โปรแกรมเพื่อติดต่อฐานข้อมูล



- เขียนโปรแกรมเพื่อติดต่อฐานข้อมูล host ชื่อ myServer มี username คือ root และรหัสผ่านคือ 54321 ฐานข้อมูลชื่อ employeeDB โดยให้ดึงข้อมูลจากตาราง employee เฉพาะคอลัมน์ emp_id, emp_name และ salary ให้แสดงเฉพาะพนักงานที่มีเงินเดือน(salary) น้อยกว่า 40,000 บาท ให้แสดงผลดังภาพที่ 2

ตาราง employee

emp_id	emp_name	position	salary
S001	Joe	Manager	54000
S002	Bob	Programmer	30000
S003	Peter	Manager	54000
S004	Alice	IT Support	25000

ภาพที่ 1

รายชื่อพนักงานที่มีเงินเดือนน้อยกว่า 40,000 บาท

S002 Bob เงินเดือน: 30000

S004 Alice เงินเดือน: 25000

จำนวนพนักงานที่มีเงินเดือนน้อยกว่า 40,000บาท ทั้งหมด : 2 คน

ภาพที่ 2

ลองเขียน โปรแกรมเพื่อติดต่อฐานข้อมูล(ต่อ)



```
<?php
```

```
?>
```

Reference



- บัญชา ปะสีละเตสัง. พัฒนาเว็บแอปพลิเคชันด้วย PHP ร่วมกับ MySQL และ jQuery. พิมพ์ลักษณ์, กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2557
- <http://th.easyhostdomain.com/dedicated-servers/mysql.html>
- <http://pibul2.psu.ac.th/~urairux/download/%E0%A2%D5%C2%B9%E2%BB%C3%E1%A1%C3%C1%20PHP%20%E0%BA%D7%E9%CD%A7%B5%E9%B9.ppt>
- http://www.mnst.go.th/save_file.php?pathatt=file_name&file_name=php.ppt