

บทที่ 14

การใช้ PHP ร่วมกับ MySQL

ในบทนี้จะกล่าวถึงฟังก์ชันของภาษา PHP สำหรับการเชื่อมต่อและประยุกต์ใช้งานร่วมกับ MySQL สำหรับฟังก์ชันที่เกี่ยวข้องนั้นมีจำนวนมาก แต่มีข้อแนะนำ คือ ชื่อฟังก์ชันที่เกี่ยวข้องเหล่านี้จะขึ้นต้นด้วยคำว่า mysql เป็นส่วนใหญ่ ทำให้จำได้ไม่ยาก เน้นประยุกต์การใช้งานเป็นสำคัญเพื่อเป็นพื้นฐานสำหรับการนำไปประยุกต์ใช้สำหรับการพัฒนาระบบที่มีการเชื่อมต่อกับฐานข้อมูล MySQL ต่อไป

14.1 ฟังก์ชันเชื่อมต่อ/ยกเลิกการเชื่อมต่อระหว่างภาษา PHP และระบบฐานข้อมูล MySQL

ก่อนที่จะใช้งานฐานข้อมูลได้ จะต้องเปิดการเชื่อมต่อกับฐานข้อมูลให้ได้ก่อน โดยฟังก์ชันที่ใช้ในการเปิดการเชื่อมต่อกับฐานข้อมูล MySQL ประกอบด้วย 1) `mysql_connect ()` และ 2) `mysql_pconnect ()` แต่เนื่องจากระบบฐานข้อมูล MySQL เป็นระบบฐานข้อมูลเชิงสัมพันธ์ที่มีการตรวจสอบสิทธิ์ในการเข้าใช้งานก่อนการเข้าใช้งานจัดการกับระบบฐานข้อมูล ดังนั้นฟังก์ชันที่ใช้จะต้องระบุข้อมูลที่สำคัญประกอบด้วย ชื่อหรือหมายเลข IP ของเซิร์ฟเวอร์ที่ติดตั้งฐานข้อมูล MySQL หากเป็นเครื่องเซิร์ฟเวอร์เดียวกันกับเว็บเซิร์ฟเวอร์ที่กำลังใช้งานอยู่ในปัจจุบัน สามารถระบุเป็น localhost แทนชื่อเซิร์ฟเวอร์ได้ ตามด้วยชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) เพื่อใช้ตรวจสอบสิทธิ์ก่อนการเข้าใช้งานฐานข้อมูล MySQL หากข้อมูลที่ระบุถูกต้อง ฟังก์ชันจะคืนค่าเป็น MySQL link identifier (Resource) แต่หากไม่ถูกต้องจะคืนค่าเป็นเท็จ (False) สำหรับความแตกต่างของฟังก์ชัน `mysql_connect ()` และ `mysql_pconnect ()` มี 2 ประเด็น ดังนี้

14.1.1 การใช้งานด้วยฟังก์ชัน `mysql_pconnect ()` จะค้นหาว่ามีการเชื่อมต่อกับระบบฐานข้อมูล MySQL เซิร์ฟเวอร์ ด้วยชื่อผู้ใช้ และรหัสผ่าน อยู่ก่อนหน้านี้หรือไม่ หากพบ จะใช้ MySQL link identifier เดิม (เรียกการเชื่อมต่อแบบนี้ว่า "Persistent") แต่หากไม่พบการเชื่อมต่อเดิม จะทำการสร้างการเชื่อมต่อใหม่ แต่สำหรับฟังก์ชัน `mysql_connect ()` จะทำการเชื่อมต่อใหม่ทุกครั้งเมื่อมีการเรียกใช้ฟังก์ชัน

14.1.2 การยกเลิกการเชื่อมต่อกับฐานข้อมูล MySQL ฟังก์ชัน `mysql_pconnect ()` ไม่ต้องใช้คำสั่งยกเลิกการเชื่อมต่อกับ MySQL link identifier หลังจากที่ใช้งานระบบฐานข้อมูลเสร็จแล้ว เพราะจะเก็บไว้ใช้งานในครั้งต่อไป ส่วนฟังก์ชัน `mysql_connect ()` จะต้องใช้ฟังก์ชัน `mysql_close ()` สำหรับยกเลิกการเชื่อมต่อกับ MySQL link identifier เพื่อคืนหน่วยความจำให้กับระบบ

รูปแบบการใช้งานฟังก์ชัน `mysql_connect ()` และ `mysql_pconnect ()` ใช้สำหรับการเชื่อมต่อกับระบบฐานข้อมูล MySQL มีรูปแบบดังนี้

รูปแบบ

```
resource mysql_connect | mysql_pconnect (string $server, string $username, string $password)
```

เมื่อ `$server` หมายถึง ชื่อเครื่องเซิร์ฟเวอร์ที่ติดตั้งระบบฐานข้อมูล MySQL หรือหมายเลข IP หรือสามารถใช้ `localhost` (กรณีเป็นเครื่องเซิร์ฟเวอร์เดียวกันกับเว็บเซิร์ฟเวอร์ปัจจุบันที่กำลังใช้งานอยู่)

`$username` หมายถึง ชื่อผู้ใช้ที่มีสิทธิเชื่อมต่อและเรียกใช้งานระบบฐานข้อมูล MySQL

`$password` หมายถึง รหัสผ่านของผู้ใช้ที่มีสิทธิเชื่อมต่อและเรียกใช้งานระบบฐานข้อมูล MySQL

รูปแบบการใช้งานฟังก์ชัน `mysql_close ()` ใช้สำหรับการยกเลิกการเชื่อมต่อกับระบบฐานข้อมูล MySQL มีรูปแบบการใช้งานดังนี้

รูปแบบ

```
bool mysql_close ([ resource $link_identifier = NULL ] )
```

เมื่อ `$link_identifier` หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.1 การเชื่อมต่อและยกเลิกการเชื่อมต่อระบบฐานข้อมูล MySQL ด้วยฟังก์ชัน `mysql_connect ()` และ `mysql_close ()`

```
1 <?php
2     $link = mysql_connect ("localhost", "mysql_user", "mysql_password");
3     if (!$link) {
4         echo "<br>ไม่สามารถเชื่อมต่อกับระบบฐานข้อมูล MySQL<br>";
5         exit;
6     } else {
7         echo "<br>การเชื่อมต่อเสร็จสมบูรณ์ ด้วยฟังก์ชัน mysql_connect<br>";
8         mysql_close ($link);
9     }
10 ?>
```

จากตัวอย่างที่ 14.1 การเชื่อมต่อและยกเลิกการเชื่อมต่อด้วยฟังก์ชัน `mysql_connect ()` อธิบายรายละเอียด ดังนี้

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดตัวแปร \$link (ทำหน้าที่เป็น MySQL link identifier) รับผลจากการเรียกใช้ฟังก์ชัน mysql_connect () โดยกำหนดพารามิเตอร์ ประกอบด้วย

"localhost" คือ ชื่อเครื่องเซิร์ฟเวอร์ที่ติดตั้งระบบฐานข้อมูล MySQL โดยการที่จะกำหนดแบบนี้ได้นั้นแสดงว่าเครื่องเซิร์ฟเวอร์ที่เรียกใช้งานเป็นเครื่องเดียวกันกับเว็บเซิร์ฟเวอร์ปัจจุบันที่กำลังใช้งานอยู่

"mysql_user" คือ ชื่อผู้ใช้ที่มีสิทธิเชื่อมต่อและเรียกใช้งานระบบฐานข้อมูล MySQL

"mysql_password" คือ รหัสผ่านของผู้ใช้ที่มีสิทธิเชื่อมต่อและเรียกใช้งานระบบฐานข้อมูล MySQL

บรรทัดที่ 3 ใช้โครงสร้างเงื่อนไข if ตรวจสอบค่าของตัวแปร \$link ว่าสามารถเชื่อมต่อระบบฐานข้อมูล MySQL ได้หรือไม่ หากเชื่อมต่อไปได้ ไปบรรทัดที่ 4 หากสามารถเชื่อมต่อได้ไปบรรทัดที่ 6

บรรทัดที่ 4 แสดงข้อความ "ไม่สามารถเชื่อมต่อกับระบบฐานข้อมูล MySQL"

บรรทัดที่ 5 หยุดการทำงานคำสั่งที่เหลือทั้งหมด

บรรทัดที่ 6 สิ้นสุดสคริปต์โครงสร้างเงื่อนไข if และเริ่มโครงสร้างเงื่อนไข else (กรณีเป็นเท็จจากบรรทัดที่ 3)

บรรทัดที่ 7 แสดงข้อความ "การเชื่อมต่อเสร็จสมบูรณ์ ด้วยฟังก์ชัน mysql_connect" หรือโดยความหมาย คือ ใช้คำสั่งอื่นๆ ตามที่ผู้พัฒนาต้องการจะกำหนดให้ระบบทำอะไรต่อจากนี้เป็นต้นไป

บรรทัดที่ 8 ใช้ฟังก์ชัน mysql_close () เพื่อปิดการเชื่อมต่อกับระบบฐานข้อมูล MySQL

บรรทัดที่ 9 สิ้นสุดสคริปต์โครงสร้างเงื่อนไข else

บรรทัดที่ 10 สิ้นสุดสคริปต์ PHP

ตัวอย่างที่ 14.2 การเชื่อมต่อระบบฐานข้อมูล MySQL ด้วยฟังก์ชัน mysql_pconnect ()

```

1 <?php
2     $link = mysql_pconnect ("localhost", "mysql_user", "mysql_password");
3     if (!$link) {
4         echo "<br>ไม่สามารถเชื่อมต่อกับระบบฐานข้อมูล MySQL<br>";
5         exit;
6     } else {
7         echo "<br>การเชื่อมต่อเสร็จสมบูรณ์ ด้วยฟังก์ชัน mysql_pconnect<br>";
8     }
9     ?>

```

จากตัวอย่างที่ 14.1 และ 14.2 จะเห็นได้ว่าการใช้งานฟังก์ชันอื่นๆ หลังจากเชื่อมต่อกับระบบฐานข้อมูล MySQL เหมือนกันทุกประการแต่ต่างกันว่า หากใช้ฟังก์ชัน `mysql_connect ()` จำเป็นต้องใช้ `mysql_close ()` ร่วมด้วยเสมอเมื่อต้องการยกเลิกการเชื่อมต่อ

นอกจากเรียกใช้ฟังก์ชันสำหรับเชื่อมต่อ/ยกเลิกการเชื่อมต่อระหว่างภาษา PHP และระบบฐานข้อมูล MySQL แล้ว ยังมีการตรวจสอบผลลัพธ์ของฟังก์ชันก่อนจะดำเนินการใดๆ ต่อไป หรือหากไม่ใช้การตรวจสอบตามตัวอย่างข้างต้น อาจใช้เทคนิค Error Suppression ร่วมกับฟังก์ชัน `die ()` หรือ `exit ()` มีตัวอย่างดังนี้

ตัวอย่างที่ 14.3 เทคนิคการประยุกต์ใช้ Error Suppression ร่วมกับฟังก์ชัน `die ()` หรือ `exit ()`

```

1 <?php
2     $link = @mysql_pconnect ("localhost", "mysql_user", "mysql_password");
3     if (!$link) die ("ไม่สามารถเชื่อมต่อกับฐานข้อมูลได้: " . mysql_error ( ));
4     echo "การเชื่อมต่อเสร็จสมบูรณ์";
5     ?>

```

แต่อย่างไรก็ตาม การใช้ฟังก์ชัน `mysql_close ()` ไม่ใช่สิ่งที่จำเป็น เนื่องจากภาษา PHP จะปิดการเชื่อมต่อเองโดยอัตโนมัติอยู่แล้วเมื่อสิ้นสุดสคริปต์ของเว็บเพจนั้นๆ จึงทำให้การเขียนสคริปต์คำสั่งทำได้สะดวกขึ้น เพราะในบางครั้งการเปิดการเชื่อมต่อกับระบบฐานข้อมูล MySQL จะขึ้นอยู่กับเงื่อนไขต่างๆ ที่กำหนด ดังนั้นจึงไม่ต้องมาคอยสั่งปิดการเชื่อมต่อตามเงื่อนไขนั้นๆ

14.2 ฟังก์ชันสำหรับการเลือกใช้งานฐานข้อมูล

การที่จะเรียกใช้งานตารางฐานข้อมูลใดๆ ได้นั้น จำเป็นต้องเลือกฐานข้อมูลที่มีตารางฐานข้อมูลนั้นบรรจุอยู่ก่อน หากใช้คำสั่งโดยตรง ก็คือ การใช้คำสั่ง USE แล้วตามด้วยฐานข้อมูลที่ต้องการเลือกใช้สำหรับภาษา PHP มีฟังก์ชันที่ใช้สำหรับเลือกใช้งานฐานข้อมูล คือ `mysql_select_db ()` มีรูปแบบดังนี้

รูปแบบ

```
bool mysql_select_db ( string $database_name [, resource $link_identifier ] )
```

เมื่อ `$database_name` หมายถึง ชื่อฐานข้อมูลที่ต้องการเลือกใช้งาน

`$link_identifier` หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.4 แสดงตัวอย่างการใช้งานฟังก์ชัน `mysql_select_db ()`

```

1 <?php
2     $link = mysql_pconnect ("localhost", "mysql_user", "mysql_password");
3     if (!$link) {
4         die ("ไม่สามารถเชื่อมต่อกับฐานข้อมูลได้: " . mysql_error ( ));

```

```

5         }
6         $db_selected = mysql_select_db ("employees", $link);
7         if (!$db_selected) {
8             die ("ไม่สามารถเลือกใช้งานฐานข้อมูล employees ได้ : " . mysql_error ());
9         }
10    ?>

```

หากไม่ใช้ฟังก์ชัน `mysql_select_db ()` สามารถประยุกต์ใช้ฟังก์ชัน `mysql_query ()` แทนได้ ระบุชื่อฐานข้อมูลที่ต้องการเลือกใช้งานโดยตรงไปยังระบบฐานข้อมูล MySQL โดยใช้คำสั่งสอบถาม USE แล้วตามด้วยชื่อฐานข้อมูลที่ต้องการเลือกใช้งาน มีตัวอย่างดังนี้

ตัวอย่างที่ 14.5 การใช้ฟังก์ชัน `mysql_query ()` แทน ฟังก์ชัน `mysql_select_db ()`

```

1    <?php
2        $link = mysql_pconnect ("localhost", "mysql_user", "mysql_password");
3        if (!$link) {
4            die ("ไม่สามารถเชื่อมต่อกับฐานข้อมูลได้ : " . mysql_error ());
5        }
6        $db_selected = mysql_query ("USE employees;" , $link);
7        if (!$db_selected) {
8            die ("ไม่สามารถเลือกฐานข้อมูล employees ได้ : " . mysql_error ());
9        }
10    ?>

```

การใช้ฟังก์ชัน `mysql_query ()` จะมีความสามารถการทำงานได้เร็วกว่าการใช้ฟังก์ชัน `mysql_select_db ()` เพราะเป็นการใช้คำนิยามข้อมูลโดยตรงไปยังระบบฐานข้อมูล MySQL สำหรับรายละเอียดของฟังก์ชัน `mysql_query ()` จะกล่าวถึงในลำดับต่อไป

14.3 การส่งคำสั่งสอบถามหรือคำสั่งไปยังระบบฐานข้อมูล MySQL

การส่งคำสั่งสอบถามหรือคำสั่งไปยังระบบฐานข้อมูล MySQL การที่จะเลือกใช้คำสั่งสอบถามหรือคำสั่งอะไรนั้น ขึ้นอยู่กับว่าต้องการให้ระบบฐานข้อมูล MySQL ทำงานอะไร เช่น การสอบถามเพื่อเรียกดูข้อมูล การใช้คำสั่งเพื่อเพิ่มข้อมูล เปลี่ยนแปลงค่าข้อมูล เป็นต้น (ศึกษาเพิ่มเติมได้ในบทที่ 13) ทั้งนี้ภาษา PHP ไม่ได้เป็นผู้ประมวลผลคำสั่งสอบถามหรือคำสั่ง SQL เอง แต่เป็นเพียงผู้ส่งคำสั่งสอบถามหรือคำสั่งมาตรฐาน SQL ที่กำหนดขึ้นในรูปแบบข้อความไปยังระบบฐานข้อมูล MySQL เท่านั้น การที่จะสามารถใช้คำสั่งสอบถามหรือคำสั่งมาตรฐาน SQL ได้นั้น จำเป็นต้องใช้ฟังก์ชัน `mysql_query ()` สำหรับการส่งคำสั่งสอบถามหรือคำสั่งไปยังระบบฐานข้อมูล MySQL มีรูปแบบดังนี้



รูปแบบ

```
resource mysql_query ( string $query [, resource $link_identifier = NULL ] )
```

เมื่อ \$query หมายถึง คำสอบถามหรือคำสั่งที่ต้องการใช้งาน

\$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.6 การส่งคำสอบถามเพื่อเรียกดูข้อมูลไปยังระบบฐานข้อมูล MySQL

```

1  <?php
2      $host = "localhost";
3      $user = "parinya";
4      $password = "12345";
5      $link = mysql_pconnect ($host, $user, $password);
6      $db_selected = mysql_select_db ("company", $link);
7      $query = sprintf ("SELECT firstname, lastname, address FROM employees
WHERE address='%s';", "Suratthani");
8      $result = mysql_query ($query, $link);
9      if (!$result) die ("Can not Query because: " . mysql_error ( ));
10     echo "<table border=1 width=50%>";
11     while ($row = mysql_fetch_assoc ($result)) {
12         echo "<tr>";
13         echo "<td> " . $row ["firstname"] . "</td>";
14         echo "<td> " . $row ["lastname"] . "</td>";
15         echo "<td> " . $row ["address"] . "</td>";
16         echo "</tr>";
17     }
18     echo "<table>";
19     mysql_free_result ($result);
20     ?>

```

ผลลัพธ์

Parinya	Noidonprai	Suratthani
Jaidee	Ruktum	Suratthani
Dennis	Robert	Suratthani

จากตัวอย่างที่ 14.6 การส่งคำถามเพื่อเรียกดูข้อมูลไปยังระบบฐานข้อมูล MySQL อธิบาย ดังนี้ (ตัวอย่างที่ 14.6 และผลลัพธ์ที่ได้ จะใช้ประกอบการบรรยายการใช้งานฟังก์ชันสำหรับอ่านข้อมูลผลลัพธ์ ในหัวข้อถัดไป)

บรรทัดที่ 1 เริ่มต้นสคริปต์ PHP

บรรทัดที่ 2 กำหนดให้ตัวแปร \$host มีค่าเท่ากับ "localhost"

บรรทัดที่ 3 กำหนดให้ตัวแปร \$user มีค่าเท่ากับ "parinya" (ชื่อผู้ใช้ที่มีสิทธิใช้งานฐานข้อมูล)

บรรทัดที่ 4 กำหนดให้ตัวแปร \$password มีค่าเท่ากับ "12345"

บรรทัดที่ 5 กำหนดให้ตัวแปร \$link ทำหน้าที่เป็น MySQL link identifier รับค่าที่ส่งมาจากฟังก์ชัน mysql_pconnect () (ใช้สำหรับเชื่อมต่อระบบฐานข้อมูล MySQL) ระบุพารามิเตอร์ 3 ค่า ประกอบด้วย 1) ชื่อเซิร์ฟเวอร์ 2) ชื่อผู้ใช้ที่มีสิทธิใช้งานฐานข้อมูล และ 3) รหัสผ่าน

บรรทัดที่ 6 กำหนดให้ตัวแปร \$db_selected รับค่าที่ส่งมาจากฟังก์ชัน mysql_select_db () ใช้สำหรับเลือกฐานข้อมูล ระบุพารามิเตอร์ 2 ค่า ประกอบด้วย 1) ชื่อฐานข้อมูล และ 2) ตัวแปรที่ทำหน้าที่ MySQL link identifier

บรรทัดที่ 7 กำหนดให้ตัวแปร \$query รับค่าที่ส่งมาจากฟังก์ชัน sprintf () ใช้สำหรับต่อประโยคข้อความให้เป็นประโยคเดียวกัน

บรรทัดที่ 8 กำหนดให้ตัวแปร \$result รับค่าที่ส่งมาจากฟังก์ชัน mysql_query () ใช้สำหรับส่งคำถามไปยังระบบฐานข้อมูล MySQL โดยระบุค่าพารามิเตอร์ 2 ค่า ประกอบด้วย 1) คำถาม และ 2) MySQL link identifier

บรรทัดที่ 9 ตรวจสอบผลของการสอบถามข้อมูลจากตัวแปร \$result โดยใช้โครงสร้างเงื่อนไข if ในการตรวจสอบ หากไม่สามารถสอบถามข้อมูลได้ สั่งให้หยุดการทำงานบรรทัดที่เหลือทั้งหมดด้วยฟังก์ชัน die () และแสดงเหตุผลที่ไม่สามารถสอบถามข้อมูลได้

บรรทัดที่ 10 เริ่มต้นแท็ก <table> (ใช้สำหรับแสดงข้อมูลให้เป็นระเบียบ)

บรรทัดที่ 11 ใช้โครงสร้างเงื่อนไขทำซ้ำ while โดยจะวนรอบอ่านข้อมูลไปเรื่อยๆ มีเงื่อนไขการอ่านค่าดังนี้ กำหนดให้ตัวแปร \$row รับค่าที่ส่งมาจากฟังก์ชัน mysql_fetch_assoc (\$result) ซึ่งฟังก์ชัน mysql_fetch_assoc () ใช้สำหรับอ่านค่าชุดของข้อมูลครั้งละ 1 แถว หรือ 1 ระเบียบนโครงสร้างเงื่อนไขทำซ้ำ while จะหยุดก็ต่อเมื่อไม่มีชุดข้อมูลให้อ่านและไปทำคำสั่งในบรรทัดที่ 18 หากยังสามารถอ่านชุดข้อมูลได้ ไปทำคำสั่งบรรทัดที่ 12 ถึงบรรทัดที่ 17

บรรทัดที่ 12 เริ่มต้นแท็ก <tr>

บรรทัดที่ 13 เริ่มต้นแท็ก <td> แสดงค่าของตัวแปร \$row [คีย์ firstname] ปิดแท็ก </td>

บรรทัดที่ 14 เริ่มต้นแท็ก <td> แสดงค่าของตัวแปร \$row [คีย์ lastname] ปิดแท็ก </td>

บรรทัดที่ 15 เริ่มต้นแท็ก <td> แสดงค่าของตัวแปร \$row [คีย์ address] ปิดแท็ก </td>

บรรทัดที่ 16 ปิดแท็ก </tr>



บรรทัดที่ 17 สิ้นสุดสคริปต์โครงสร้างเงื่อนไขทำซ้ำ while กลับไปตรวจสอบเงื่อนไขบรรทัดที่ 11
 บรรทัดที่ 18 ปิดแท็ก </table>
 บรรทัดที่ 19 ฟังก์ชัน mysql_free_result () ใช้สำหรับคืนหน่วยความจำให้กับระบบ
 บรรทัดที่ 20 สิ้นสุดสคริปต์ PHP

ฟังก์ชัน mysql_query () จะใช้คำสอบถามหรือคำสั่งในรูปแบบข้อความ ดังนั้นจึงสามารถใช้ร่วมกับฟังก์ชัน sprintf () เพื่อต่อข้อความให้เป็นประโยคเดียวกัน ก่อนส่งเข้าไปประมวลผลในฟังก์ชัน mysql_query () และผลลัพธ์ที่ได้จากฟังก์ชันนี้จะเป็นอะไรนั้น ขึ้นอยู่กับคำสอบถามหรือคำสั่งที่กำหนด สำหรับผลลัพธ์ที่คืนกลับมาจากฟังก์ชัน mysql_query () นั้นจะขึ้นอยู่กับคำสอบถามหรือคำสั่งที่ส่งเข้าไปในฟังก์ชัน สามารถสรุปได้ดังนี้

14.3.1 หากเป็นคำสอบถามสำหรับการอ่านข้อมูล เช่น SELECT, SHOW, DESCRIBE หากทำงานสำเร็จจะคืนค่ากลับมาเป็นชุดข้อมูล ผลลัพธ์ที่ได้หรือเรียกว่า result set

14.3.2 หากเป็นคำสั่งในกรณีอื่นๆ เช่น การเปลี่ยนแปลงข้อมูล (INSERT, UPDATE, DELETE) หากการทำงานสำเร็จจะคืนค่า true

14.3.3 หากเกิดข้อผิดพลาดจากการใช้ฟังก์ชัน mysql_query () เช่น เขียนคำสอบถามหรือคำสั่งผิด จะคืนค่า false

14.4 การอ่านข้อมูลผลลัพธ์

หลังจากการใช้ฟังก์ชัน mysql_query () หากคำสอบถามเกี่ยวข้องกับการค้นหาข้อมูล ข้อมูลที่ได้จะเป็นผลลัพธ์มาจากขอบเขตข้อมูลใดๆ เรียกว่า result set โดยข้อมูลนี้จะถูกเก็บไว้ในตัวแปรที่ใช้รับค่าที่ส่งมาจากฟังก์ชัน mysql_query () สำหรับการอ่านข้อมูลนั้นสามารถประยุกต์ใช้ได้หลายฟังก์ชัน ตัวอย่างฟังก์ชันที่ใช้สำหรับอ่านข้อมูลผลลัพธ์ มีดังนี้

14.4.1 ฟังก์ชัน mysql_num_rows ()

เป็นฟังก์ชันที่ใช้สำหรับนับจำนวนแถวผลลัพธ์ มีรูปแบบดังนี้

รูปแบบ

```
int mysql_num_rows ( resource $result )
```

เมื่อ \$result หมายถึง ตัวแปรที่เก็บผลลัพธ์ที่ได้จากฟังก์ชัน mysql_query ()

ตัวอย่างที่ 14.7 การใช้ฟังก์ชัน mysql_num_rows ()

```
1 <?php
2     $host = "localhost";
3     $user = "parinya";
4     $password = "12345";
```



```

5      $link = mysql_pconnect ($host, $user, $password);
6      $db_selected = mysql_select_db ("company", $link);
7      $query = sprintf ("SELECT firstname, lastname, address FROM employees
WHERE address='%s';", "Suratthani");
8      $result = mysql_query ($query, $link);
9      if (!$result) die ("Can not Query because: " . mysql_error ( ));
10     echo "ข้อมูลที่อ่านได้มีทั้งหมด " . mysql_num_rows ($result) . " ระเบียบ";
11     mysql_free_result ($result)
12     ?>

```

ผลลัพธ์ (อ้างอิงจากตัวอย่างที่ 14.6 และผลลัพธ์)

ข้อมูลที่อ่านได้มีทั้งหมด 3 ระเบียบ

14.4.2 ฟังก์ชัน mysql_result ()

ฟังก์ชันสำหรับการอ่านผลลัพธ์แบบระบุแถวและคอลัมน์ วิธีนี้เป็นการอ่านข้อมูลโดยใช้ฟังก์ชันในการดึงข้อมูล result set ในคอลัมน์ และลำดับแถวที่ระบุโดยมีรูปแบบดังนี้

```
string mysql_result ( resource $result , int $row [, mixed $field = 0 ] )
```

เมื่อ \$result หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากฟังก์ชัน mysql_query ()

\$row หมายถึง แถวของข้อมูล

\$field หมายถึง ขอบเขตข้อมูลหรือคอลัมน์ของข้อมูล (เป็นตัวเลขหรือข้อความก็ได้)

ตัวอย่างที่ 14.8 การใช้ฟังก์ชัน mysql_result ()

```

1      <?php
2      $host = "localhost";
3      $user = "parinya";
4      $password = "12345";
5      $link = mysql_pconnect ($host, $user, $password);
6      $db_selected = mysql_select_db ("company", $link);
7      $query = sprintf ("SELECT firstname, lastname, address FROM employees
WHERE address='%s';", "Suratthani");
8      $result = mysql_query ($query, $link);
9      if (!$result) die ("Can not Query because: " . mysql_error ( ));
10     echo mysql_result ($result, 1);
11     echo "<br>";

```



```

12     echo mysql_result ($result, 2);
13     mysql_free_result ($result)
14     ?>

```

ผลลัพธ์ (อ้างอิงจากตัวอย่างที่ 14.6 และผลลัพธ์)

```

Jaidee
Dennis

```

14.4.3 ฟังก์ชัน mysql_fetch_row ()

การอ่านผลลัพธ์แบบตามลำดับแถว เป็นการอ่านแบบเลื่อนพอยน์เตอร์ (Pointer) ไปยังตำแหน่งที่ต้องการอ่าน แล้วอ่านข้อมูลที่พอยน์เตอร์ชี้อยู่ ด้วยฟังก์ชัน mysql_fetch_row () ข้อมูลที่อ่านได้จะอยู่ในรูปแบบของอาร์เรย์เรียกลำดับตามอินเด็กซ์ รูปแบบของฟังก์ชันมีดังนี้

รูปแบบ

```
array mysql_fetch_row ( resource $result )
```

เมื่อ \$result หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากฟังก์ชัน mysql_query ()

ตัวอย่างที่ 14.9 การใช้ฟังก์ชัน mysql_fetch_row ()

```

1     <?php
2         $host = "localhost";
3         $user = "parinya";
4         $password = "12345";
5         $link = mysql_pconnect ($host, $user, $password);
6         $db_selected = mysql_select_db ("company", $link);
7         $query = sprintf ("SELECT firstname, lastname, address FROM employees
WHERE address='%s';", "Suratthani");
8         $result = mysql_query ($query, $link);
9         if (!$result) die ("Can not Query because: " . mysql_error ( ));
10        echo "<table border=1 width=50%>";
11        while ($row = mysql_fetch_row ($result)) {
12            echo "<tr>";
13            echo "<td> " . $row [0] . "</td>";
14            echo "<td> " . $row [1] . "</td>";
15            echo "<td> " . $row [2] . "</td>";
16            echo "</tr>";

```

```

17     }
18     echo "<table>";
19     mysql_free_result ($result)
20 ?>
    
```

ผลลัพธ์ (ผลลัพธ์ที่ได้จะเหมือนกับตัวอย่างที่ 14.6)

Parinya	Noidonprai	Suratthani
Jaidee	Ruktum	Suratthani
Dennis	Robert	Suratthani

14.4.4 ฟังก์ชัน mysql_fetch_array ()

การอ่านผลลัพธ์แบบอาร์เรย์เป็นการอ่านข้อมูลจาก result set ในรูปแบบคีย์และค่าอาร์เรย์ โดยคีย์จะเป็นชื่อขอบเขตข้อมูล และค่าจะเป็นข้อมูลที่ส่งมาจากฟังก์ชัน mysql_fetch_array () มีรูปแบบดังนี้

```
array mysql_fetch_array ( resource $result [, int $result_type ] )
```

เมื่อ \$result หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากฟังก์ชัน mysql_query ()
 \$result_type หมายถึง กำหนดค่าอินเด็กซ์ของอาร์เรย์ (ผลลัพธ์ที่เก็บอยู่ในหน่วยความจำ) สามารถกำหนดได้ 3 ค่า คือ MYSQL_ASSOC (ใช้ชื่อฟิลด์เป็นอินเด็กซ์), MYSQL_NUM (ใช้ตัวเลขลำดับเป็นอินเด็กซ์) และ MYSQL_BOTH (ใช้ชื่อฟิลด์และตัวเลขลำดับเป็นอินเด็กซ์ร่วมกัน) หากไม่กำหนด ตัวแปร \$result_type จะมีอินเด็กซ์เป็นแบบ MYSQL_BOTH เป็นค่าปริยาย

ตัวอย่างที่ 14.10 การใช้ฟังก์ชัน mysql_fetch_array ()

```

1 <?php
2     $host = "localhost";
3     $user = "parinya";
4     $password = "12345";
5     $link = mysql_pconnect ($host, $user, $password);
6     $db_selected = mysql_select_db ("company", $link);
7     $query = sprintf ("SELECT firstname, lastname, address FROM employees
    WHERE address='%s'", "Suratthani");
8     $result = mysql_query ($query, $link);
9     if (!$result) die ("Can not Query because: " . mysql_error ( ));
    
```

```

10     echo "<table border=1 width=50%>";
11     while ($row = mysql_fetch_array ($result, MYSQL_NUM)) {
12         echo "<tr>";
13         echo "<td> " . $row [0] . "</td>";
14         echo "<td> " . $row [1] . "</td>";
15         echo "<td> " . $row [2] . "</td>";
16         echo "</tr>";
17     }
18     echo "<table>";
19     mysql_free_result ($result)
20     ?>
    
```

ผลลัพธ์ (ผลลัพธ์ที่ได้จะเหมือนกับตัวอย่างที่ 14.6)

Parinya	Noidonprai	Suratthani
Jaidee	Ruktum	Suratthani
Dennis	Robert	Suratthani

14.4.5 ฟังก์ชัน mysql_fetch_object ()

เป็นฟังก์ชันที่ใช้สำหรับอ่านข้อมูลแบบออบเจกต์ การอ่านแบบนี้เหมาะสำหรับผู้ที่ถนัดในแนว OOP โดยใช้ฟังก์ชัน mysql_fetch_object () ผลลัพธ์ที่ได้จะอยู่ในรูปแบบออบเจกต์ ทั้งนี้การอ้างถึงขอบเขตข้อมูลต่างๆ จะใช้รูปแบบการอ้างถึงพรีอเปอร์ตี้ของออบเจกต์ จะใช้หลักการเลื่อนพอยน์เตอร์ไปยังแถวที่ต้องการอ่านข้อมูล คล้ายกับฟังก์ชันที่ได้ศึกษาผ่านมาแล้ว มีรูปแบบดังนี้

รูปแบบ

```
object mysql_fetch_object ( resource $result )
```

เมื่อ \$result หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากฟังก์ชัน mysql_query ()

ตัวอย่างที่ 14.11 การใช้ฟังก์ชัน mysql_fetch_object ()

```

1     <?php
2         $host = "localhost";
3         $user = "parinya";
4         $password = "12345";
5         $link = mysql_pconnect ($host, $user, $password);
6         $db_selected = mysql_select_db ("company", $link);
    
```

```

7      $query = sprintf ("SELECT firstname, lastname, address FROM employees
WHERE address='%s';", "Suratthani");
8      $result = mysql_query ($query, $link);
9      if (!$result) die ("Can not Query because: " . mysql_error ( ));
10     echo "<table border=1 width=50%>";
11     while ($row = mysql_fetch_object ($result)) {
12         echo "<tr>";
13         echo "<td> " . $row->firstname . "</td>";
14         echo "<td> " . $row->lastname . "</td>";
15         echo "<td> " . $row->address . "</td>";
16         echo "</tr>";
17     }
18     echo "<table>";
19     mysql_free_result ($result)
20     ?>

```

ผลลัพธ์ (ผลลัพธ์ที่ได้จะเหมือนกับตัวอย่างที่ 14.6)

Parinya	Noidonprai	Suratthani
Jaidee	Ruktum	Suratthani
Dennis	Robert	Suratthani

14.5 การเปลี่ยนแปลงข้อมูล

การเปลี่ยนแปลงข้อมูลในระบบฐานข้อมูล MySQL ประกอบด้วย 1) การแทรกข้อมูลใหม่ (INSERT) 2) การแก้ไขข้อมูล (UPDATE) และ 3) การลบข้อมูล (DELETE) มีแนวทางประยุกต์ใช้ดังนี้ (ใช้ตารางที่ 14.1 เป็นตารางอ้างอิงขอบเขตข้อมูลเมื่อมีการเปลี่ยนแปลงข้อมูลในรูปแบบต่าง)

ตารางที่ 14.1 ตัวอย่างโครงสร้างตาราง customer

ขอบเขตข้อมูล	ชนิด	ว่างเปล่า (null)	ค่าปริยาย	คีย์	หมายเหตุ
cus_id	VARCAHR(5)	NOT NULL		PK	รหัสลูกค้า
name	VARCHAR(40)	-	-	-	ชื่อ-สกุลลูกค้า
address	VARCHAR(40)	-	-	-	ที่อยู่ลูกค้า
email	VARCHAR(40)	-	-	-	อีเมล
tel	VARCHAR(20)	-	-	-	เบอร์โทรศัพท์



สร้างตารางข้อมูล customer มีขั้นตอนดังนี้

ตัวอย่างที่ 14.12 ตัวอย่างการสร้างตาราง customer ในฐานข้อมูล company

```
mysql> USE company;
Database changed
mysql> CREATE TABLE customer (
-> cus_id VARCHAR (5) NOT NULL PRIMARY KEY,
-> name VARCHAR (40),
-> address VARCHAR (40),
-> email VARCHAR (40),
-> tel VARCHAR (20));
```

เพื่อเพิ่มความเข้าใจให้สามารถประยุกต์ใช้งานได้ ดังนั้นจะอธิบายแต่ละส่วนแยกเป็นเว็บเพจ ตามหน้าที่การใช้งาน มีรายละเอียดดังนี้

14.5.1 การแทรกข้อมูลใหม่

การเปลี่ยนแปลงข้อมูลในระบบฐานข้อมูล MySQL ด้วยรูปแบบการแทรกข้อมูลใหม่นั้น ก่อนอื่น จะเริ่มด้วยการสร้างหน้าเว็บเพจสำหรับใช้เป็นส่วนนำเข้าข้อมูล ดังนี้

- 1) ออกแบบแบบเว็บเพจส่วนนำเข้าข้อมูล (customer_form.php)

ตัวอย่างที่ 14.13 สคริปต์เว็บเพจสำหรับใช้เป็นส่วนนำเข้าข้อมูลลูกค้า (customer_form.php)

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>เพิ่มข้อมูลลูกค้าใหม่</title>
</head>
<body>
<p>เพิ่มข้อมูลลูกค้าใหม่</p>
<form id="form1" name="form1" method="POST" action="customer_insert.php">
  <table width="550" border="1">
    <tr>
      <td width="135">รหัสลูกค้า</td>
      <td width="249"><label>
        <input name="cus_id" type="text" id="cus_id" size="20" maxlength="5" />
      </label></td>
```



```

</tr>
<tr>
<td>ชื่อ-สกุลลูกค้า</td>
<td><label>
<input name="name" type="text" id="name" size="40" maxlength="40" />
</label></td>
</tr>
<tr>
<td>ที่อยู่</td>
<td><label>
<input name="address" type="text" id="address" size="40" maxlength="40" />
</label></td>
</tr>
<tr>
<td>อีเมล</td>
<td><label>
<input name="email" type="text" id="email" size="40" maxlength="40" />
</label></td>
</tr>
<tr>
<td>เบอร์โทรศัพท์</td>
<td><label>
<input name="tel" type="text" id="tel" size="20" maxlength="20" />
</label></td>
</tr>
<tr>
<td>&nbsp;</td>
<td><label>
<input type="submit" name="btnSend" id="btnSend" value="Save" />
</label></td>
</tr>
</table>

```

4 ส่วนนำเข้าสู่ลูกค้า name

5 ส่วนนำเข้าสู่ที่อยู่ลูกค้า address

6 ส่วนนำเข้าสู่อีเมลลูกค้า email

7 ส่วนนำเข้าสู่เบอร์โทรศัพท์ tel

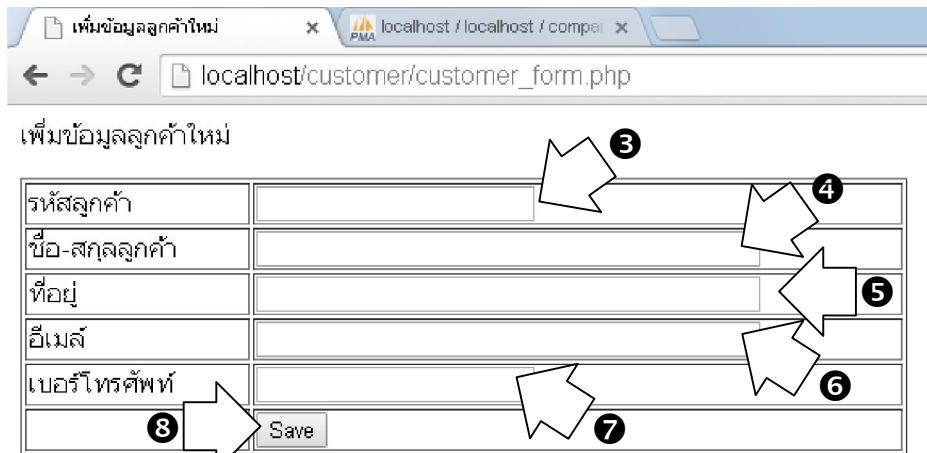
8 ปุ่มนำส่งข้อมูล



```
</form>
</body>
</html>
```

จากตัวอย่างที่ 14.13 หน้าเว็บเพจสำหรับใช้เป็นส่วนนำเข้าสู่ข้อมูลลูกค้า มีส่วนที่สำคัญสำหรับใช้เป็นข้อมูลนำส่งไปยังสคริปต์เพื่อการบันทึกข้อมูลต่อไป อธิบายดังนี้

- ❶ ไฟล์สคริปต์ที่จะทำงานต่อเมื่อกดปุ่มบันทึกข้อมูล
- ❷ รูปแบบการส่งเมธอดแบบ POST
- ❸ ส่วนนำเข้ารหัสลูกค้า ใช้ชื่อ cus_id
- ❹ ส่วนนำเข้าชื่อลูกค้า ใช้ชื่อ name
- ❺ ส่วนนำเข้าที่อยู่ลูกค้า ใช้ชื่อ address
- ❻ ส่วนนำเข้าอีเมลลูกค้า ใช้ชื่อ email
- ❼ ส่วนนำเข้าเบอร์โทรศัพท์ ใช้ชื่อ tel
- ❽ ปุ่มนำส่งข้อมูล



ภาพที่ 14.1 หน้าเว็บเพจสำหรับใช้เป็นส่วนนำเข้าสู่ข้อมูลลูกค้า

2) หน้าเว็บเพจสำหรับแทรกข้อมูลลูกค้าใหม่ลงในระบบฐานข้อมูล MySQL (customer_insert.php) เมื่อหน้าเว็บเพจนี้ทำงานจะทำการแทรกข้อมูลลูกค้าใหม่ลงในฐานข้อมูล MySQL แล้วใช้ฟังก์ชัน header () ย้ายหน้าเว็บเพจไปยังหน้ารายงานข้อมูลลูกค้า (report.php)

ตัวอย่างที่ 14.14 สคริปต์สำหรับแทรกข้อมูลลูกค้าใหม่ (customer_insert.php)

```
<?php
ob_start();
$host = "localhost";
$user = "parinya";
$password = "12345";
```



```

$link = mysql_pconnect ($host, $user, $password);
$db_selected = mysql_select_db ("company", $link);
$query = sprintf ("INSERT INTO customer VALUES ('%s', '%s', '%s', '%s', '%s');",
$_POST["cus_id"], $_POST["name"], $_POST["address"], $_POST["email"], $_POST["tel"]);
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ( ));
header('Location: report.php');
?>

```

3) สคริปต์รายงานผลข้อมูลลูกค้าทั้งหมด (report.php) รายละเอียด ดังนี้
ตัวอย่างที่ 14.15 สคริปต์สำหรับรายงานผลข้อมูลลูกค้าทั้งหมด

```

<?php
    $host = "localhost";
    $user = "parinya";
    $password = "12345";
    $link = mysql_pconnect ($host, $user, $password);
    $db_selected = mysql_select_db ("company", $link);
    $query = sprintf ("SELECT * FROM customer");
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ( ));
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>รายงานข้อมูลลูกค้า</title>
</head>
<body>
<p>รายงานข้อมูลลูกค้า</p>
<table width="70%" border="1">
    <tr>
        <td width="10%">รหัสลูกค้า</td>
        <td width="20%">ชื่อ-สกุลลูกค้า</td>
        <td width="20%">ที่อยู่</td>

```



```

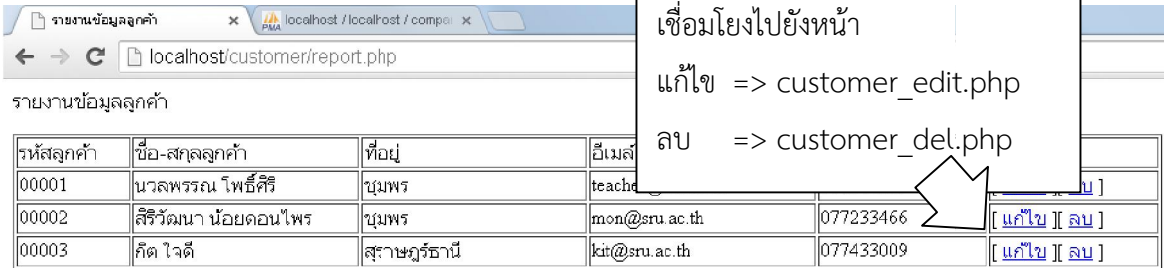
<td width="20%">อีเมล</td>
<td width="15%">เบอร์โทรศัพท์</td>
<td width="15%">&nbsp;</td>
</tr>
<?php
    while ($row = mysql_fetch_assoc ($result)) {
        echo "<tr>";
        echo "<td>" . $row ["cus_id"] . "</td>";
        echo "<td>" . $row ["name"] . "</td>";
        echo "<td>" . $row ["address"] . "</td>";
        echo "<td>" . $row ["email"] . "</td>";
        echo "<td>" . $row ["tel"] . "</td>";
        printf("<td>[ <a href = 'customer_edit.php?cus_id=%s'=>แก้ไข</a> ]", $row
["cus_id"]);
        printf("[ <a href = 'customer_del.php?cus_id=%s'=>ลบ</a> ]</td>", $row
["cus_id"]);
        echo "</tr>";
    }
?>
</table>
<p>&nbsp;</p>
</body>
</html>
<?php
    mysql_free_result ($result);
?>

```

เชื่อมโยงไปยังหน้าแก้ไข
customer_edit.php
และสั่งรหัสลูกค้าไปด้วย



เชื่อมโยงไปยังหน้าลบ
customer_del.php
และสั่งรหัสลูกค้าไปด้วย



ภาพที่ 14.2 แสดงหน้าเว็บเพจรายงานข้อมูลลูกค้าทั้งหมด



14.5.2 การแก้ไขข้อมูล

การเปลี่ยนแปลงข้อมูลในระบบฐานข้อมูล MySQL ด้วยรูปแบบการแก้ไขข้อมูล หน้าแก้ไขข้อมูลลูกค้า (customer_edit.php) รายละเอียด ดังนี้

ตัวอย่างที่ 14.16 สคริปต์สำหรับแสดงข้อมูลลูกค้าก่อนการแก้ไข

```
<?php
    $host = "localhost";
    $user = "parinya";
    $password = "12345";
    $link = mysql_pconnect ($host, $user, $password);
    $db_selected = mysql_select_db ("company", $link);
    $query = sprintf ("SELECT * FROM customer WHERE cus_id='%s'", $_GET["cus_id"]);
    $result = mysql_query ($query, $link);
    if (!$result) die ("Can not Query because: " . mysql_error ( ));
    $row = mysql_fetch_assoc ($result);
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>แก้ไขข้อมูลลูกค้า</title>
</head>
<body>
<p>แก้ไขข้อมูลลูกค้าใหม่</p>
<form id="form1" name="form1" method="POST" action="edit.php">
    <table width="550" border="1">
        <tr>
            <td width="135">รหัสลูกค้า</td>
            <td width="249"><label>
                <input name="cus_id" type="text" id="cus_id" value="<?php echo $row["cus_id"];
?>" size="20" maxlength="5" />
            </label></td>
        </tr>
    </table>
</form>
```

\$_GET["cus_id"] เป็นตัวแปรผ่าน URL จะต้องใช้



สคริปต์สำหรับบันทึกข้อมูลที่แก้ไข



```

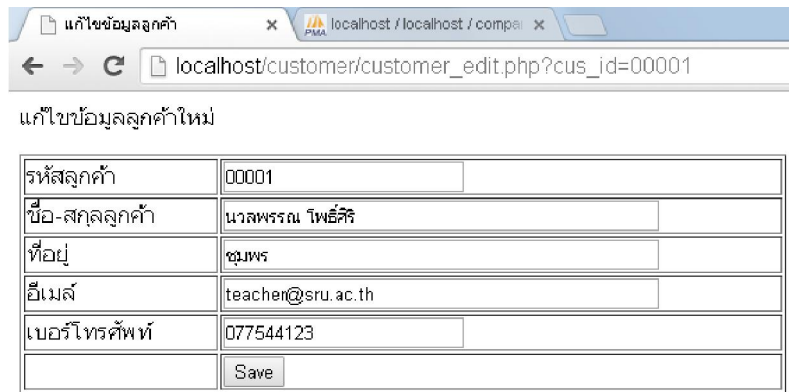
<td>ชื่อ-สกุลลูกค้า</td>
<td><label>
  <input name="name" type="text" id="name" value="<?php echo $row["name"]; ?>"
size="40" maxlength="40" />
</label></td>
</tr>
<tr>
<td>ที่อยู่</td>
<td><label>
  <input name="address" type="text" id="address" value="<?php echo
$row["address"]; ?>" size="40" maxlength="40" />
</label></td>
</tr>
<tr>
<td>อีเมล</td>
<td><label>
  <input name="email" type="text" id="email" value="<?php echo $row["email"]; ?>"
size="40" maxlength="40" />
</label></td>
</tr>
<tr>
<td>เบอร์โทรศัพท์</td>
<td><label>
  <input name="tel" type="text" id="tel" value="<?php echo $row["tel"]; ?>" size="20"
maxlength="20" />
</label></td>
</tr>
<tr>
<td>&nbsp;</td>
<td><label>
  <input type="submit" name="btnSend" id="btnSend" value="Save" />
</label></td>

```

```

</tr>
</table>
</form>
<p>&nbsp;</p>
</body>
</html>
<?php
    mysql_free_result ($result);
?>

```



ภาพที่ 14.3 หน้าเว็บเพจแสดงข้อมูลลูกค้าที่ต้องการแก้ไข

5) หน้าเว็บเพจสำหรับ UPDATE ข้อมูลลงฐานข้อมูล MySQL (edit.php) รายละเอียด ดังนี้

ตัวอย่างที่ 14.17 สคริปต์สำหรับแก้ไขข้อมูลลูกค้า

```

<?php
    ob_start();
    $host = "localhost";
    $user = "parinya";
    $password = "12345";
    $link = mysql_pconnect ($host, $user, $password);
    $db_selected = mysql_select_db ("company", $link);
    $query = sprintf ("UPDATE customer SET name='%s', address='%s', email='%s',
tel='%s' WHERE cus_id='%s';", $_POST["name"], $_POST["address"], $_POST["email"],
$_POST["tel"], $_POST["cus_id"]);

```

```
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ( ));
header('Location: report.php');
?>
```

14.5.3 การลบข้อมูล

การเปลี่ยนแปลงข้อมูลในระบบฐานข้อมูล MySQL ด้วยรูปแบบการลบข้อมูล

ตัวอย่างที่ 14.18 สคริปต์สำหรับลบข้อมูลลูกค้า

```
<?php
ob_start();
$host = "localhost";
$user = "parinya";
$password = "12345";
$link = mysql_pconnect ($host, $user, $password);
$db_selected = mysql_select_db ("company", $link);
$query = sprintf ("DELETE FROM customer WHERE cus_id='%s'", $_GET["cus_id"]);
$result = mysql_query ($query, $link);
if (!$result) die ("Can not Query because: " . mysql_error ( ));
header('Location: report.php');
?>
```

ส่งรหัสลูกค้ามาจาก
หน้ารายงานผ่าน URL



14.6 การแก้ไขข้อมูลภาษาไทยกลายเป็น "?????...."

สำหรับผู้ที่เกิดปัญหาการแสดงผลที่เป็นภาษาไทย แล้วกลายเป็น "???...." ให้ลองแก้ปัญหาโดยการกำหนดเซตอักขระใหม่ให้รองรับภาษาไทย ด้วยการส่งคำสั่งต่อไปนี้ก่อนที่จะอ่านข้อมูล

ตัวอย่างที่ 14.19 การแก้ไขข้อมูลภาษาไทยกลายเป็น "?????...."

```
mysql_query ("SET CHARACTER SET tis620");
//หรือใช้ mysql_query ("SET NAMES tis620");
```

14.7 การตรวจสอบการเปลี่ยนแปลงข้อมูล

การตรวจสอบการเปลี่ยนแปลงข้อมูล เป็นสิ่งที่ทำให้ผู้พัฒนาเว็บแอปพลิเคชันทราบถึงการเปลี่ยนแปลงต่างๆ ที่เกิดขึ้นครั้งล่าสุดกับระบบฐานข้อมูล MySQL เช่นการ INSERT, DELETE และ UPDATE เป็นต้น การตรวจสอบการเปลี่ยนแปลงที่เกิดขึ้นในกรณีต่างๆ ดังต่อไปนี้



14.7.1 ฟังก์ชัน mysql_affected_rows ()

เป็นฟังก์ชันที่ใช้สำหรับการนับจำนวนแถวที่เปลี่ยนแปลงข้อมูล โดยถ้าต้องการตรวจสอบว่ามีข้อมูลจำนวนกี่แถว ที่เกิดการเปลี่ยนแปลงจากการใช้คำสั่งสอบถามหรือคำสั่งข้อมูลครั้งสุดท้าย ก็ยังสามารถตรวจสอบได้โดยใช้ฟังก์ชัน mysql_affected_rows () มีรูปแบบดังนี้

รูปแบบ

```
int mysql_affected_rows ([ resource $link_identifier = NULL ] )
```

เมื่อ \$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.20 การใช้ฟังก์ชัน mysql_affected_rows ()

```

1  <?php
2      $link = mysql_connect ('localhost', 'mysql_user', 'mysql_password');
3      if (!$link) {
4          die ("ไม่สามารถเชื่อมต่อกับฐานข้อมูลได้ เนื่องจาก: " . mysql_error ());
5      }
6      mysql_select_db ('mydb');
7      mysql_query ('DELETE FROM mytable WHERE id < 10');
8      printf ("Records deleted: %d<br>", mysql_affected_rows ());
9  ?>

```

14.7.2 ฟังก์ชัน mysql_insert_id ()

เป็นฟังก์ชันที่ใช้สำหรับการตรวจสอบ ID ของข้อมูลที่เพิ่มใหม่ จากการที่ขอบเขตข้อมูลแบบ "AUTO_INCREMENT" นั้น ข้อมูลจะถูกเพิ่มต่อจากข้อมูลล่าสุดโดยอัตโนมัติ โดยส่วนใหญ่จะใช้ AUTO_INCREMENT กับข้อมูลที่เป็นค่า ID เนื่องจากจะมีลักษณะการเรียงลำดับที่ต่อเนื่อง และไม่ซ้ำกัน และหลังจากที่เพิ่มข้อมูลลงในตารางแล้ว อาจจำเป็นต้องนำค่า ID นั้นไปใช้งานอื่นต่อไป เช่น นำไปใช้กับการประมวลผลอื่นๆ ของ PHP หรือการนำไปเติมกับตารางอื่นๆ ที่เกี่ยวข้องกัน เป็นต้น นี่คือเหตุผลที่จำเป็นต้องทราบว่า แถวล่าสุดที่ถูกเพิ่มลงในตารางมีค่า ID เป็นเท่าไร โดย PHP มีฟังก์ชันในการตรวจสอบค่า ID นี้ได้คือ ฟังก์ชัน mysql_insert_id () โดยมีรูปแบบดังนี้

รูปแบบ

```
int mysql_insert_id ([ resource $link_identifier = NULL ] )
```

เมื่อ \$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ฟังก์ชันนี้จะคืนค่ากลับมาเป็นข้อมูลในขอบเขตข้อมูลที่เป็น AUTO_INCREMENT ของแถวล่าสุดที่ถูกเพิ่มเข้าไปด้วยคำสั่ง INSERT หรือ REPLACE เช่น สมมติจากเดิมข้อมูลในตารางแรก และขอบเขตข้อมูล id เป็นแบบ AUTO_INCREMENT



ตัวอย่างที่ 14.21 การใช้ฟังก์ชัน mysql_insert_id ()

```

1  <?php
2      $link = mysql_connect ('localhost', 'mysql_user', 'mysql_password');
3      if (!$link) {
4          die ('ไม่สามารถเชื่อมต่อกับฐานข้อมูล เนื่องจาก: ' . mysql_error ());
5      }
6      mysql_select_db (' mydb');
7      mysql_query ("INSERT INTO mytable (product) values ('kossu)");
8      printf ("Last inserted record has id %d<br>", mysql_insert_id ());
9  ?>

```

แต่อย่างไรก็ตาม ต้องเรียกใช้ฟังก์ชัน mysql_insert_id () ทันทีหลังจากเพิ่มข้อมูล ด้วยคำสั่ง INSERT หรือคำสั่ง REPLACE ทั้งนี้เพื่อให้ได้ค่า ID ของแถวล่าสุดที่ถูกเพิ่มเข้าใหม่อย่างแท้จริง

14.7.3 ฟังก์ชัน mysql_data_seek ()

เป็นฟังก์ชันที่ใช้สำหรับระบุตำแหน่งของแถวข้อมูลที่ต้องการให้เริ่มต้น นับแถวแรกเป็น 0 โดยมีรูปแบบดังนี้

รูปแบบ

```
bool mysql_data_seek ( resource $result , int $row_number )
```

เมื่อ \$result หมายถึง ตัวแปรที่ใช้เก็บผลลัพธ์ที่ได้จากฟังก์ชัน mysql_query ()

\$row_number หมายถึง ค่าของแถวที่ต้องการเริ่มต้น

ตัวอย่างที่ 14.22 การใช้ฟังก์ชัน mysql_data_seek ()

```

1  <?php
2      $link = mysql_pconnect ('localhost', 'mysql_user', 'mysql_password');
3      if (!$link) {
4          die ('ไม่สามารถเชื่อมต่อกับฐานข้อมูลได้ เนื่องจาก: ' . mysql_error());
5      }
6      $db_selected = mysql_select_db ('sample_db');
7      if (!$db_selected) {
8          die ('ไม่สามารถเลือกฐานข้อมูลได้ เนื่องจาก: ' . mysql_error());
9      }
10     $query = 'SELECT last_name, first_name FROM friends';
11     $result = mysql_query ($query);
12     if (!$result) {

```



```

13         die('การสอบถามข้อมูลเกิดข้อผิดพลาด เนื่องจาก: ' . mysql_error ( ));
14     }
15     /* เรียกดูข้อมูลในตำแหน่งลำดับแบบย้อนกลับ */
16     for ($i = mysql_num_rows ($result) - 1; $i >= 0; $i--) {
17         if (!mysql_data_seek ($result, $i)) {
18             echo "ไม่สามารถระบุตำแหน่งแถวข้อมูลแถวที่ $i: " . mysql_error ( ) . "<br/>";
19             continue;
20         }
21         if (!(($row = mysql_fetch_assoc ($result))) {
22             continue;
23         }
24         echo $row ["last_name"] . ' ' . $row ["first_name"] . "<br />\n";
25     }
26     mysql_free_result ($result);
27     ?>

```

14.8 การตรวจสอบข้อผิดพลาดระหว่างการทำงานร่วมกับ MySQL

เมื่อเกิดข้อผิดพลาดในการทำงาน MySQL ขึ้น หากต้องการทราบว่าข้อผิดพลาดนั้นมาจากสาเหตุอะไร ก็สามารถตรวจสอบได้โดยใช้ฟังก์ชัน `mysql_error ()` มีรูปแบบดังนี้

รูปแบบ

```
string mysql_error ([ resource $link_identifier = NULL ] )
```

เมื่อ `$link_identifier` หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ในทางปฏิบัติแล้ว มักจะใช้ฟังก์ชันนี้ร่วมกับขั้นตอนที่คาดว่า อาจเกิดข้อผิดพลาดขึ้น แล้วอ่านข้อผิดพลาดนั้นมาแสดง

14.9 การตรวจสอบชื่อฐานข้อมูลและตาราง

บางครั้งต้องการทราบว่า ภายในฐานข้อมูล MySQL นั้นประกอบด้วยฐานข้อมูล หรือตารางชื่ออะไรบ้าง ก็จะมีการตรวจสอบดังนี้

14.9.1 การตรวจสอบชื่อฐานข้อมูล

ชื่อฐานข้อมูลที่ใช้ตรวจสอบได้ต้องเป็นชื่อของฐานข้อมูลที่เป็นผู้สร้างขึ้นมา หรือมีสิทธิ์ในการใช้งาน โดยการตรวจสอบจะใช้ฟังก์ชัน `mysql_list_dbs ()` ฟังก์ชันนี้จะคืนค่ากลับคืนมาเป็น result set โดยที่ข้อมูลในแต่ละแถวก็คือ ชื่อฐานข้อมูลที่สามารถอ่านได้ มีรูปแบบดังนี้



รูปแบบ

```
resource mysql_list_dbs ([ resource $link_identifier = NULL ] )
```

เมื่อ `$link_identifier` หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier
จากการที่ฟังก์ชันนี้คืนมาเป็น result set หากต้องการจะอ่านรายชื่อฐานข้อมูลเหล่านี้ จะต้องใช้ฟังก์ชันที่ใช้ในการอ่านข้อมูลจาก result set เช่น `mysql_fetch_row ()` หรือ `mysql_fetch_array ()` หรือ `mysql_fetch_object ()` หรือ `mysql_fetch_assoc ()` เป็นต้น มีตัวอย่างดังนี้

ตัวอย่างที่ 14.23 การตรวจสอบชื่อฐานข้อมูล

```
1 <?php
2 // ตัวอย่างการใช้งานแบบทั่วไปถ้าไปใช้ฟังก์ชัน mysql_list_dbs ( )
3 $link = mysql_pconnect ('localhost', 'mysql_user', 'mysql_password');
4 $res = mysql_query ("SHOW DATABASES");
5 while ($row = mysql_fetch_assoc ($res)) {
6     echo $row ['Database'] . "<br/>";
7 }
8 // ตัวอย่างการใช้งานแบบทั่วไปถ้าไปใช้ฟังก์ชัน mysql_list_dbs ( ) สำหรับ PHP 5.4.0
9 $link = mysql_pconnect ('localhost', 'mysql_user', 'mysql_password');
10 $db_list = mysql_list_dbs ($link);
11 while ($row = mysql_fetch_object ($db_list)) {
12     echo $row->Database . "<br/>";
13 }
14 ?>
```

14.9.2 การตรวจสอบชื่อตารางที่มีอยู่ในฐานข้อมูล

ถ้าต้องการทราบว่า ภายในฐานข้อมูลนั้นมีตารางชื่ออะไรบ้าง ก็สามารถตรวจสอบด้วยฟังก์ชัน `mysql_list_tables ()` มีรูปแบบดังนี้

รูปแบบ

```
resource mysql_list_tables ( string $database [, resource $link_identifier = NULL ] )
```

เมื่อ `$database` หมายถึง ชื่อของฐานข้อมูล ต้องกำหนดในแบบข้อความผลลัพธ์ที่ได้จากฟังก์ชันนี้จะอยู่ในรูปแบบของ result set
`$link_identifier` หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.24 การตรวจสอบชื่อตารางที่มีอยู่ในฐานข้อมูล

```

1  <?php
2      $dbname = 'mysql_dbname';
3      if (!mysql_pconnect ('mysql_host', 'mysql_user', 'mysql_password')) {
4          echo 'ไม่สามารถเชื่อมต่อกับฐานข้อมูล MySQL ได้ เนื่องจาก:' . mysql_error ( );
5          exit;
6      }
7      $sql = "SHOW TABLES FROM $dbname";
8      $result = mysql_query ($sql);
9      if (!$result) {
10         echo "เกิดข้อผิดพลาด ไม่พบชื่อตารางในฐานข้อมูล $dbname ";
11         exit;
12     }
13     while ($row = mysql_fetch_row ($result)) {
14         echo "Table: {$row [0]} <br/>";
15     }
16     mysql_free_result ($result);
17     ?>

```

14.10 การตรวจสอบโครงสร้างของขอบเขตข้อมูล

การตรวจสอบโครงสร้างของขอบเขตข้อมูลเป็นการอ่านรายละเอียดทั้งหมดเกี่ยวกับขอบเขตข้อมูล ชนิดข้อมูล และแอททริบิวต์ต่างๆ สำหรับใน PHP มีฟังก์ชันที่ใช้ในการตรวจสอบองค์ประกอบของขอบเขตข้อมูล ดังต่อไปนี้

14.10.1 การนับจำนวนขอบเขตข้อมูล

การนับจำนวนฟิลด์จะนับจำนวนจากขอบเขตข้อมูลที่มีอยู่ใน result set ที่กำหนด เนื่องจากเป็นการนับจำนวนขอบเขตข้อมูลที่มีอยู่ใน \$result ที่ได้จากการใช้ฟังก์ชัน mysql_query () กำหนดให้อ่านข้อมูลจากขอบเขตข้อมูลใดบ้าง ดังนั้นจำนวนขอบเขตข้อมูลที่นับได้ อาจแตกต่างจำนวนจริงที่มีอยู่ในตารางมีรูปแบบดังนี้

รูปแบบ

```
int mysql_num_fields ( resource $result )
```

เมื่อ \$result หมายถึง ผลลัพธ์ที่ได้จากฟังก์ชัน mysql_query ()



ตัวอย่างที่ 14.25 การนับจำนวนขอบเขตข้อมูล

```

1  <?php
2      $result = mysql_query ("SELECT id, email FROM people WHERE id = '42'");
3      if (!$result) {
4          echo 'ไม่สามารถสอบถามข้อมูลได้ เนื่องจาก: ' . mysql_error ();
5          exit;
6      }
7      echo mysql_num_fields ($result);
8  ?>

```

14.10.2 การตรวจสอบชื่อขอบเขตข้อมูล

การตรวจสอบชื่อขอบเขตข้อมูลในตาราง จะมีฟังก์ชันดังนี้

1) ฟังก์ชัน mysql_list_fields ()

เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบชื่อขอบเขตข้อมูลในตารางของฐานข้อมูลที่กำหนด มีรูปแบบการใช้งานดังนี้

รูปแบบ

```

resource mysql_list_fields ( string $database_name , string $table_name
    [, resource $link_identifier = NULL ] )

```

เมื่อ \$database_name หมายถึง ชื่อฐานข้อมูล

\$table_name หมายถึง ชื่อตาราง

\$link_identifier หมายถึง ตัวแปรที่ใช้เก็บค่า MySQL link identifier

ตัวอย่างที่ 14.26 การใช้ฟังก์ชัน mysql_list_fields ()

```

1  <?php
2      $result = mysql_query ("SHOW COLUMNS FROM sometable");
3      if (!$result) {
4          echo 'ไม่สามารถสอบถามข้อมูลได้ เนื่องจาก: ' . mysql_error();
5          exit;
6      }
7      if (mysql_list_fields ($result) > 0) {
8          while ($row = mysql_fetch_assoc ($result)) {
9              print_r ($row);
10             }
11     }

```

12 ?>

2) ฟังก์ชัน mysql_field_name ()

เป็นฟังก์ชันที่ใช้สำหรับตรวจสอบชื่อขอบเขตข้อมูลในตารางของฐานข้อมูลที่กำหนด เช่นเดียวกับฟังก์ชัน mysql_list_fields () มีรูปแบบการใช้งานดังนี้

รูปแบบ

string mysql_field_name (resource \$result)

เมื่อ \$result หมายถึง ผลลัพธ์ที่ได้จากฟังก์ชัน mysql_query ()

ตัวอย่างที่ 14.27 การใช้ฟังก์ชัน mysql_field_name ()

```

1 <?php /* ตัวอย่าง ตารางประกอบด้วย 3 ขอบเขตข้อมูล ดังนี้:
2     * user_id , username และ password */
3     $link = mysql_pconnect ('localhost', 'mysql_user', 'mysql_password');
4     if (!$link) {
5         die ('ไม่สามารถเชื่อมต่อกับ MySQL server เนื่องจาก: ' . mysql_error ( ));
6     }
7     $dbname = 'mydb';
8     $db_selected = mysql_select_db ($dbname, $link);
9     if (!$db_selected) {
10        die ("ไม่สามารถเชื่อมต่อกับฐานข้อมูล $dbname ได้เนื่องจาก: " . mysql_error ( ));
11    }
12    $res = mysql_query ('select * from users', $link);
13    echo mysql_field_name ($res, 0) . "<br/>";
14    echo mysql_field_name ($res, 2);
15 ?>
```

ผลลัพธ์

user_id
password

14.10.3 การตรวจสอบความยาวของขอบเขตข้อมูล

การตรวจสอบความยาวของขอบเขตข้อมูลจะใช้ฟังก์ชัน mysql_fields_len () มีรูปแบบดังนี้

รูปแบบ

int mysql_field_len (resource \$result)

เมื่อ \$result หมายถึง ผลลัพธ์ที่ได้จากฟังก์ชัน mysql_query ()

ตัวอย่างที่ 14.28 การตรวจสอบความยาวของขอบเขตข้อมูล

```

1 <?php
2     $result = mysql_query ("SELECT id,email FROM people WHERE id = '42'");
3     if (!$result) {
4         echo "ไม่สามารถสอบถามข้อมูลได้ เนื่องจาก: " . mysql_error ( ); exit;
5     }
6     $length = mysql_field_len ($result, 0); // ตรวจสอบความยาวของขอบเขตข้อมูล "id"
7     echo $length;
8     ?>

```

14.10.4 การตรวจสอบชนิดข้อมูลของขอบเขตข้อมูล

การตรวจสอบชนิดข้อมูลของขอบเขตข้อมูลด้วยฟังก์ชัน `mysql_field_type ()` มีรูปแบบดังนี้

รูปแบบ

```
string mysql_field_type ( resource $result )
```

เมื่อ `$result` หมายถึง ผลลัพธ์ที่ได้จากฟังก์ชัน `mysql_query ()`

ตัวอย่างที่ 14.29 การตรวจสอบชนิดข้อมูลของขอบเขตข้อมูล

```

<?php
mysql_pconnect ("localhost", "mysql_username", "mysql_password");
mysql_select_db ("mysql");
$result = mysql_query ("SELECT * FROM func");
$fields = mysql_num_fields ($result);
$rows = mysql_num_rows ($result);
$table = mysql_field_table ($result, 0);
echo "ตาราง " . $table . " มี " . $fields . " ขอบเขตข้อมูล จำนวน " . $rows . " ระเบียบ\nbr/>";
echo "โดยตารางประกอบด้วยขอบเขตข้อมูลดังนี้:<br/>";
for ($i=0; $i < $fields; $i++) {
    $type = mysql_field_type ($result, $i);
    $name = mysql_field_name ($result, $i);
    $len = mysql_field_len ($result, $i);
    $flags = mysql_field_flags ($result, $i);
    echo $type . " " . $name . " " . $len . " " . $flags . "\nbr/>";
}

```

```
mysql_free_result ($result);
?>
```

ผลลัพธ์

```
ตาราง 'func' มี 4 ขอบเขตข้อมูล จำนวน 1 ระเบียบน
โดยตารางประกอบด้วยขอบเขตข้อมูลดังนี้:
string name 64 not_null primary_key binary
int ret 1 not_null
string dl 128 not_null
string type 9 not_null enum
```

สรุป

ในการเขียนสคริปต์ PHP สำหรับการติดต่อกับฐานข้อมูล MySQL นั้น จะสังเกตได้ว่า การเริ่มต้นการเขียนสคริปต์นั้น มีขั้นตอนสำคัญ คือ 1) เริ่มจากการติดต่อกับฐานข้อมูล 2) เลือกฐานข้อมูลที่จะใช้งาน และ 3) การส่งคำสั่งสอบถามหรือคำสั่งไปยังระบบฐานข้อมูล MySQL เพื่อประมวลผลตามคำสั่งสอบถามหรือคำสั่งที่ส่งเข้าไป ซึ่งอาจเป็นการสอบถามเพื่อเรียกดูข้อมูล หรือคำสั่งเพื่อแทรกข้อมูล แก้ไขข้อมูล และลบข้อมูล เป็นต้น นี่คือลักษณะพื้นฐานของการพัฒนาเว็บแอปพลิเคชันที่มีการติดต่อกับฐานข้อมูล จากตัวอย่างทั้งหมดที่อธิบายทั้งหมดจะสังเกตได้ว่าคำสั่งช่วงบรรทัดบนสุดของแต่ละสคริปต์จะมีการเปิดการเชื่อมต่อ เลือกใช้ฐานข้อมูล มีรูปแบบซ้ำๆ ในทุกหน้าเว็บเพจ ดังนั้น ผู้พัฒนาเว็บแอปพลิเคชันสามารถเขียนสคริปต์แยกเป็นไฟล์เฉพาะเก็บไว้ และเมื่อต้องการใช้งานในหน้าเว็บเพจใดสามารถใช่วิธีการนำเข้าไฟล์เพื่อประมวลผล จะทำให้ช่วยลดบรรทัดสำหรับการเขียนสคริปต์ และสะดวกในการใช้งาน

คำถามท้ายบท

- จงอธิบายหน้าที่ รูปแบบ และความแตกต่าง ระหว่างฟังก์ชัน `mysql_connect ()` และ `mysql_pconnect ()`
- จงอธิบายหน้าที่ของฟังก์ชัน `mysql_query ()` พร้อมยกตัวอย่างสำหรับเลือกใช้งานฐานข้อมูลให้ทำหน้าที่เหมือนกับฟังก์ชัน `mysql_select_db ()`

ชื่อตาราง employees (พนักงาน) ใช้ประกอบการตอบคำถามข้อที่ 3 -5

emp_id	firstname	lastname	position	salary	address	email	phone
100	Parinya	Noidonprai	Manager	45000	Suratthani	pa@sru.ac.th	077355466
200	Jaidee	Ruktum	Programmer	17000	Suratthani	ja@sru.ac.th	077355123
300	Dennis	Robert	Programmer	19000	Suratthani	ha@sru.ac.th	077355443
400	John	Mackay	Finance	18000	Chumphon	jo@sru.ac.th	077355889



3. จงอธิบายหลักการทำงานของสคริปต์ และผลลัพธ์ที่ได้ จากสคริปต์ต่อไปนี้

```

1  <?php
2      $link = mysql_pconnect ("localhost", "username", "password");
3      $db_selected = mysql_select_db ("company", $link);
4      $query = "SELECT firstname, position, address FROM employees WHERE
position=" . "Programmer";
5      $result = mysql_query ($query, $link);
6      if (!$result) die ("Can not Query because: " . mysql_error ( ));
7          echo "<table border=1 width=50%>";
8      while ($row = mysql_fetch_assoc ($result)) {
9          echo "<tr>";
10         echo "<td> " . $row ["firstname"] . "</td>";
11         echo "<td> " . $row ["position"] . "</td>";
12         echo "<td> " . $row ["address"] . "</td>";
13         echo "</tr>";
14     }
15     echo "<table>";
16     mysql_free_result ($result);
17  ?>

```

4. สคริปต์จากข้อที่ 3. หากลบบรรทัดที่ 7 - 15 ออก แล้วเพิ่มฟังก์ชัน echo mysql_affected_rows (\$link) จะได้ผลลัพธ์อย่างไร และอธิบายเหตุผลประกอบ

5. จงเขียนสคริปต์ PHP เพื่อสำหรับเรียกดูขอบเขตข้อมูลพนักงานทั้งหมด โดยมีเงื่อนไขว่าดูข้อมูลเฉพาะพนักงานที่มีช่วงเงินเดือนระหว่าง 18000 – 30000 บาทเท่านั้น