

คู่มือการใช้งาน phpMyAdmin

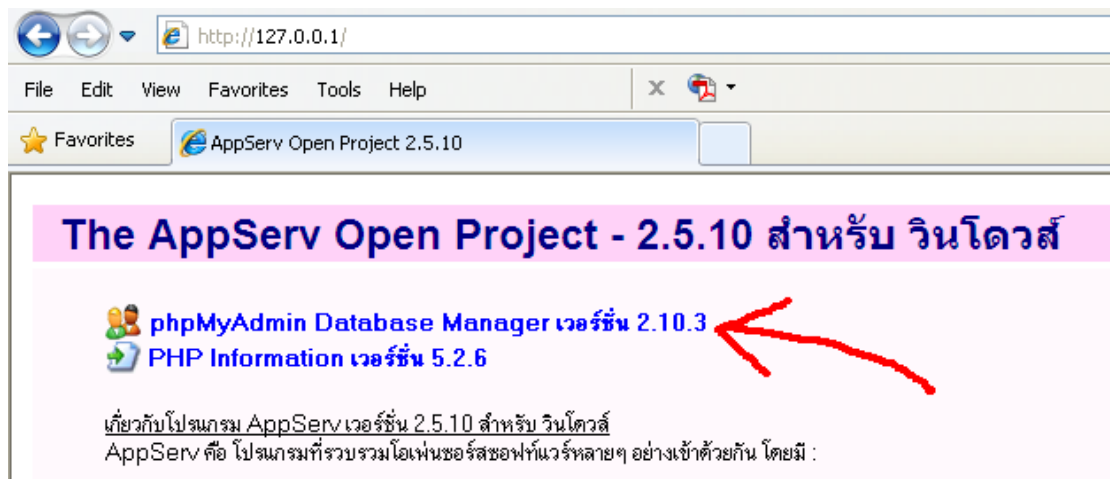
phpMyAdmin เป็นโปรแกรมตัวหนึ่งที่ใช้ในการจัดการฐานข้อมูล MySQL ผ่านบราวเซอร์ โดยโปรแกรมนี้เป็นโปรแกรมที่ใช้งานได้ฟรี เพราะเป็นโปรแกรมในกลุ่ม Open Source

phpMyAdmin จะทำงานเพื่อใช้ควบคุมจัดการฐานข้อมูล MySQL ความสามารถของ phpMyAdmin มีดังนี้

1. สร้างและลบ Database
2. สร้างและจัดการ Table เช่น เพิ่ม record, ลบ record, แก้ไข record หรือลบ Table, เพิ่มหรือแก้ไข field ในตาราง
3. โหลดเท็กซ์ไฟล์ และ ไฟล์นามสกุล CSV เข้าไปเก็บเป็นข้อมูลใน Table ได้
4. สามารถใช้คำสั่ง SQL ได้

การเข้าใช้งาน

การเข้าใช้งานโปรแกรม phpMyAdmin จะใช้งานผ่านโปรแกรม AppServ ดังนั้นคุณต้องติดตั้งโปรแกรม AppServ ก่อน แล้วเปิด Internet Explorer พิมพ์ที่ URL ว่า 127.0.0.1 แล้ว enter เมื่อจอภาพแสดงโปรแกรม AppServ แล้วให้คลิกที่ phpMyAdmin Database Manager

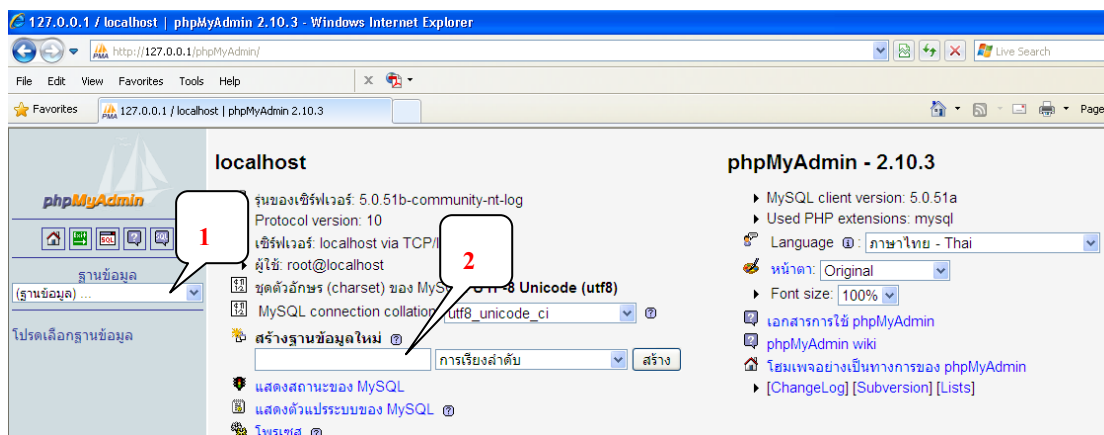


หลักจากนั้นจอภาพอาจจะให้ป้อน user name และ password

user name คือ root

password ซึ่งขึ้นอยู่กับเวลาที่ติดตั้ง AppServ ได้ป้อน password ของ MySQL ไว้อย่างไร

หน้าจอแรกของโปรแกรม phpMyAdmin



รูปที่ 1 แสดงหน้าจอแรกของโปรแกรม phpmyadmin

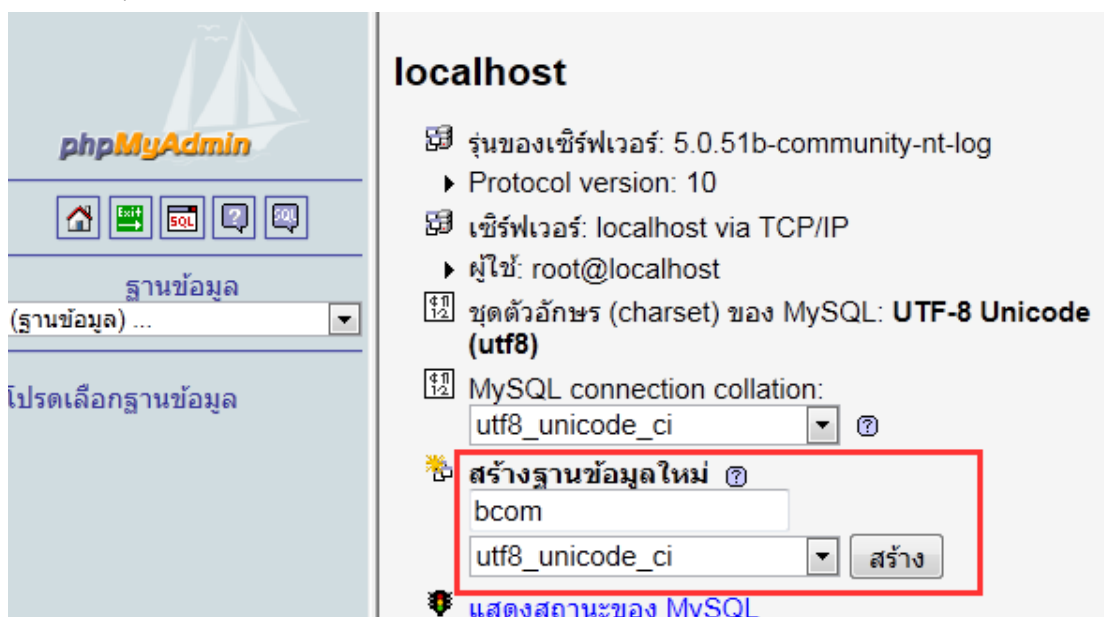
1. ใช้สำหรับ เลือก/เปลี่ยน จัดการกับฐานข้อมูลเดิมที่มีอยู่แล้ว
2. สำหรับใส่ชื่อฐานข้อมูลใหม่ที่ต้องการสร้าง

การสร้างฐานข้อมูลใหม่

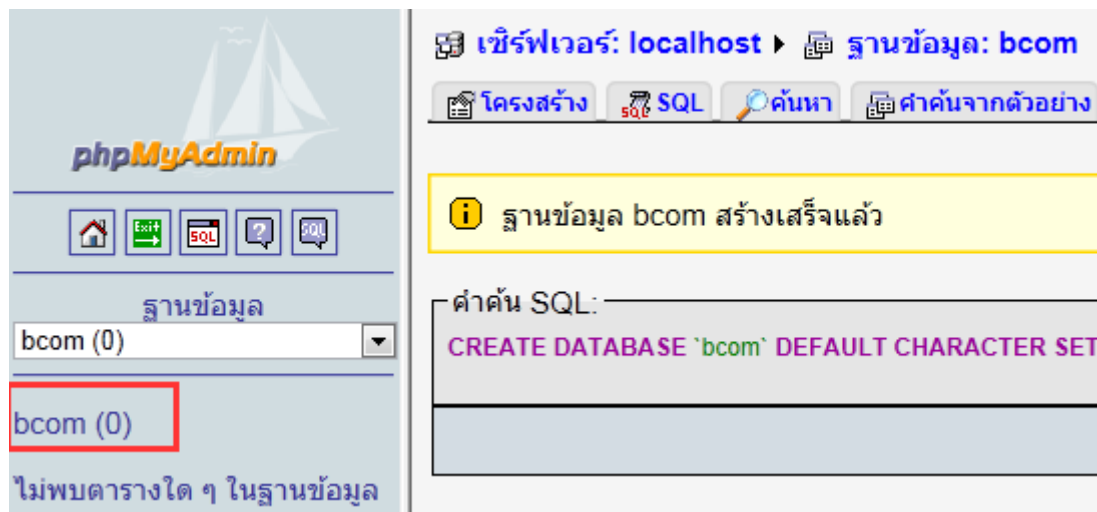
สมมติว่าต้องการสร้างฐานข้อมูลชื่อ bcom

และให้ encode เป็น utf8_unicode_ci

แล้วคลิกปุ่ม สร้าง



รูปที่ 2 สร้างฐานข้อมูลชื่อ bcom



รูปที่ 3 แสดงหน้าจอเมื่อสร้าง database ชื่อ bcom

การสร้างตาราง (Table)

ภายหลังจากที่สร้างฐานข้อมูลเรียบร้อยแล้ว phpMyAdmin จะแจ้งให้เราทราบว่า ได้ทำการสร้างฐานข้อมูล bcom3209 เรียบร้อยแล้ว ซึ่งยังไม่มีเทเบิลใดๆ อยู่เลย ในที่นี่จะสร้างตารางสำหรับเก็บข้อมูลของบุคคล โดยการออกแบบให้มีเทเบิล 4 เทเบิล คือ

division, employee, province, users

division จะใช้เก็บข้อมูลหน่วยงานของพนักงาน

employee จะใช้เก็บข้อมูลพนักงาน

province จะใช้เก็บข้อมูลจังหวัด

users จะใช้เก็บข้อมูล ผู้ใช้โปรแกรม

<p>เทเบิล division มีโครงสร้างดังนี้</p> <table border="1"> <thead> <tr> <th>ฟิลด์</th> <th>ชนิด</th> </tr> </thead> <tbody> <tr> <td><u>id</u></td> <td>int(2)</td> </tr> <tr> <td>name_div</td> <td>varchar(40)</td> </tr> </tbody> </table>	ฟิลด์	ชนิด	<u>id</u>	int(2)	name_div	varchar(40)	<p>เทเบิล users มีโครงสร้างดังนี้</p> <table border="1"> <thead> <tr> <th>ฟิลด์</th> <th>ชนิด</th> </tr> </thead> <tbody> <tr> <td><u>id</u></td> <td>int(3)</td> </tr> <tr> <td>username</td> <td>varchar(60)</td> </tr> <tr> <td>password</td> <td>varchar(60)</td> </tr> <tr> <td>nameis</td> <td>varchar(40)</td> </tr> <tr> <td>flag</td> <td>varchar(1)</td> </tr> </tbody> </table>	ฟิลด์	ชนิด	<u>id</u>	int(3)	username	varchar(60)	password	varchar(60)	nameis	varchar(40)	flag	varchar(1)								
ฟิลด์	ชนิด																										
<u>id</u>	int(2)																										
name_div	varchar(40)																										
ฟิลด์	ชนิด																										
<u>id</u>	int(3)																										
username	varchar(60)																										
password	varchar(60)																										
nameis	varchar(40)																										
flag	varchar(1)																										
<p>เทเบิล employee มีโครงสร้างดังนี้</p> <table border="1"> <thead> <tr> <th>ฟิลด์</th> <th>ชนิด</th> </tr> </thead> <tbody> <tr> <td><u>id</u></td> <td>int(3)</td> </tr> <tr> <td>sex</td> <td>varchar(10)</td> </tr> <tr> <td>name_emp</td> <td>varchar(40)</td> </tr> <tr> <td>emp_picture</td> <td>varchar(40)</td> </tr> <tr> <td>salary</td> <td>int(7)</td> </tr> <tr> <td>start_date</td> <td>date</td> </tr> <tr> <td>division_id</td> <td>int(2)</td> </tr> <tr> <td>province_id</td> <td>int(2)</td> </tr> <tr> <td>flag</td> <td>varchar(1)</td> </tr> </tbody> </table>	ฟิลด์	ชนิด	<u>id</u>	int(3)	sex	varchar(10)	name_emp	varchar(40)	emp_picture	varchar(40)	salary	int(7)	start_date	date	division_id	int(2)	province_id	int(2)	flag	varchar(1)	<p>เทเบิล province มีโครงสร้างดังนี้</p> <table border="1"> <thead> <tr> <th>ฟิลด์</th> <th>ชนิด</th> </tr> </thead> <tbody> <tr> <td><u>id</u></td> <td>int(11)</td> </tr> <tr> <td>nameis</td> <td>varchar(90)</td> </tr> </tbody> </table>	ฟิลด์	ชนิด	<u>id</u>	int(11)	nameis	varchar(90)
ฟิลด์	ชนิด																										
<u>id</u>	int(3)																										
sex	varchar(10)																										
name_emp	varchar(40)																										
emp_picture	varchar(40)																										
salary	int(7)																										
start_date	date																										
division_id	int(2)																										
province_id	int(2)																										
flag	varchar(1)																										
ฟิลด์	ชนิด																										
<u>id</u>	int(11)																										
nameis	varchar(90)																										

ตัวอย่างการสร้างเทเบิล division

The screenshot shows the phpMyAdmin interface for a database named 'bcom'. The 'CREATE TABLE' wizard is active, and the table name 'division' is entered in the 'ชื่อ' (Name) field. The 'Number of fields' is set to 2. The 'ลงมือ' (Go) button is highlighted with a red box. The interface also shows the SQL command: `CREATE DATABASE `bcom` DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;`

1. ให้คลิกที่ bcom เพื่อเลือกฐานข้อมูลชื่อนี้
2. ที่ช่องชื่อ ให้ป้อน division และ ป้อน 2 ในช่อง number of fields เพราะเทเบิล division จะมีฟิลด์อยู่ 2 ฟิลด์ที่ต้องการสร้างนั่นเอง แล้วให้**คลิกปุ่ม ลงมือ**
3. ป้อนข้อมูลฟิลด์ เหมือนดังรูป

เซิร์ฟเวอร์: localhost ▶ ฐานข้อมูล: bcom ▶ ตาราง : division

ฟิลด์	ชนิด	ความยาว/เขต *1	การเรียงลำดับ
id	INT	2	
name_div	VARCHAR	40	

ทางด้านขวา ของฟิลด์ id ให้เลือก เหมือนรูปด้านล่าง ซึ่งจะหมายถึงให้ฟิลด์ id เป็น primary key ของเทเบิล division

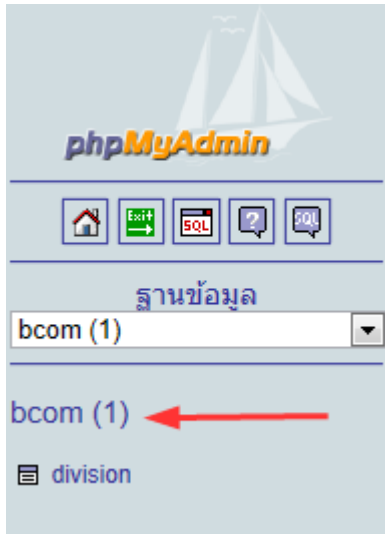
คำปรียาย ²	เพิ่มเติม					
<input type="text"/>	<input type="text"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/>	<input type="text"/>

เสร็จแล้วให้**คลิกปุ่ม บันทึก**

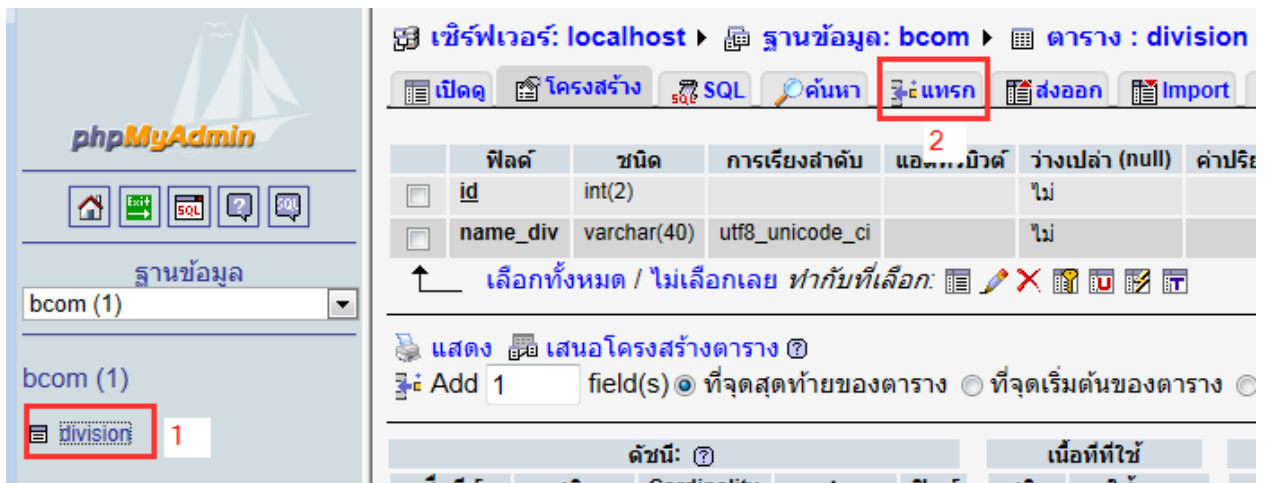
ไอคอนมีความหมายดังนี้

- สำหรับแก้ไข ชื่อฟิลด์ ชนิดของข้อมูลที่เก็บ ไพรมารีคีย์ หรือค่าเริ่มต้นอื่นๆ
- สำหรับลบฟิลด์นั้นๆ ออกจากตาราง
- สำหรับกำหนดให้ฟิลด์นั้นๆ เป็นไพรมารีคีย์ (Primary Key)
- สำหรับกำหนดให้ฟิลด์นั้นๆ เป็นเอกลักษณ์ (Unique)
- สำหรับกำหนดให้ฟิลด์นั้นๆ เป็นดัชนี (Index)

4. เสร็จสิ้นขั้นตอนการสร้างเทเบิล division จะเห็นว่า ที่ ดาต้าเบส bcom มีตัวเลข 1 ซึ่งหมายถึงมีเทเบิลอยู่ 1 เทเบิล แล้ว



5. ขั้นตอนต่อมาเป็นการเพิ่มข้อมูล ให้กับเทเบิล division



5.1 ให้คลิกที่เทเบิล division

5.2 คลิกแถบ แทรก

5.3 ให้ป้อนข้อมูลหน่วยงาน 1 ผู้จัดการ และ 2 บัญชี เหมือนรูปด้านล่าง

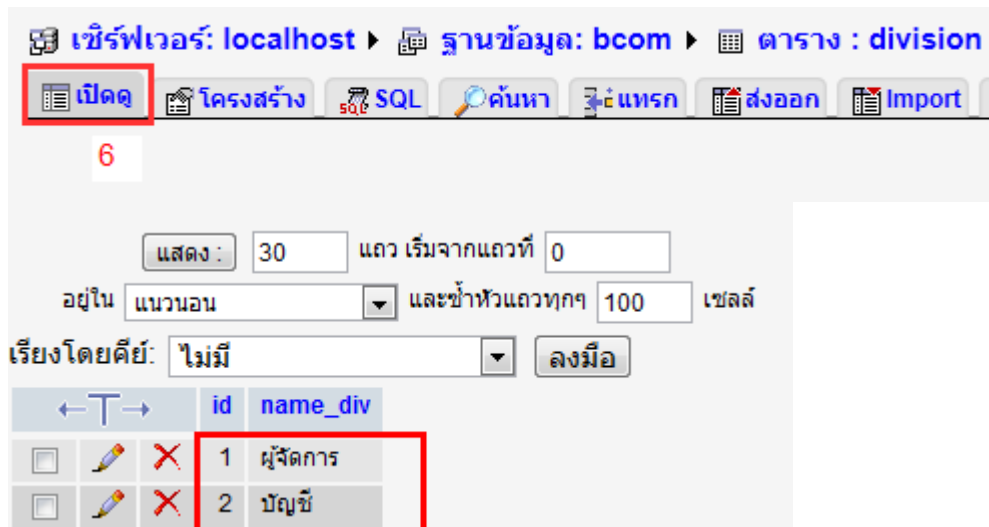
Screenshot of a database management tool showing an insert form for a table named 'division'. The form has two rows. The first row has 'id' set to 1 and 'name_div' set to 'ผู้จัดการ'. The second row has 'id' set to 2 and 'name_div' set to 'บัญชี'. A red box highlights these two rows. Below the form, there are options for 'แทรกเป็นแถวใหม่' (Insert as new row) and 'and then' (and then) with a dropdown set to 'ส่งกลับ' (Return). A red box highlights the 'ลงมือ' (Execute) button, which is labeled with a '4'.

5.4 ให้คลิกปุ่ม ลงมือ

5.5 โปรแกรมจะแสดงข้อความว่าแถวที่ถูกแทรก 2 หมายถึงเพิ่มข้อมูล 2 เรคอร์ด ในเทเบิล division เรียบร้อยแล้ว

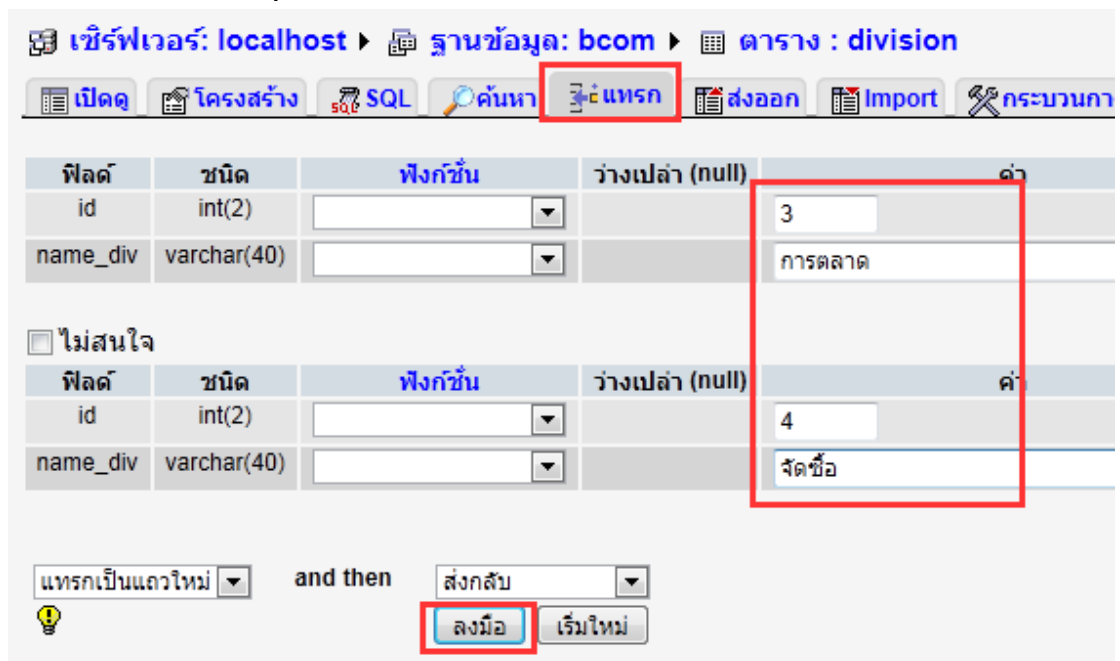
Screenshot of the database management tool showing a message box that says 'แถวที่ถูกแทรก: 2' (Rows affected: 2). Below the message box, the SQL command is displayed: 'INSERT INTO `bcom`.`division` (`id`,`name_div`) VALUES ('1','ผู้จัดการ'), ('2','บัญชี');'. A red box highlights the SQL command, and a '5' is placed next to it.

5.6 ถ้าต้องการดูว่ามีข้อมูล 2 เรคอร์ด จริงหรือไม่ ให้คลิกที่แถบเปิดดู











จะเห็นข้อมูล 2 แถว ซึ่งหมายถึง 2 เรคอร์ด ที่เราใช้โปรแกรม phpmyadmin

6. ให้สร้างข้อมูลเรคอร์ดเพิ่ม สำหรับเทเบิล division โดยคลิกแถบแทรก แล้วป้อนข้อมูล ดังรูปด้านล่าง แล้วคลิกปุ่ม **ลงมือ**



เมื่อต้องการดูข้อมูล ให้คลิกแถบเปิดดู จะเห็นข้อมูลในเทเบิล division มี 4เรคอร์ด ซึ่งเราสร้างเสร็จสมบูรณ์แล้ว

			id	name_div
<input type="checkbox"/>			1	ผู้จัดการ
<input type="checkbox"/>			2	บัญชี
<input type="checkbox"/>			3	การตลาด
<input type="checkbox"/>			4	จัดซื้อ

ขั้นตอนต่อมาให้เพิ่มข้อมูลลักษณะเดียวกับ โดยเพิ่มข้อมูลของเทเบิล employee, province, users โดยสร้างข้อมูล ตามตารางด้านล่างนี้

ข้อมูลของเทเบิล employee

id	sex	name_emp	emp_picture	salary	start_date	division_id	province_id	flag
1	ชาย	สุริยะ วรรณดี	1.jpg	10000	2011-10-29	2	14	
2	ชาย	อภิรักษ์ ดีมาก	2.jpg	15000	2010-02-10	3	2	
3	ชาย	เดช วังไชย	3.jpg	25000	2009-06-20	1	12	
4	หญิง	สมใจ ดาพร	4.jpg	9900	2009-12-11	2	12	
5	หญิง	ชนิดาพร สกุลดี	5.jpg	10000	2010-04-20	4	3	
6	ชาย	สุพจน์ วงศ์สา	6.jpg	14300	2009-03-10	2	15	1

ข้อมูลของเทเบิล users

id	username	password	nameis	flag
1	a1	a1	สุริยะ วรรณดี	1
2	a2	a2	อภิรักษ์ ดีมาก	
3	a3	a3	เดช วังไชย	

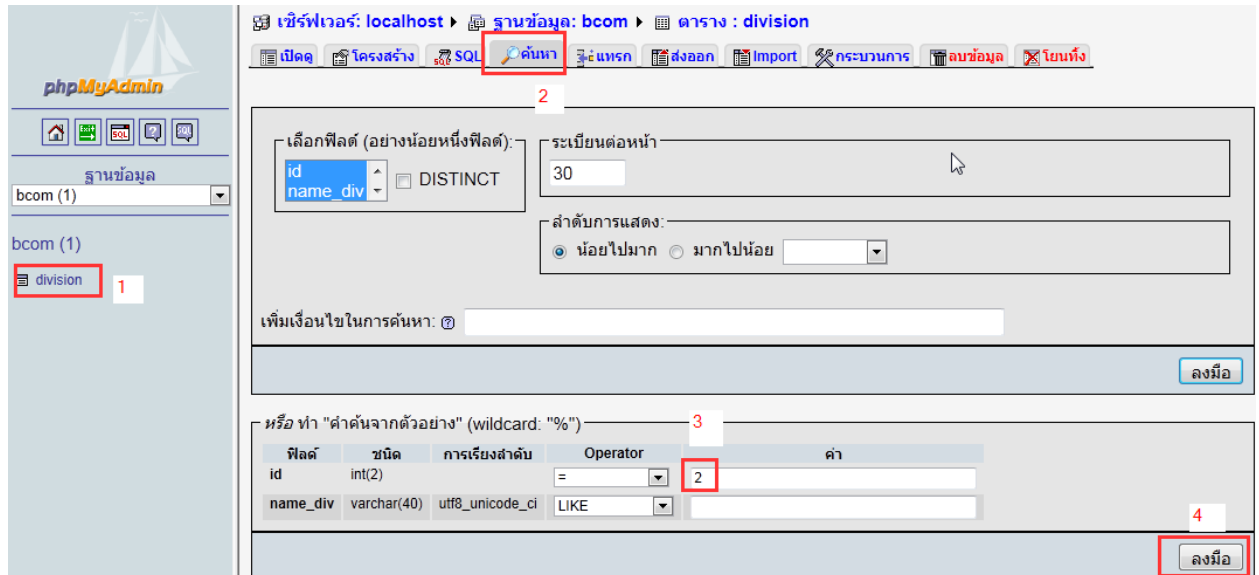
ข้อมูลของเทเบิล province

id	nameis
1	กรุงเทพ 11 ชัยภูมิ
2	กระบี่ 12 ชุมพร
3	กาญจนบุรี 13 เชียงราย
4	กาฬสินธุ์ 14 เชียงใหม่
5	กำแพงเพชร 15 ตรัง
6	ขอนแก่น 16 ตราด
7	จันทบุรี 17 ตาก
8	ฉะเชิงเทรา 18 นครนายก
9	ชลบุรี 19 นครปฐม
10	ชัยนาท 20 นครพนม

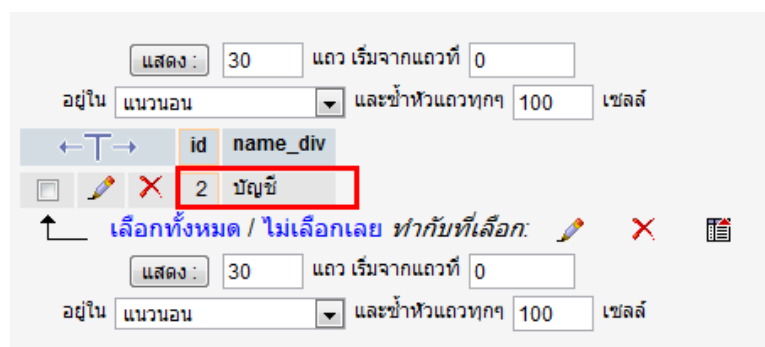
การใช้คำสั่ง SQL ในโปรแกรม PhpMyAdmin

1. การค้นหา

การค้นหาภายในแถบค้นหา ทำให้ง่ายต่อผู้ใช้ที่ต้องการความสะดวก และง่ายต่อการใช้งาน สำหรับหน้าจอของเมนู การค้นหา แสดงดังรูปด้านล่าง



1. เลือกเทเบิล division
2. คลิกแถบ ค้นหา
3. ช่อง ค่า จะให้ป้อนข้อความที่ต้องการค้นหา เช่น ป้อนเลข 2
4. คลิกปุ่ม ลงมือ



ผลลัพธ์ที่เห็นคือ หน่วยงานบัญชี เพราะเราค้นหาด้วยรหัส id ที่เท่ากับเลข 2

2. การแก้ไขข้อมูล

สมมติว่าต้องการแก้ไข ข้อมูลของเทเบิล division

1. คลิกที่เทเบิล division
2. คลิกแถบ เปิดดู
3. ที่ส่วนหน้าของแต่ละเรคคอร์ด จะมีรูปดินสอ ซึ่งหมายถึงการแก้ไข เรคคอร์ด นั้นๆ

เช่น ต้องการแก้ไขข้อมูลของหน่วยงาน ผู้จัดการ

ให้คลิกที่รูปดินสอ แถวที่ id เท่ากับ 1 แล้วจะเห็นจอภาพ ให้แก้ไขข้อมูลได้ ตามรูปด้านล่างนี้

เมื่อต้องการ save ให้คลิกปุ่ม ลงมือ

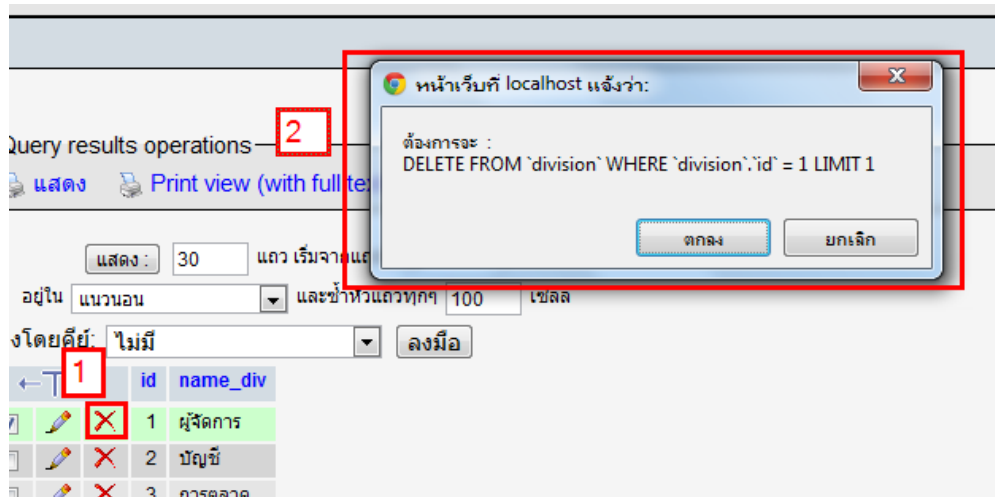
3. การลบข้อมูล

สมมติว่าต้องการลบ ข้อมูลของเทเบิล division

1. คลิกที่เทเบิล division
2. คลิกแถบ เปิดดู
3. ที่ส่วนหน้าของแต่ละเรคคอร์ด จะมีรูปกากบาทสีแดง ซึ่งหมายถึงการลบ เรคคอร์ด นั้นๆ

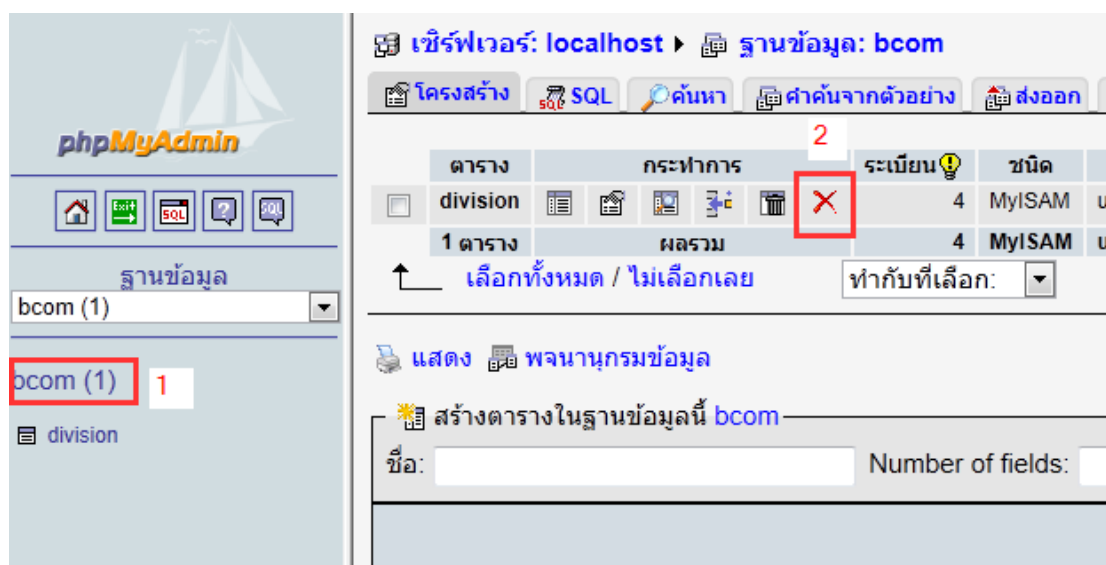
The screenshot shows the phpMyAdmin interface. At the top, the browser address bar shows 'localhost' and the database 'bcom' with table 'division'. The 'เปิดดู' (View) button is highlighted with a red box. Below the navigation bar, a yellow message box indicates 'แสดงระเบียนที่ 0 - 3 (4 ทั้งหมด, ค่าค้นใช้เวลา 0.0003 วินาที)'. The SQL query is displayed as 'SELECT * FROM `division` LIMIT 0, 30'. Below the query, there are buttons for 'แก้ไข' (Edit) and 'อธิบาย SQL' (Explain SQL). The 'Query results operations' section includes buttons for 'แสดง' (Show), 'Print view (with full texts)', and 'ส่งออก' (Export). The 'แสดง' button is set to '30' and 'แถว เริ่มจากแถวที่ 0'. The 'อยู่ใน' (In) dropdown is set to 'จำนวนอน' (All) and 'และข้ามแถวทุกๆ 100' (Skip every 100 rows) cells. The 'เรียงโดยคีย์' (Sort by key) dropdown is set to 'ไม่มี' (None). The query results table has 4 rows, each with a red 'X' icon in the first column, indicating delete options. The first row has 'id' 1 and 'name_div' 'ผู้จัดการ'.

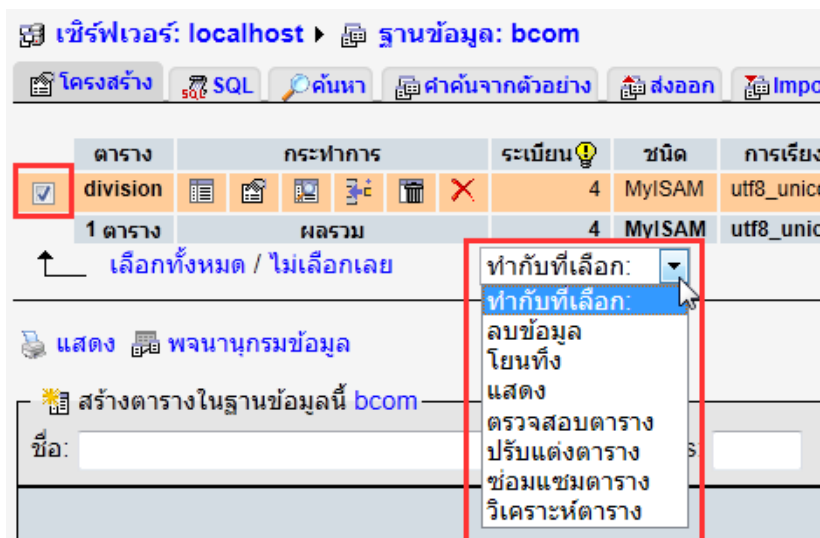
4. สมมติว่า ต้องการลบข้อมูลของเรคคอร์ด ผู้จัดการ ให้คลิกเครื่องหมาย กากบาท แถวที่ id เท่ากับ 1 จะปรากฏจอภาพ ดังรูปด้านล่าง เพื่อยืนยันการลบ
ถ้าต้องการลบ ให้คลิกปุ่ม ตกลง และถ้าไม่ต้องการลบ ให้คลิกปุ่ม ยกเลิก



4. การลบเทเบิล

การลบเทเบิลออกจากฐานข้อมูลให้คลิกที่ bcom จะเห็นเครื่องหมายกากบาท ในแต่ละแถวของตาราง(เทเบิล) ซึ่งหมายถึงต้องการลบเทเบิลใด ก็คลิกที่เครื่องหมายกากบาท ที่แถวนั้น





เราสามารถลบเฉพาะข้อมูลในเทเบิล หรือ ลบข้อมูลและลบเทเบิลนั้นได้

การเลือกด้านหน้าของเทเบิล คือการเลือกเทเบิลนั้น

ทำกับที่เลือก มีความหมายหลายอย่าง เช่น

ลบบันทึก หมายถึง ลบบันทึกอย่างเดียว แต่โครงสร้างเทเบิล ยังอยู่

โยนทิ้ง หมายถึง ลบบันทึก และ ลบโครงสร้างเทเบิลนั้นด้วย

เมื่อเราป้อนเทเบิล division, employee, province, users เข้าไปในฐานข้อมูล bcom แล้ว จะเห็นหน้าจอบ้างดังนี้



5. การสำรองข้อมูล

คลิกที่ bcom แล้วคลิกแถบ ส่งออก

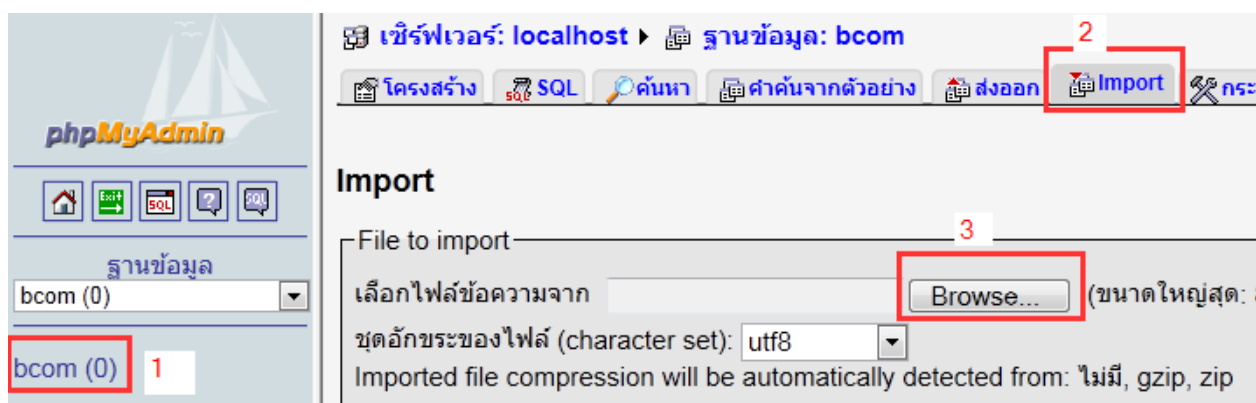
1. คลิกเลือก ฐานข้อมูลชื่อ bcom
2. คลิกแถบ ส่งออก
3. ตรวจสอบเทเบิล ว่าเลือกทั้งหมด ทุกเทเบิล หรือไม่
4. คลิกเลือกโครงสร้าง
5. คลิกเลือกข้อมูล
6. คลิกเลือก sql
7. คลิกปุ่ม ลงมือ

The screenshot shows the phpMyAdmin interface for the 'bcom' database. The 'Export' tab is active. The 'Database' dropdown is set to 'bcom (4)'. The 'Export type' is set to 'SQL'. The 'Export method' is set to 'Send to browser'. The 'Export options' section is expanded, showing 'Structure with comments' and 'Data' selected. The 'Export type' is set to 'INSERT'. The 'Export' button is highlighted with a red box and the number 7.

เราจะได้ไฟล์ชื่อ bcom.sql โดยไฟล์นี้ จะมีทั้งโครงสร้างของเทเบิล และมีข้อมูล อยู่ด้วย

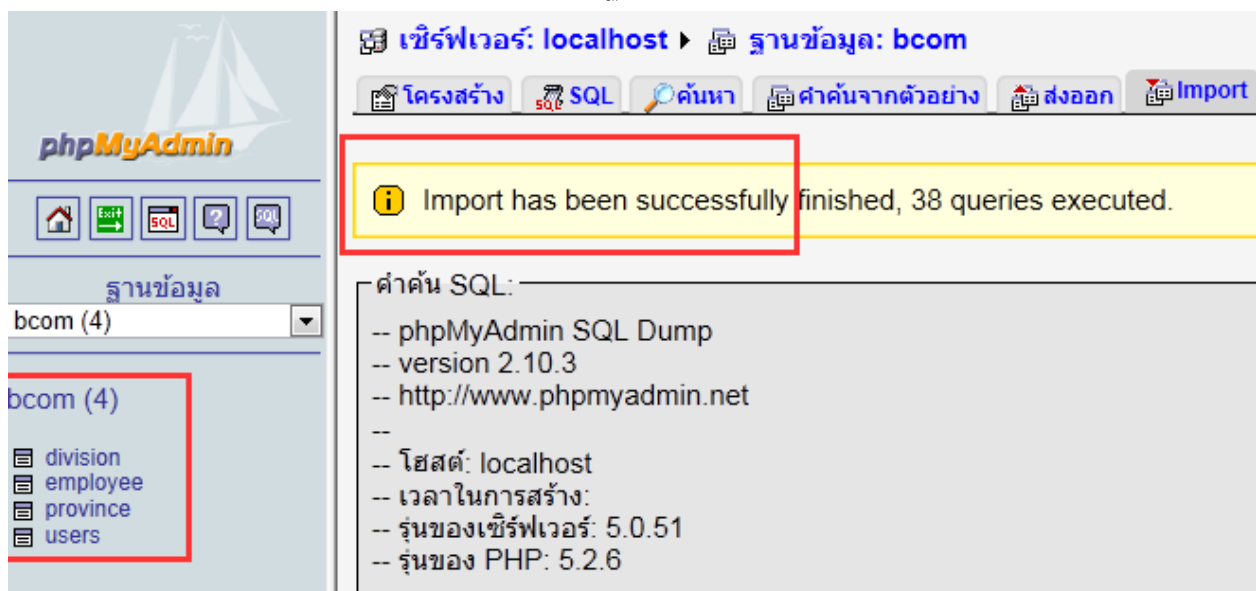
6. การนำไฟล์สำรองมาใช้

- 1 สร้างฐานข้อมูล ชื่อ bcom
- 2 คลิกแท็บ Import
- 3 คลิกปุ่ม Browse...
- 4 เลือกไฟล์ bcom.sql ที่เราเก็บไว้
5. คลิกปุ่ม ลงมือ



จะปรากฏให้รูปด้านล่าง ให้คลิกปุ่ม Browse... เพื่อเลือกไฟล์ bcom3209.sql แล้วคลิกปุ่ม ลงมือ

หน้าจอจะแสดงเทเบิล 4 เทเบิล แสดงว่านำข้อมูลเข้าฐานข้อมูลเรียบร้อยแล้ว



ต่อไปนี้เป็นคำอธิบายของชนิดข้อมูลแต่ละชนิดที่สามารถเลือกได้

VARCHAR : สำหรับเก็บข้อมูลประเภทตัวอักษร ทุกครั้งที่เลือกชนิดของฟิลด์เป็นประเภทนี้ จะต้องมีการกำหนดความยาวของข้อมูลลงไปด้วย ซึ่งสามารถกำหนดค่าได้ตั้งแต่ 1 - 255 ฟิลด์ชนิดนี้ เหมาะสำหรับการเก็บข้อมูลสั้นๆ เช่น ชื่อ นามสกุล หรือหัวข้อต่างๆ เป็นต้น... ในส่วนฟิลด์ประเภทนี้ จะสามารถเลือก "แอสกีไบต์" เป็น BINARY ได้ โดยปกติแล้วการจัดเรียงข้อมูลเวลาสืบค้น (query) สำหรับ VARCHAR จะเป็นแบบ case-sensitive (ตัวอักษรใหญ่ และเล็กมีความหมายแตกต่างกัน) แต่หากระบุ "แอสกีไบต์" เป็น BINARY การสืบค้นจะไม่คำนึงตัวอักษรว่าจะเป็นตัวใหญ่ หรือตัวเล็ก

CHAR : สำหรับเก็บข้อมูลประเภทตัวอักษร แบบที่ถูกจำกัดความกว้างเอาไว้คือ 255 ตัวอักษร ไม่สามารถปรับเปลี่ยนได้เหมือนกับ VARCHAR หากทำการสืบค้นโดยเรียงตามลำดับ ก็จะเรียงข้อมูลแบบ case-sensitive เว้นแต่จะกำหนดแอสกีไบต์เป็น BINARY ที่จะทำให้การเรียงข้อมูลเป็นแบบ non case-sensitive เช่นเดียวกับ VARCHAR

TINYTEXT : ในกรณีที่ต้องการความยาวๆ หรือต้องการที่จะค้นหาข้อความ โดยอาศัยฟีเจอร์ FULL TEXT SEARCH ของ MySQL เราอาจจะเลือกที่จะไม่เก็บข้อมูลลงในฟิลด์ประเภท VARCHAR ที่มีข้อจำกัดแค่ 256 ตัวอักษร แต่เราจะเก็บลงฟิลด์ประเภท TEXT แทน โดย TINYTEXT นี้ จะสามารถเก็บข้อมูลได้ 256 ตัวอักษร ซึ่งมองเผินๆ ก็ไม่ต่างกับเก็บลงฟิลด์ประเภท CHAR หรือ VARCHAR(255) เลย แต่จริงๆ มันต่างกันตรงที่ มันทำ FULL TEXT SEARCH ได้

TEXT : สำหรับเก็บข้อมูลประเภทตัวอักษร เช่นเดียวกับ TINYTEXT แต่สามารถเก็บได้มากขึ้น โดยสูงสุดคือ 65,535 ตัวอักษร หรือ 64KB เหมาะสำหรับการเก็บข้อมูลพวกเนื้อหาต่างๆ ที่ยาวๆ

MEDIUMTEXT : เก็บข้อมูลประเภทตัวอักษร เช่นเดียวกับ TINYTEXT แต่เก็บข้อมูลได้ 16,777,215 ตัวอักษร

LONGTEXT : เก็บข้อมูลประเภทตัวอักษร เช่นเดียวกับ TINYTEXT แต่เก็บข้อมูลได้ 4,294,967,295 ตัวอักษร

TINYINT : สำหรับเก็บข้อมูลชนิดตัวเลขที่มีขนาด 8 บิต ข้อมูลประเภทนี้เราสามารถกำหนดเพิ่มเติมในส่วนของ "แอดทริบิวต์" ได้ว่าจะเลือกเป็น UNSIGNED หรือ UNSIGNED ZEROFILL โดยจะมีความแตกต่างดังนี้

- UNSIGNED : จะหมายถึงเก็บค่าตัวเลขแบบไม่มีเครื่องหมาย แบบนี้จะทำให้สามารถเก็บค่าได้ตั้งแต่ 0 - 255
- UNSIGNED ZEROFILL : เหมือนข้างต้น แต่ว่าหากข้อมูลที่กรอกเข้ามาไม่ครบตามจำนวนหลักที่เรากำหนด ตัว MySQL จะทำการเติม 0 ให้ครบหลักเอง เช่น ถ้ากำหนดให้ใส่ได้ 3 หลัก แล้วทำการเก็บข้อมูล 25 เข้าไป เวลาที่สืบค้นดู เราจะได้ค่าออกมาเป็น 025

หากไม่เลือก "แอดทริบิวต์" สิ่งที่เราจะได้ก็คือ SIGNED นั่นก็จะต้องเสียบิตหนึ่งไปเก็บเครื่องหมาย บวก/ลบ ทำให้สามารถเก็บข้อมูลได้อยู่ในช่วง -128 ถึง 127 เท่านั้น

SMALLINT : สำหรับเก็บข้อมูลประเภทตัวเลขที่มีขนาด 16 บิต จึงสามารถเก็บค่าได้ตั้งแต่ -32768 ถึง 32767 (ในกรณีแบบคิดเครื่องหมาย) หรือ 0 ถึง 65535 (ในกรณี UNSIGNED หรือไม่คิดเครื่องหมาย) ซึ่งสามารถเลือก Attribute เป็น UNSIGNED และ UNSIGNED ZEROFILL ได้เช่นเดียวกับ TINYINT

MEDIUMINT : สำหรับเก็บข้อมูลประเภทตัวเลขที่มีขนาด 24 บิต นั่นก็หมายความว่าสามารถเก็บข้อมูลตัวเลขได้ตั้งแต่ -8388608 ไปจนถึง 8388607 (ในกรณีแบบคิดเครื่องหมาย) หรือ 0 ถึง 16777215 (ในกรณีที่เป็น UNSIGNED หรือไม่คิดเครื่องหมาย) ซึ่งสามารถเลือก Attribute เป็น UNSIGNED และ UNSIGNED ZEROFILL ได้เช่นเดียวกับ TINYINT

INT : สำหรับเก็บข้อมูลประเภทตัวเลขที่มีขนาด 32 บิต หรือสามารถเก็บข้อมูลได้ตั้งแต่ -2147483648 ไปจนถึง 2147483647 ครับ (ในกรณีแบบคิดเครื่องหมาย) หรือ 0 ถึง 4294967295 (ในกรณีที่เป็น UNSIGNED หรือไม่คิดเครื่องหมาย) ซึ่งสามารถเลือก Attribute เป็น UNSIGNED และ UNSIGNED ZEROFILL ได้เช่นเดียวกับ TINYINT

BIGINT : สำหรับเก็บข้อมูลประเภทตัวเลขที่มีขนาด 64 บิต สามารถเก็บข้อมูลได้ตั้งแต่ -9223372036854775808 ไปจนถึง 9223372036854775807 เลขที่เดียว (แบบคิดเครื่องหมาย) หรือ 0 ถึง 18446744073709551615 (ในกรณีที่เป็น UNSIGNED หรือไม่

คิดเครื่องหมาย) ซึ่งสามารถเลือก Attribute เป็น UNSIGNED และ UNSIGNED ZEROFILL ได้เช่นเดียวกับ TINYINT

FLOAT(M,D) : ที่กล่าวถึงไปทั้งหมด ในตระกูล INT นั้นจะเป็นเลขจำนวนเต็ม หากเราบันทึกข้อมูลที่มีเศษทศนิยม มันจะถูกปัดทันทันที ดังนั้นหากต้องการจะเก็บค่าที่เป็นเลขทศนิยม ต้องเลือกชนิดของฟิลด์เป็น FLOAT โดยจะเก็บข้อมูลแบบ 32 บิต คือมีค่าตั้งแต่ -3.402823466E+38 ไปจนถึง -1.175494351E-38, 0 และ 1.175494351E-38 ถึง 3.402823466E+38

DOUBLE(M,D) : สำหรับเก็บข้อมูลประเภทตัวเลขทศนิยม เช่นเดียวกับ FLOAT แต่มีขนาดเป็น 64 บิต สามารถเก็บได้ตั้งแต่ -1.7976931348623157E+308 ถึง -2.2250738585072014E-308, 0 และ 2.2250738585072014E-308 ถึง 1.7976931348623157E+308

DECIMAL(M,D) : สำหรับเก็บข้อมูลประเภทตัวเลขทศนิยม เช่นเดียวกับ FLOAT แต่ใช้กับข้อมูลที่ต้องการความละเอียดและถูกต้องของข้อมูลสูง

ข้อสังเกต เกี่ยวกับข้อมูลประเภท FLOAT, DOUBLE และ DECIMAL ก็คือ เวลากำหนดความยาวของข้อมูลในฟิลด์ จะถูกกำหนดอยู่ในรูปแบบ (M,D) ซึ่งหมายความว่า ต้องมีการระบุว่าจะให้มีตัวเลขส่วนที่เป็นจำนวนเต็มกี่หลัก และมีเลขทศนิยมกี่หลัก เช่น ถ้าเรากำหนดว่า FLOAT(5,2) จะหมายความว่า เราจะเก็บข้อมูลเป็นตัวเลขจำนวนเต็ม 5 หลัก และทศนิยม 2 หลัก ดังนั้นหากทำการใส่ข้อมูล 12345.6789 เข้าไป สิ่งที่จะเข้าไปอยู่ในข้อมูลจริงๆ ก็คือ 12345.68 (ปัดเศษให้มีจำนวนหลักตามที่กำหนดไว้)

DATE : สำหรับเก็บข้อมูลประเภทวันที่ โดยเก็บได้จาก 1 มกราคม ค.ศ. 1000 ถึง 31 ธันวาคม ค.ศ. 9999 โดยจะแสดงผลในรูปแบบ YYYY-MM-DD

DATETIME : สำหรับเก็บข้อมูลประเภทวันที่ และเวลา โดยจะเก็บได้ตั้งแต่ 1 มกราคม ค.ศ. 1000 เวลา 00:00:00 ไปจนถึง 31 ธันวาคม ค.ศ. 9999 เวลา 23:59:59 โดยรูปแบบการแสดงผล เวลาที่ทำการสืบค้น (query) ออกมา จะเป็น YYYY-MM-DD HH:MM:SS

TIMESTAMP[(M)] : สำหรับเก็บข้อมูลประเภทวันที่ และเวลาเช่นกัน แต่จะเก็บในรูปแบบของ YYYYMMDDHHMMSS หรือ YYMMDDHHMMSS หรือ YYYYMMDD หรือ YYMMDD แล้วแต่ว่าจะระบุค่า M เป็น 14, 12, 8 หรือ 6 ตามลำดับ สามารถเก็บได้ตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 ไปจนถึงประมาณปี ค.ศ. 2037

TIME : สำหรับเก็บข้อมูลประเภทเวลา มีค่าได้ตั้งแต่ -838:59:59 ไปจนถึง 838:59:59 โดยจะแสดงผลออกมาในรูปแบบ HH:MM:SS

YEAR[(2/4)] : สำหรับเก็บข้อมูลประเภทปี ในรูปแบบ YYYY หรือ YY แล้วแต่ว่าจะเลือก 2 หรือ 4 (หากไม่ระบุ จะถือว่าเป็น 4 หลัก) โดยหากเลือกเป็น 4 หลัก จะเก็บค่าได้ตั้งแต่ ค.ศ. 1901 ถึง 2155 แต่หากเป็น 2 หลัก จะเก็บตั้งแต่ ค.ศ. 1970 ถึง 2069

ข้อสังเกต ค่าที่เก็บในข้อมูลประเภท TIMESTAMP และ YEAR นั้นจะมีความสามารถพอๆ กับการเก็บข้อมูลวันเดือนปี และเวลา ด้วยฟิลด์ชนิด VARCHAR แต่ต่างกันตรงที่จะใช้เนื้อที่เก็บข้อมูลน้อยกว่า เพราะ ฟิลด์ประเภท TIMESTAMP นั้นจะมีข้อจำกัดในเรื่องของเวลาที่สามารถเก็บได้ คือจะต้องอยู่ในระหว่าง 1 มกราคม ค.ศ. 1000 ไปจนถึงแถวๆ ค.ศ. 2037 อย่างที่บอก แต่หากเก็บเป็น VARCHAR นั้นจะไม่ติดข้อจำกัดนี้

- ฟิลด์ชนิด YEAR ก็เหมือนกัน ใช้เนื้อที่แค่ 1 ไบต์เท่านั้นในการเก็บข้อมูล แต่ข้อจำกัดจะอยู่ที่ ปี ค.ศ. 1901 ถึง 2155 เท่านั้น (หรือ ค.ศ. 1970 ถึง 2069 ในกรณี 2 หลัก) แต่หากเก็บเป็น VARCHAR จะได้ตั้งแต่ 0000 ถึง 9999 เลย อันนี้เลยอยู่ที่ความจำเป็น

TINYBLOB : สำหรับเก็บข้อมูลประเภทไบนารี ได้แก่ ไฟล์ข้อมูลต่างๆ, ไฟล์รูปภาพ, ไฟล์มัลติมีเดีย เป็นต้น คือไฟล์อะไรก็ตามที่อัปโหลดผ่านฟอร์มอัปโหลดไฟล์ในภาษา HTML โดย TINYBLOB นั้นจะมีเนื้อที่ให้เก็บข้อมูลได้ 256 ไบต์

BLOB : สำหรับเก็บข้อมูลประเภทไบนารี เช่นเดียวกับ TINYBLOB แต่สามารถเก็บข้อมูลได้ 64KB

MEDIUMBLOB : สำหรับเก็บข้อมูลประเภทไบนารี เช่นเดียวกับ TINYBLOB แต่เก็บข้อมูลได้ 16MB

LONGBLOB : สำหรับเก็บข้อมูลประเภทไบนารี เช่นเดียวกับ TINYBLOB แต่เก็บข้อมูลได้ 4GB

ข้อสังเกต ข้อมูลประเภท BLOB นั้น แม้จะมีประโยชน์ในเรื่องของการเก็บข้อมูลประเภท BINARY ให้อยู่กับตัวฐานข้อมูล ทำให้สะดวกเวลาสืบค้นก็ตาม แต่มันก็ทำให้ฐานข้อมูลมีขนาดใหญ่เกินความจำเป็นด้วย ทำให้เกิดความไม่สะดวกในการสำรองฐานข้อมูลในกรณีที่มีข้อมูลอัปเดตไปเก็บมากๆ โดยปกติแล้ว จะใช้วิธีการอัปเดตไปเก็บไว้ในไฟล์เดออร์แล้วเก็บลิงก์ไปยังไฟล์เหล่านั้น เป็นฟิลด์ชนิด VARCHAR มากกว่า

SET : สำหรับเก็บข้อมูลที่เป็นกลุ่มของข้อมูลที่ยอมให้เลือกได้ 1 ค่าหรือหลายๆ ค่า ซึ่งสามารถกำหนดได้ถึง 64 ค่า