

บทที่ 7

ความรู้พื้นฐานเกี่ยวกับ MySQL

MySQL (อ่านว่า “มาย-เอส-คิว-แอล”) จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) ตัวหนึ่ง ซึ่งเป็นที่นิยมกันมากในปัจจุบัน โดยเฉพาะอย่างยิ่งในโลกของอินเทอร์เน็ต สาเหตุเพราะว่า MySQL เป็นฟรีแวร์ทางด้านฐานข้อมูลที่มีประสิทธิภาพสูง เป็นทางเลือกใหม่จากผลิตภัณฑ์ระบบจัดการฐานข้อมูลในปัจจุบัน ที่มักจะเป็นการผูกขาดของผลิตภัณฑ์เพียงไม่กี่ตัว นักพัฒนาระบบฐานข้อมูลที่เคยใช้ MySQL ต่างยอมรับในความสามารถความรวดเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมหาศาล ทั้งยังสนับสนุนการใช้งานบนระบบปฏิบัติการมากมาย ไม่ว่าจะเป็น Unix, OS/2, Mac OS หรือ Windows ก็ตาม นอกจากนี้ MySQL ยังสามารถใช้งานร่วมกับ Web Development Platform ทั้งหลาย ไม่ว่าจะเป็น C, C++, Java, Perl, PHP, Python, Tcl หรือ ASP ก็ตามที่ ดั้งนั้นจึงไม่เป็นที่น่าแปลกใจเลยว่า ทำไม MySQL จึงได้รับความนิยมอย่างมากในปัจจุบัน และมีแนวโน้มสูงยิ่งขึ้นต่อไปในอนาคต

MySQL จัดเป็นซอฟต์แวร์ประเภท Open Source Software สามารถดาวน์โหลด Source Code ต้นฉบับได้จากอินเทอร์เน็ต โดยไม่เสียค่าใช้จ่ายใดๆ การแก้ไขก็สามารถกระทำได้ตามความต้องการ MySQL ยึดถือสิทธิบัตรตาม GPL (GNU General Public License) ซึ่งเป็นข้อกำหนดของซอฟต์แวร์ประเภทนี้ส่วนใหญ่ โดยจะเป็นการชี้แจงว่า สิ่งใดทำได้ หรือทำไม่ได้สำหรับการใช้งานในกรณีต่างๆ ทั้งนี้ถ้าต้องการข้อมูลเพิ่มเติม หรือรายละเอียดของ GPL สามารถหาข้อมูลได้จากเว็บไซต์ <http://www.gnu.org/>

MySQL ได้รับการยอมรับและทดสอบเรื่องของความรวดเร็วในการใช้งาน โดยจะมีการทดสอบและเปรียบเทียบกับผลิตภัณฑ์ทางด้านฐานข้อมูลอื่นอยู่เสมอ มีการพัฒนาอย่างต่อเนื่อง โดยเริ่มตั้งแต่เวอร์ชันแรกๆ ที่ยังไม่ค่อยมีความสามารถมากนัก มาจนถึงทุกวันนี้ MySQL ได้รับการพัฒนาให้มีความสามารถมากยิ่งขึ้น รองรับข้อมูลจำนวนมหาศาล สามารถใช้งานหลายผู้ใช้ได้พร้อมๆ กัน (Multi-user) มีการออกแบบให้สามารถทำงานออก เพื่อช่วยการทำงานให้เร็วยิ่งขึ้น รองรับข้อมูลจำนวนมหาศาล เพื่อช่วยการทำงานเร็วยิ่งขึ้น (Multi-threaded) วิธีและการเชื่อมต่อที่ดีขึ้น การกำหนดสิทธิและการรักษาความปลอดภัยของข้อมูลมีความรัดกุมน่าเชื่อถือยิ่งขึ้น เครื่องมือหรือโปรแกรมสนับสนุนทั้งของตัวเองและของผู้พัฒนาอื่นๆ มีมากยิ่งขึ้น นอกจากนี้สิ่งหนึ่งที่สำคัญคือ “MySQL ได้รับการพัฒนาไปในแนวทางตามข้อกำหนดมาตรฐาน SQL ดังนั้น เราสามารถใช้คำสั่ง SQL ในการทำงาน MySQL ได้” นักพัฒนาที่ใช้ SQL มาตรฐานอยู่แล้ว ไม่ต้องศึกษาคำสั่งเพิ่มเติม แต่อาจจะต้องเรียนรู้ถึงรูปแบบและข้อจำกัดบางอย่างโดยเฉพาะ ทั้งนี้ทั้งนั้น ทางทีมงานผู้พัฒนา MySQL มีเป้าหมายอย่างชัดเจนที่จะพัฒนาให้ MySQL มีความสามารถสนับสนุนตามข้อกำหนด SQL92 มากที่สุด และจะพัฒนาให้เป็นไปตามข้อกำหนด SQL99 ต่อไป

ทุกวันนี้มีการนำ MySQL ไปใช้ในระบบต่างๆ มากมาย ไม่ว่าจะเป็นระบบเล็กๆ ที่มีจำนวนตารางข้อมูลน้อย มีความสัมพันธ์ของข้อมูลในแต่ละตารางไม่ซับซ้อน เช่น ระบบฐานข้อมูลบุคคลในแผนกเล็กๆ ไปจนถึงระบบจัดการข้อมูลขนาดใหญ่ ที่ประกอบด้วยตารางข้อมูลมากมาย มีความสัมพันธ์ของข้อมูลในแต่ละตารางซับซ้อน เช่น ระบบสต็อกสินค้า ระบบบัญชีเงินเดือน เป็นต้น โดยเฉพาะอย่างยิ่งในปัจจุบัน มีการใช้ MySQL เป็น Database Server เพื่อการทำงานสำหรับ Web Database Application ในโลกของอินเทอร์เน็ตมากขึ้น สิ่งนี้เป็นคำตอบว่า “ทำไมคุณต้องเรียนรู้และใช้งาน MySQL ?”

สถาปัตยกรรมของ MySQL

สถาปัตยกรรม หรือ โครงสร้างภายในของ MySQL ก็คือ การออกแบบการทำงานในลักษณะของ Client/Server นั่นเอง ซึ่งประกอบด้วยส่วนหลักๆ 2 ส่วน คือ ส่วนของผู้ให้บริการ (Server) และ ส่วนของผู้ใช้บริการ (Client) โดยในแต่ละส่วนจะมีโปรแกรมสำหรับการทำงานตามหน้าที่ของตน

ส่วนของผู้ให้บริการ หรือ Server จะเป็นส่วนที่ทำหน้าที่บริหารจัดการระบบฐานข้อมูลในที่นี้ก็หมายถึงตัว MySQL Server นั่นเอง และเป็นที่จัดเก็บข้อมูลทั้งหมด ข้อมูลที่เก็บไว้นี้มีข้อมูลที่จำเป็นสำหรับการทำงานกับระบบฐานข้อมูล และข้อมูลที่เกิดจากการที่ผู้ใช้แต่ละคนสร้างขึ้นมา

ส่วนของผู้ใช้บริการ หรือ Client ก็คือผู้ใช้นั่นเอง โดยโปรแกรมสำหรับใช้งานในส่วนนี้ได้แก่ MySQL Client, Access, Web Development Platform ต่างๆ (เช่น Java, Perl, PHP, ASP เป็นต้น)

หลักการทำงานในลักษณะ Client/ Server มีดังนี้

1. ที่ฝั่งของ Server จะมีโปรแกรมหรือระบบสำหรับจัดการฐานข้อมูลทำงานรออยู่ เพื่อเตรียมหรือรอคอยการร้องขอการใช้บริการจาก Client
2. เมื่อมีการร้องขอการใช้บริการเข้ามา Server จะทำการตรวจสอบตามวิธีการของตน เช่น อาจจะมีการให้ผู้ให้บริการระบุชื่อและรหัสผ่าน และสำหรับ MySQL สามารถกำหนดได้ว่าจะอนุญาตหรือปฏิเสธ Client ใดๆ ในระบบที่จะเข้าใช้บริการอีกด้วย ซึ่งจะได้อธิบายรายละเอียดในเรื่องต่อไป
3. ถ้าผ่านการตรวจสอบ Server ก็จะอนุมัติการใช้บริการแก่ Client ที่ร้องขอการใช้บริการนั้นๆ ต่อไป และถ้าในกรณีที่ไม่ได้รับการอนุมัติ Server ก็จะส่งข่าวสารความผิดพลาดแจ้งกลับไป Client ที่ร้องขอการใช้บริการนั้น

เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client หรือ Server อาจอยู่บนเครื่องเดียวกัน หรือแยกเครื่องกันก็ได้ ทั้งนี้ขึ้นอยู่กับลักษณะการทำงาน หรือการกำหนดของผู้บริหารระบบ ตามปกติถ้าเป็นการทำงานลักษณะ Web-based มีการใช้ฐานข้อมูลขนาดไม่ใหญ่นัก ตัว MySQL Server และ Client มักจะมีอยู่บนเครื่องเดียวกัน โดยเครื่องคอมพิวเตอร์ดังกล่าวจะต้องมีทรัพยากรเพื่อการทำงาน เช่น เนื้อที่ฮาร์ดดิสก์, RAM มากพอสมควร แต่สำหรับการทำงานจริง (Real-world Application) ก็มักจะแยก Client และ Server ออกเป็นคนละเครื่องกัน และสามารถรองรับงานได้ดีมากกว่า ดังนั้น ผู้บริหารระบบ หรือผู้กำหนดนโยบายสำหรับการทำงานเครือข่าย จะต้องคำนึงถึงเรื่องที่เกี่ยวข้องเหล่านี้ให้ดี เพื่อที่จะทำให้ระบบมีการทำงานรับการให้บริการแก่ผู้ใช้อย่างมีประสิทธิภาพ และข้อมูลมีความปลอดภัยมากที่สุด

MySQL กับมาตรฐานภาษา SQL

จุดมุ่งหมายหนึ่งของทีมผู้พัฒนา MySQL ต้องการทำให้ MySQL เป็น DB Server ตามข้อกำหนด ANSI SQL โดยในขั้นแรกนี้อ้างอิงตาม ANSI SQL92 เป็นหลัก ความสามารถต่างๆ โดยส่วนใหญ่ครอบคลุมการใช้งาน อาจจะมีบางสิ่งที่ยังไม่สามารถทำได้ในปัจจุบัน หรือบางสิ่งก็มีแผนจะเพิ่มเติมเข้าไป ซึ่งได้กล่าวถึงรายละเอียดไปแล้วข้างต้น แต่ในส่วนนี้เราจะมาดูสำหรับความสามารถบางอย่างที่ MySQL เพิ่มเติมเข้ามาสำหรับการทำงาน นอกเหนือจากข้อกำหนด ANSI SQL ดังนี้

- ฟังก์ชันประเภท MEDIUMINT, SET, ENUM และ BLOB รวมทั้ง TEXT ต่างๆ

- การกำหนดคุณลักษณะของฟิลด์ เช่น AUTO_INCREMENT, BINARY, NULL, UNSIGNED และ ZEROFILL
- การเปรียบเทียบสตริงเป็นแบบ Case Insensitive การจัดเรียงข้อมูลใช้รูปแบบภาษา ISO-8859-1 (Latin1) เป็นค่า default
- ชื่อ database และชื่อตารางข้อมูลจะ Case Sensitive (โดยเฉพาะ Unix Platform)
- สามารถใช้คำสั่งของระบบปฏิบัติการในการสำรอง เปลี่ยนชื่อ ย้าย ลบ หรือคัดลอกตารางข้อมูลซึ่งเกี่ยวข้องกับไฟล์ .MYD, .MYI และ .frm ได้โดยตรง
- MySQL ไม่สนับสนุน Tablespaces
- โอเปอเรเตอร์ LIKE สามารถใช้กับ Numeric Columns ได้
- การใช้ INTO OUTFILE และ STRAIGHT_JOIN ในคำสั่ง SELECT
- สามารถใช้ SQL_SMALL_RESULT option ในคำสั่ง SELECT ได้
- คำสั่ง EXPLAIN SELECT เพื่อให้แสดงรายละเอียดของการเชื่อมกันของตารางข้อมูล
- สามารถใช้ TEMPORARY หรือ IF NOT EXISTS สำหรับการ CREATE TABLE ได้
- สามารถทำการ CHANGE ชื่อคอลัมน์, DROP ชื่อคอลัมน์ หรือ DROP INDEX, IGNORE หรือ RENAME ในการใช้คำสั่ง ALTER TABLE ได้
- สามารถทำการ RENAME TABLE ได้
- สามารถ ADD, ALTER, DROP หรือ CHANGE clauses หลายครั้ง ในคำสั่ง ALTER TABLE ได้
- สามารถทำการ DROP TABLE ด้วยคีย์เวิร์ด IF EXISTS
- สามารถทำการครอป (Drop) ตารางข้อมูลหลายๆ ตารางได้ด้วยคำสั่ง DROP TABLE เพียงครั้งเดียว
- การใช้ LIMIT clause ในคำสั่ง DELETE
- การใช้ DELAYED clause ในคำสั่ง INSERT และ REPLACE
- การใช้ LOW_PRIORITY clause ในคำสั่ง INSERT, REPLACE, DELETE และ UPDATE
- การใช้งาน LOAD DATA INFILE ซึ่งไวยากรณ์ใช้ได้กับรูปแบบการโหลดคาด้า (Load Data) ของ Oracle
- คำสั่ง ANALYZE TABLE, CHECK TABLE, OPTIMIZE TABLE และ REPAIR TABLE
- คำสั่ง SHOW
- คำสั่ง SET OPTION
- ฟังก์ชันหลายฟังก์ชันที่จำเป็น เช่น ฟังก์ชันสำหรับทำงานสตริง ซึ่งสามารถใช้ได้ทันที เนื่องจากมีชื่อและการทำงานที่เหมือนกันของ ANSI SQL และ ODBC
- สามารถใช้ || แทน OR และ && แทน AND ซึ่งเหมือนกับในภาษา C แต่ไม่สามารถใช้ในการเชื่อมสตริงได้ ซึ่ง MySQL ให้ใช้ฟังก์ชัน CONCAT() แทน
- คำสั่งในการ CREATE DATABASE หรือ DROP DATABASE
- การใช้โอเปอเรเตอร์ % โดยจะให้ผลลัพธ์เดียวกันการใช้ฟังก์ชัน MOD() ตัวอย่าง เช่น $M \% N = \text{MOD}(M,N)$ เป็นต้น

- โอเปอเรเตอร์ =, <>, <=, <, >=, <<, >>, <=>, AND, OR หรือ LIKE สามารถใช้ในการเปรียบเทียบคอลัมน์ในการใช้คำสั่ง SELECT ได้ ทั้งนี้จะอยู่ทางด้านหน้า FROM ดังตัวอย่างเช่น
SELECT id = 12345 AND name='somchai' FROM person;
- ฟังก์ชัน LAST_INSERT_ID()
- ฟังก์ชัน BIT_COUNT(), CASE, ELT(), FROM_DATS(), FORMAT(), IF(), PASSWORD(), ENCRYPT(), md5(), ENCODE(), DECODE(), PERIOD_ADD(), PERIOD_DIFF(), TO_DAYS(), หรือ WEEKDAY()
- ฟังก์ชัน TRIM() เพื่อการตัดส่วนของสตริง ซึ่งใน ANSI SQL จะตัดได้เพียงหนึ่งตัวอักษรเท่านั้น
- GROUP BY function STD(), BIT_OR() และ BIT_AND()
- การใช้คำสั่ง REPLACE แทนการใช้คำสั่ง DELETE ตามด้วยคำสั่ง INSERT
- คำสั่ง FLUSH flush_option
- การใช้ตัวแปร sql statement ด้วย := ตัวอย่างเช่น SELECT @asum:=SUM (total), @cnt:=COUNT(*), @asum/@cnt AS avg FROM person;

ประเภทข้อมูลและตารางข้อมูลใน MySQL

ในส่วนนี้จะกล่าวถึงรายละเอียดของประเภทข้อมูลที่สามารถกำหนดให้แก่แต่ละฟิลด์ ซึ่งสำหรับระบบฐานข้อมูล MySQL จะมีส่วนที่สนับสนุนกับมาตรฐาน และส่วนที่แตกต่างเพิ่มเติมจากมาตรฐาน และประเภทตารางข้อมูลที่สามารถเลือกใช้ได้ ใน MySQL

การศึกษาริเริ่มรู้ถึงประเภทข้อมูลใน MySQL จะทำให้เราเข้าใจ และสามารถทำการกำหนดประเภทของข้อมูลให้ถูกต้องเหมาะสมตามจุดประสงค์ หรือตามนโยบายการทำงานของเรา มิฉะนั้นแล้วอาจก่อให้เกิดความเสียหาย หรือทำให้เกิดข้อจำกัดในเรื่องข้อมูลของการทำงานได้

การศึกษาในเรื่องของประเภทข้อมูลใน MySQL ก็มีจุดประสงค์เช่นเดียวกัน ซึ่งจะได้ศึกษาในรายละเอียดต่อไป

ประเภทข้อมูลใน MySQL

หลักการพิจารณาเลือกใช้ประเภทข้อมูล ดังนี้

1. ประเภทข้อมูลหลัก ได้แก่ ตัวเลข, ตัวอักษร, วันที่และเวลา, ข้อมูลไบนารี และอื่นๆ สำหรับข้อมูลประเภทตรรกะ MySQL ยังไม่สนับสนุนในเวอร์ชันปัจจุบัน แต่จะสนับสนุนในเร็วๆ นี้

2. ในแต่ละประเภทข้อมูล จะมีประเภทย่อยๆ ลงไปอีก เพื่อให้ผู้ใช้เลือกใช้ประเภทข้อมูลที่เหมาะสม เพราะนอกจากจะทำงานได้ถูกต้องแล้ว ยังเป็นการช่วยประหยัดเนื้อที่การเก็บข้อมูลอีกด้วย ซึ่งต่อไปเราจะได้ศึกษาเพิ่มเติมว่าในแต่ละประเภทข้อมูลเป็นอย่างไร มีขอบเขตข้อมูลอะไรบ้าง ใช้เนื้อที่ในการเก็บข้อมูลเท่าไร

3. การเลือกใช้ประเภทข้อมูล หากมีความต้องการที่จะสนับสนุนหรือใช้มาตรฐาน เช่น ODBC หรือต้องการ port ข้อมูลไปใช้ในระบบอื่น ควรเลือกใช้ประเภทข้อมูลที่สนับสนุนกับมาตรฐานเท่านั้น เพื่อป้องกันความผิดพลาดที่อาจจะเกิดขึ้นต่อไปในอนาคต

4. ฟังก์ชันที่ไม่มีการคำนวณ หรือไม่มีแนวโน้มจะเกี่ยวข้องกับการคำนวณเลย ควรเลือกใช้ข้อมูลประเภทตัวอักษร เพราะจะประหยัดเนื้อที่ในการจัดเก็บจริง

5. การเลือกใช้ประเภทของข้อมูลในแต่ละฟิลด์ นอกจากจะดูความเป็นไปได้ในปัจจุบันแล้ว จะต้องคำนึงและดูแนวโน้มความเป็นไปได้ของข้อมูล หรือปริมาณของข้อมูลต่อไปในอนาคตด้วย เพื่อป้องกันข้อจำกัดของข้อมูลเมื่อมีการขยายตัวเพิ่มขึ้น

6. การเลือกใช้ข้อมูลประเภทวันที่และเวลาใน MySQL จะต้องเลือกใช้อย่างระมัดระวัง เนื่องจากมีข้อจำกัดและรายละเอียดที่แตกต่างไปจากมาตรฐาน เช่น จะสามารถใช้ข้อมูลเริ่มต้นได้ที่ปี ค.ศ. 1000 เท่านั้น เป็นต้น

ประเภทข้อมูลใน MySQL แบ่งออกเป็นกลุ่มได้ ดังนี้

1. ประเภทข้อมูลสำหรับตัวเลข

ไว้สำหรับเก็บข้อมูลตัวเลข ซึ่งอาจจะใช้ในการคำนวณ หรือการจัดเรียงข้อมูลเปรียบเทียบกันในฟิลด์นั้นๆ ประกอบด้วยประเภทข้อมูลย่อยๆ ได้แก่ จำนวนเต็ม, จำนวนทศนิยม, จำนวนจริง

1.1 TINYINT(M) [UNSIGNED] [ZEROFILL]

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -128 ถึง 127 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 255 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 1 ไบต์

1.2 SMALLINT(M) [UNSIGNED] [ZEROFILL]

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -32768 ถึง 32767 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 65535 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 2 ไบต์

1.3 MEDIUMINT(M) [UNSIGNED] [ZEROFILL]

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -8388608 ถึง 8388607 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 16777215 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 3 ไบต์

1.4 INT(M) [UNSIGNED] [ZEROFILL]

หรือ INTEGER(M) [UNSIGNED] [ZEROFILL]

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -2147483648 ถึง 2147483647 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 4294967295 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 4 ไบต์

1.5 BIGINT(M) [UNSIGNED] [ZEROFILL]

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -9223372036854775808 ถึง 9223372036854775807 แต่ถ้าแบบไม่คิดเครื่องหมาย จะใช้ได้ตั้งแต่ 0 ถึง 18446744073709551615 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 8 ไบต์

1.6 FLOAT(precision) [ZEROFILL]

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย precision เป็นค่าความละเอียดทศนิยม ซึ่งแบ่งเป็นชนิด single (มีค่าตั้งแต่ 0-24) และแบบ double (ตั้งแต่ 25-53) ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 4 หรือ 8 ไบต์ โดยแบบ single จะใช้ 4 ไบต์ และแบบ double จะใช้ 8 ไบต์ ชนิดข้อมูลประเภทนี้สำหรับกรณีสืบกับ ODBC มาตรฐาน

1.7 FLOAT(M,D) [ZEROFILL]

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -3.402823466E+38 ถึง

-1.175494351E -38,0 และ 1.175494351E-39 ถึง 3.402823466E+38 ค่า M เป็นจำนวนหลักที่ต้องการแสดงผล และ D เป็นจำนวนจุดทศนิยม ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 4 ไบต์

1.8 DOUBLE(M,D) [ZEROFILL]

หรือ DOUBLE PRECISION(M,D) [ZEROFILL]

หรือ REAL(M,D) [ZEROFILL]

ข้อมูลชนิดตัวเลขแบบคิดเครื่องหมาย จะใช้ได้ตั้งแต่ -1.7976931348623157E+308 ถึง

-2.2250738585072014E-308, 0 และ 2.2250738585072014E-308 ถึง 1.7976931348623157E+308 ค่า M เป็นจำนวนหลักที่ต้องการแสดงผล และ D เป็นจำนวนจุดทศนิยม ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูล ขนาด 8 ไบต์

1.9 DECIMAL(M,D) [ZEROFILL]

หรือ NUMERIC(M,D) [ZEROFILL]

ข้อมูลชนิดตัวเลขชนิด unpacked คืออนุญาตให้สามารถเก็บข้อมูลตัวอักษรเข้าไปด้วย โดยตัวอักษรหนึ่งตัวแทนแต่ละหลัก สามารถใช้ตัวเลขได้เท่ากับแบบ DOUBLE ใช้เนื้อที่เก็บข้อมูลเท่ากับ M ไบต์ (D+2, ถ้า M < D)

2. ประเภทข้อมูลสำหรับวันที่และเวลา

2.1 DATE

ข้อมูลชนิดวันที่ ตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 ถึง 31 ธันวาคม ค.ศ. 9999 การแสดงผลวันที่อยู่ในรูปแบบ 'YYYY-MM-DD' ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 3 ไบต์

2.2 DATETIME

ข้อมูลชนิดวันที่และเวลา ตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 เวลา 00:00:00 ถึง 31 ธันวาคม ค.ศ. 9999 เวลา 23:59:59 การแสดงผลวันที่และเวลาอยู่ในรูปแบบ 'YYYY-MM-DD HH:MM:SS' ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 8 ไบต์

2.3 TIMESTAMP(M)

ค่า timestamp เป็นค่าตัวเลขที่นับจำนวนวินาทีตั้งแต่วันที่ 1 มกราคม ค.ศ. 1000 เวลา 00:00:00 เป็นต้นมา โดยมีจุดสิ้นสุดประมาณ ค.ศ. 2037 การแสดงผลค่า timestamp อยู่ในรูปแบบ YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD หรือ YYMMDD ค่า M คือจำนวนตัวเลขที่บรรจุ ซึ่งอาจจะเป็น 14, 12, 8 หรือ 6 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 4 ไบต์

2.4 TIME

ข้อมูลประเภทเวลา สามารถเป็นได้ตั้งแต่ '-838:59:59' ถึง '838:59:59' แสดงผลในรูปแบบ HH:MM:SS ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 3 ไบต์

2.5 YEAR(2/4)

ข้อมูลประเภทปี ค.ศ. โดยสามารถเลือกว่าจะใช้แบบ 2 หรือ 4 หลัก (ค่าโดยปริยายจะเป็น 4 หลัก) ถ้าเลือกใช้แบบ 4 หลัก จะใช้ได้ตั้งแต่ปี ค.ศ. 1901 ถึง 2155 ถ้าเลือกใช้แบบ 2 หลัก จะใช้ได้ตั้งแต่ปี ค.ศ. 1970 ถึง 2069 แสดงในรูปแบบ 70 ถึง 69 ข้อมูลชนิดนี้ใช้เนื้อที่เก็บข้อมูลขนาด 1 ไบต์

3. ประเภทข้อมูลสำหรับตัวอักษร

ข้อมูลประเภทตัวอักษรถือว่าเป็นสิ่งที่ใช้กันมากที่สุด เพราะแทบจะครอบคลุมข้อมูลที่เป็นไปได้ทั้งหมด การจัดเรียงข้อมูลจะเป็นในลักษณะไม่คำนึงตัวใหญ่ตัวเล็ก (case-insensitive) ซึ่งเมื่อเปรียบเทียบกับข้อมูลประเภทไบนารี (BLOB) แล้วจะพบว่าคล้ายกันมาก จะแตกต่างกันที่ BLOB จะมีการจัดเรียงข้อมูลเป็น case-insensitive เท่านั้น แต่ข้อมูลประเภทตัวอักษรก็มีรายละเอียดเพิ่มเติมไปอีก โดยมีรายละเอียดดังนี้

3.1 [NATIONAL] CHAR(M) [BINARY]

ข้อมูลประเภทสตริงที่จำกัดขนาดความกว้าง (ไม่สามารถปรับขนาดได้) โดยขนาดความกว้างเป็นได้ตั้งแต่ 1 ถึง 255 ตัวอักษร ตามปกติเมื่อมีการเรียงข้อมูล จะเป็นลักษณะ case-sensitive คือคำนึงถึงตัวเล็กตัวใหญ่ การระบุชนิดข้อมูลย่อว่าเป็น BINARY จะเป็นลักษณะไม่คำนึงตัวเล็กตัวใหญ่ ดังนั้นเมื่อมีการเรียงข้อมูลก็จะให้ผลต่างจาก CHAR ธรรมดา การเก็บข้อมูลก็ใช้ไปต์ตามจำนวนตัวอักษรที่ระบุ

3.2 [NATIONAL] VARCHAR(M) [BINARY]

ข้อมูลประเภทนี้ก็คล้ายกับแบบ CHAR ต่างกันตรงที่ VARCHAR จะสามารถปรับขนาดตามข้อมูลที่เก็บในฟิลด์ ขนาดความกว้างอยู่ตั้งแต่ 1 ถึง 255 การเก็บข้อมูลจะเท่ากับของข้อมูลจริงในฟิลด์ฯ + 1 ไบต์

3.3 TINYTEXT

ข้อมูลประเภทนี้สามารถใช้ความกว้างข้อมูลได้สูงสุด 255 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 1 ไบต์

3.4 TEXT

ข้อมูลประเภทนี้สามารถใช้ความกว้างข้อมูลได้สูงสุด 65,535 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 2 ไบต์

3.5 MEDIUMTEXT

ข้อมูลประเภทนี้สามารถใช้ความกว้างข้อมูลได้สูงสุด 16,777,215 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 3 ไบต์

3.6 LONGTEXT

ข้อมูลประเภทนี้สามารถใช้ความกว้างข้อมูลได้สูงสุด 4,294,967,295 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 4 ไบต์

3.6 ENUM('value1','value2',...)

ข้อมูลประเภทระบุเฉพาะค่าที่ต้องการ หรือถ้าไม่มีจะให้ค่า NULL สามารถกำหนดค่าได้ถึง 65,535 ค่า และใช้เนื้อที่เก็บข้อมูลตามจำนวนค่าที่ระบุ

3.7 SET('value1','value2',...)

ข้อมูลประเภทเซต ประกอบด้วยข้อมูลตั้งแต่ไม่มีค่า หรือมีค่าตามสมาชิกที่กำหนด สามารถมีจำนวนสมาชิกในเซตได้ทั้งสิ้น 64 ตัว และใช้เนื้อที่เก็บข้อมูลตามจำนวนสมาชิกที่ระบุ

4. ประเภทข้อมูลสำหรับไบนารี (BLOB: Binary Large Object)

จากที่กล่าวไปแล้วถึงความคล้ายกันของแต่ละข้อมูลประเภทตัวอักษรและข้อมูลประเภทไบนารี ดังนั้นประเภทข้อมูลย่อและการเก็บข้อมูลจึงไม่แตกต่างกัน แต่ตามปกติเรามักใช้ BLOB สำหรับเก็บข้อมูลที่ไม่เป็น

ภาษามนุษย์ ข้อมูลไบนารี ได้แก่ รูปภาพ, ไฟล์ข้อมูล หรือข้อมูลที่ประกอบด้วยตัวอักษรพิเศษ ข้อมูลที่กำหนดด้วยรหัสควบคุม เป็นต้น

4.1 TINYBLOB

สำหรับข้อมูลไบนารี สามารถใช้ความกว้างข้อมูลได้สูงสุด 255 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 1 ไบต์

4.2 BLOB

สำหรับข้อมูลไบนารี สามารถใช้ความกว้างข้อมูลได้สูงสุด 65,535 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 2 ไบต์

4.3 MEDIUMBLOB

สำหรับข้อมูลไบนารี สามารถใช้ความกว้างข้อมูลได้สูงสุด 16,777,215 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 3 ไบต์

4.2 LONGBLOB

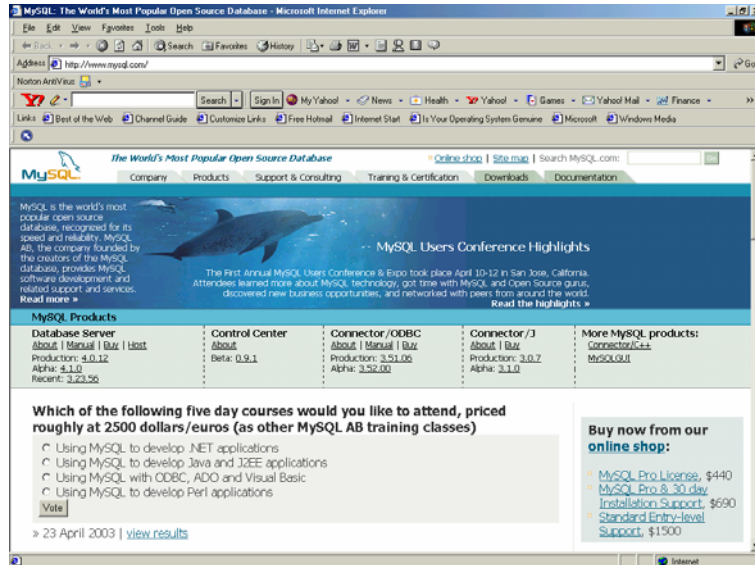
สำหรับข้อมูลไบนารี สามารถใช้ความกว้างข้อมูลได้สูงสุด 4,294,967,295 ตัวอักษร และใช้เนื้อที่เก็บข้อมูลเท่ากับจำนวนข้อมูลจริง + 4 ไบต์

การติดตั้ง MySQL และการปรับแต่งระบบเพื่อการใช้งาน

MySQL source

- แนะนำเว็บไซต์ www.mysql.com

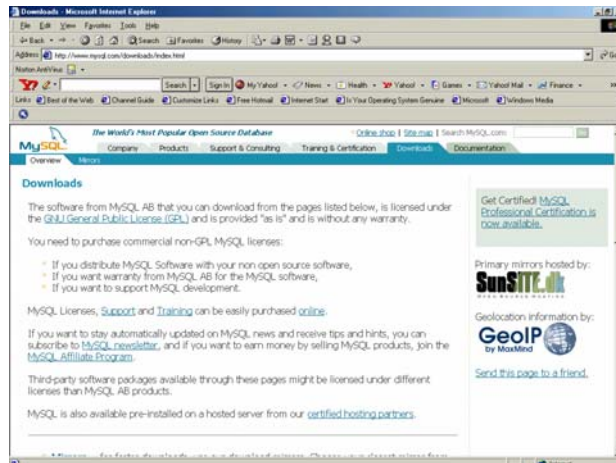
เว็บไซต์ www.mysql.com เป็นเว็บไซต์อย่างเป็นทางการของ MySQL ซึ่งเนื้อหาภายในเว็บไซต์หลักๆ จะเป็นการให้ข้อมูลและรายละเอียดเกี่ยวกับ MySQL ไม่ว่าจะเป็น Server, Client หรือ เครื่องมืออื่นๆ ที่เกี่ยวข้อง จุดประสงค์ที่สำคัญอันหนึ่ง ก็คือไว้สำหรับการเผยแพร่ MySQL ที่ได้รับการพัฒนา มีให้ดาวน์โหลด ตั้งแต่เวอร์ชันที่สมบูรณ์ (Stable) หรือที่กำลังพัฒนา (Beta) มีหลากหลายระบบปฏิบัติการ ผู้ใช้ที่จะนำ MySQL ไปรันบนระบบปฏิบัติการตัวใดก็สามารถเลือกได้จากที่เว็บไซต์นี้ นอกเหนือจากนี้ก็จะยังมีลิงก์ต่างๆ ที่มีการนำ MySQL ไปพัฒนาเพิ่มเติม โดยมีทั้งแบบ Free และ Commercial ให้เลือกตามอัธยาศัย



เว็บไซต์ <http://www.mysql.com/>

- ดาวน์โหลด Source Code หรือ Binary Installer

การดาวน์โหลด Source Code หรือ Binary Installer ของ MySQL ก็ให้เข้าไปที่ <http://www.mysql.com/downloads/index.html> ซึ่งเลือกได้ตามระบบปฏิบัติการที่คุณใช้ตามปกติการรัน MySQL บน Unix เช่น Solaris, FreeBSD หรือ Linux จะได้รับประสิทธิภาพดีที่สุด ส่วน MySQL Client (GUI) นั้นส่วนใหญ่เราก็มักจะใช้กันบน Windows Platform สิ่งที่สามารถดาวน์โหลดได้ก็คือต้องนำมาทำการคอมไพล์ (Compile) เอง แต่ถ้าเลือก Binary Installer ที่ได้รับการจัดเตรียมไว้สำหรับระบบปฏิบัติการหนึ่งๆ แล้ว ก็ไม่ต้องเปลืองแรงเรา สามารถนำมาใช้งานได้ทันที รายละเอียดการติดตั้งจะได้อธิบายต่อไป



ดาวน์โหลด MySQL

ติดตั้ง MySQL สำหรับ Windows Platform

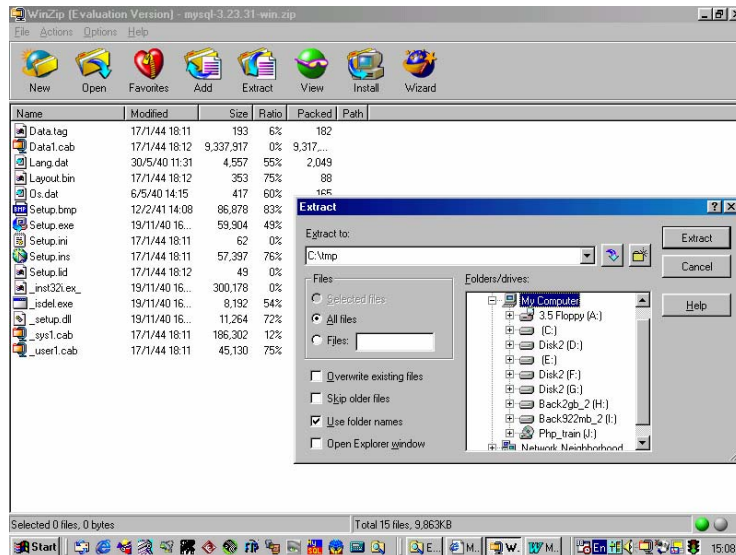
สำหรับ Windows Platform เมื่อดาวน์โหลดไฟล์ติดตั้งมาเรียบร้อยแล้ว ให้ทำตามขั้นตอนดังนี้

1. unzip ไฟล์ที่ดาวน์โหลดมาภายใต้ใคร่ทอริใดๆ (ตามปกติแล้วจะเป็น Temporary Directory เช่น C:\temp หรือ C:\windows\temp)
2. เข้าไปที่ Temporary Directory นั้น จากนั้นทำการรัน setup.exe

- เมื่อให้ระบบไคลเอนท์ที่ต้องการติดตั้งก็ให้ใช้ค่าดีฟอลต์คือ `C:\mysql`

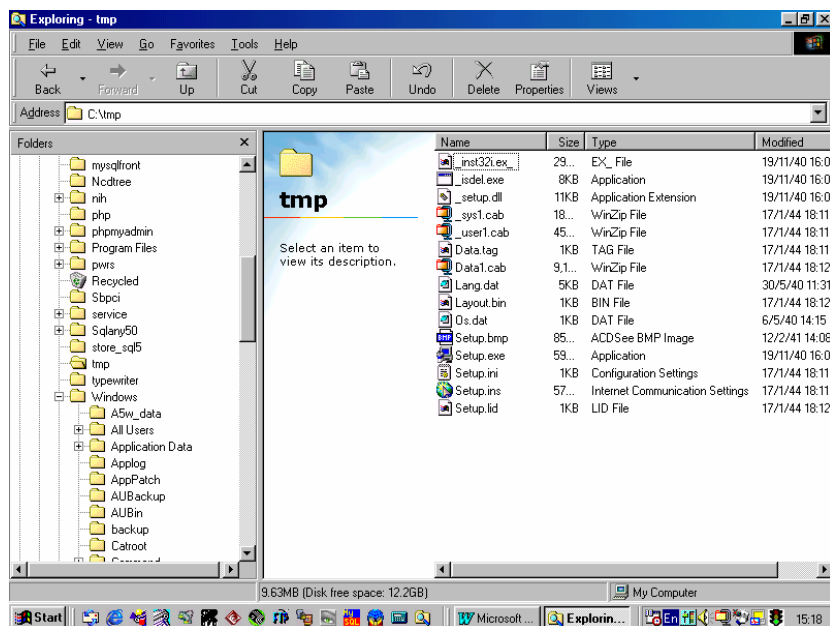
วิธีการติดตั้งจากแผ่นซีดี

- ดับเบิลคลิก `mysql-3.23.31-win.zip` จากซีดี เพื่อ Extract ลงที่ `C:\tmp`



- ดับเบิลคลิกไฟล์ `C:\tmp\Setup.exe` ติดตั้งลงที่ `C:\mysql`

- ดับเบิลคลิกไฟล์ `C:\mysql\bin\winmysqladmin.exe` เพื่อ Start MySQL



หมายเหตุ

- คัดลอกไฟล์ `mysql\my-example.cnf` ไปไว้ที่ราก `C:\` และเปลี่ยนชื่อเป็น `my.cnf`
- ปรับแต่งค่าต่างๆ ในไฟล์ `my.cnf` ให้เหมาะสม เช่นการแก้ปัญหาในเรื่องการ sort ภาษาไทย ใช้ editor notepad หรืออะไรก็ได้ find หาคำว่า `default-character-set=latin1` แล้วเปลี่ยน `latin1` เป็น `tis620`
- shutdown เครื่องประมาณ 2 นาที แล้วเปิดเครื่องใหม่

หรือ เปิดไฟล์ `my.ini` ใน Windows หาคำว่า `default-character-set=latin1` แล้วเปลี่ยน `latin1` เป็น `tis620`

คำสั่งต่างๆ ใน MySQL

ในบทนี้จะได้เรียนรู้เกี่ยวกับคำสั่งต่างๆ ใน MySQL ซึ่งเป็นการรวมคำสั่งทั้งหมด ไม่ว่าจะเป็นคำสั่งที่ใช้สำหรับการกำหนด (Data Definition Language), คำสั่งที่ใช้สำหรับทำงานข้อมูล (Data Manipulation Language) และคำสั่งที่ใช้สำหรับการควบคุมการทำงาน (Data Control Language)

คำสั่ง CREATE DATABASE

```
CREATE DATABASE demo
```

เป็นคำสั่งสร้าง database ชื่อ `db_name` ขึ้นมาใหม่ ในกรณีที่มี database นั้นอยู่แล้ว ก็จะปรากฏข่าวสารแสดงความผิดพลาด (Error Message) `Can't create database 'demo'. Database exists`

คำสั่ง DROP DATABASE

```
DROP DATABASE demo
```

เป็นคำสั่งลบ database ชื่อ `db_name` ออกจากระบบ

คำสั่ง CREATE TABLE

```
CREATE TABLE phonebook (id INT (4) NOT NULL  
AUTO_INCREMENT, fname VARCHAR (35) , lname VARCHAR (35) ,  
email VARCHAR (50) , office VARCHAR (200) , province CHAR (2) ,  
PRIMARY KEY(id), UNIQUE(id), INDEX(id))
```

โดยทั่วไปขนาดของตารางข้อมูลหนึ่งๆ จะมีขนาดไม่เกิน 2G หรือ 4G ขึ้นอยู่กับระบบปฏิบัติการ เราสามารถทำให้ประหยัดเนื้อที่ได้อีกโดยการบีบอัด (PACK) ข้อมูลดัชนี ซึ่งสามารถให้บีบอัดโดยกำหนดค่า PACK_KEYS=1 การบีบอัดนี้จะทำให้การทำงานช้าลง แต่ก็ประหยัดเนื้อที่ขึ้น และตามปกติถ้าผู้ใช้ไม่ได้กำหนดให้บีบอัด ข้อมูลที่เป็นสตริงหรือตัวอักษรก็จะได้รับการบีบอัดข้อมูลอยู่แล้ว สำหรับข้อมูลตัวเลข ค่าการทำงานปกติจะไม่รับการบีบอัด เว้นแต่กำหนดการบีบอัดด้วย

การใช้ UNION จะใช้กรณีที่ต้องการรวมตารางข้อมูลหลายๆ ตารางเข้าด้วยกันเป็นอันเดียว ซึ่งจะทำงานกับตารางประเภท MERGE เท่านั้น

คำสั่ง ALTER TABLE

```
ALTER TABLE phonebook ADD tel VARCHAR(30)
```

ALTER TABLE เป็นคำสั่งสำหรับการปรับเปลี่ยนแก้ไข โครงสร้างของตารางข้อมูล เช่น การเพิ่ม-ลบฟิลด์, การสร้าง-ลบดัชนี, การเปลี่ยนประเภทข้อมูลของฟิลด์, การเปลี่ยนชื่อฟิลด์ เป็นต้น

คำสั่ง RENAME TABLE

```
RENAME TABLE phonebook to phone
```

เป็นคำสั่งเพื่อเปลี่ยนชื่อตารางข้อมูลเป็นชื่อใหม่

คำสั่ง DROP TABLE

```
DROP TABLE phonebook
```

เป็นคำสั่งเพื่อลบตารางข้อมูลออกจากระบบ สำหรับอปชัน RESTRICT และ CASCADE ยังไม่สามารถงานได้ในปัจจุบัน เนื่องจากเป็นความสามารถที่มีแผนจะรองรับในอนาคต สำหรับการใช้คำสั่ง DROP TABLE จะต้องให้ความระมัดระวังในการทำงานเป็นอย่างยิ่ง โดยเฉพาะการทำงานกับตารางข้อมูลประเภท non-transaction เพราะจะทำการ commit การทำงานโดยอัตโนมัติ

คำสั่ง OPTIMIZE TABLE

```
OPTIMIZE TABLE phonebook
```

เป็นคำสั่งสำหรับจัดระเบียบข้อมูลให้เป็นระเบียบ หรือการทำดีแฟร็กเมนต์นั่นเองนั่นเอง เพราะถ้าตารางข้อมูลมีการใช้งานมานาน มีการใช้งานที่ทำให้ข้อมูลมีการเปลี่ยนแปลงความยาว หรือเปลี่ยนขนาดของเนื้อที่เก็บข้อมูล โดยเฉพาะกับข้อมูลประเภทที่ยืดหยุ่นไปตามข้อมูล เช่น VARCHAR, BLOB หรือ TEXT การใช้คำสั่ง OPTIMIZE TABLE ก็จะช่วยให้มีการจัดระเบียบข้อมูลให้ดียิ่งขึ้น หากปล่อยระยะเวลาต่างๆ จะเกิดช่องว่างระหว่างข้อมูล การอ่านข้อมูลของสื่อจะไม่ราบรื่น ซ้ำ เนื่องจากเสียเวลาในการกระโดดข้ามช่องว่างนั้นนั่นเอง

คำสั่ง DELETE

```
DELETE FROM phonebook WHERE id=4
```

เป็นคำสั่งสำหรับลบข้อมูลจากตาราง โดยสามารถกำหนดเงื่อนไขสำหรับการลบได้ ซึ่งถ้าไม่ได้ระบุเงื่อนไข จะเป็นการลบข้อมูลทั้งหมดออกจากตาราง และถ้าอยู่ใน AUTOCOMMIT mode (กรณีปกติทั่วไป) จะไม่สามารถทำการยกเลิกการลบครั้งนี้ได้

คำสั่ง TRUNCATE

```
TRUNCATE phonebook
```

เป็นคำสั่งที่ให้ผลเช่นเดียวกับ DELETE FROM table_name แต่มีข้อแตกต่างดังนี้

- จะทำการครอบตารางข้อมูลก่อน แล้วทำการสร้างใหม่ ซึ่งจะให้ผลที่เร็วกว่าการลบตารางทั้งหมด
- การทำงานเป็น non-transaction ไม่สามารถทำการยกเลิกได้

คำสั่ง SELECT

```
SELECT id, fname FROM phonebook ORDER BY fname DESC
```

เป็นคำสั่งสำหรับการแสดงผล หรือการดึงข้อมูลจากตาราง โดยสามารถกำหนดเงื่อนไข รูปแบบและวิธีการได้อย่างหลากหลาย การดึงข้อมูลสามารถระบุว่าจะเลือกจากตารางเดียว หรือหลายตารางก็ได้

การใช้ LIMIT rows ก็จะช่วยกำหนดให้ทำการดึงข้อมูลจากตาราง ได้ไม่เกินจำนวนแถว rows ที่ระบุ การใช้คำสั่งคีย์เวิร์ดตามหลัง SELECT จะต้องใช้ให้ถูกต้อง มีบางคีย์เวิร์ดที่ต้องใช้เรียงลำดับกัน เช่น HAVING clause ต้องใช้ตามหลัง GROUP BY clause และใช้ก่อน ORDER BY clause เป็นต้น

ตามปกติหากไม่ได้ระบุวิธีการจัดเรียงข้อมูล ก็จะเป็นการเรียงจากน้อยไปมาก (Ascending) หรือ คีย์เวิร์ด ASC แต่ถ้าต้องการเรียงจากมากไปน้อย (Descending) ให้ใช้คีย์เวิร์ด DESC

การใช้คีย์เวิร์ด **HAVING** จะใช้ในกรณีร่วมกับ **GROUP BY** ซึ่งก็เป็นการกำหนดเงื่อนไขในการดึงข้อมูล ด้วยคำสั่ง **SELECT**

```
SELECT id, fname FROM phonebook GROUP BY fname HAVING id>2
```

คำสั่ง **INSERT**

```
INSERT INTO phonebook (id, fname, lname, email, office, province)
VALUES (6, 'sineenat', 'phradmali', 'sineenat@health.moph.go.th', 'ศทส.', '12')
```

เป็นคำสั่งสำหรับการเพิ่มแถวข้อมูล หรือระเบียบใหม่เข้าตารางข้อมูล

คำสั่ง **UPDATE**

```
UPDATE phonebook SET office= 'ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร' WHERE id=6
```

เป็นคำสั่งเพื่อทำการเปลี่ยนแปลงข้อมูลในตารางให้เป็นตามที่ต้องการ

คำสั่ง **USE**

```
USE demo
```

USE เป็นคำสั่งสำหรับเลือกใช้ database **SHOW TABLES** แสดงตารางที่อยู่ใน database demo ทั้งหมด

คำสั่ง **FLUSH**

```
FLUSH TABLES phonebook
```

เป็นคำสั่งที่ใช้สำหรับการลบหน่วยความจำภายใน (Internal Cache) ที่ใช้งาน ทั้งนี้การใช้คำสั่ง **FLUSH** จะอนุญาตเฉพาะ user ที่มีสิทธิ์ (Privilege) เท่านั้น

Flush_option สามารถระบุได้ดังนี้

HOSTS ทำการล้าง host cache table เพื่อปรับปรุงข้อมูลของ host หรือ client ที่คอนเน็คต์เข้าใช้งาน ซึ่งอาจมีการเปลี่ยนแปลง IP address หรืออื่นๆ โดยเมื่อทำการ flush เรียบร้อยแล้ว ก็จะอนุญาตให้ host นั้นๆ ทำการ

	คอนเน็คต์เข้ามาใหม่อีกครั้งหนึ่ง
LOGS	ปิดและเปิด log files ใหม่
PRIVILEGES	ทำการ reload สิทธิต่างๆ ของ user จาก grant tables ของ MySQL Server
TABLES	ปิด tables ทั้งหมด หากตารางใดยังถูกเปิดอยู่ ก็จะถูก force ให้ปิดลง

คำสั่ง SHOW

SHOW DATABASES

เป็นคำสั่งสำหรับแสดงรายละเอียด (Information) ของ databases, tables, columns หรือ สถานะต่างๆ ของ Server

แสดงรายละเอียดของ Databases, Tables, Columns และ Indexes

SHOW DATABASES	แสดงรายชื่อ databases ทั้งหมดในระบบฐานข้อมูล
SHOW TABLES	แสดงรายชื่อ tables ทั้งหมดของ current database
SHOW OPEN TABLES	แสดงรายชื่อ tables ที่ถูกเปิดอยู่ในปัจจุบัน
SHOW COLUMNS	แสดงรายละเอียดฟิลด์ทั้งหมดใน table นั้น (ให้ผลลัพธ์เช่นเดียวกับคำสั่ง DESCRIBE)
SHOW FIELDS	ให้ผลลัพธ์เช่นเดียวกับ SHOW COLUMNS
SHOW INDEX	แสดงรายละเอียดว่ามีดัชนีอะไรบ้างสำหรับ table นั้นๆ
SHOW KEYS	ให้ผลลัพธ์เช่นเดียวกับ SHOW INDEX

คำสั่ง GRANT และ REVOKE

```
GRANT ALL PRIVILEGES ON demo TO user1@localhost WITH GRANT OPTION
```

```
REVOKE ALL PRIVILEGES ON demo FROM user1@localhost
```

- คำสั่ง **GRANT** เป็นคำสั่งสำหรับการกำหนดสิทธิ์หรือความสามารถใดๆ ให้กับ user
- คำสั่ง **REVOKE** หรือคำสั่งที่ตรงกันข้ามกับ GRANT คือเป็นการถอนหรือยกเลิกสิทธิ์ใดๆ จาก user

การใช้คำสั่ง GRANT หรือ REVOKE จะอนุญาตเฉพาะผู้บริหารระบบในการกำหนดสิทธิ์ต่างๆ สำหรับผู้ใช้

MySQL จะแบ่งสิทธิ์ (Privilege) เป็น 4 ระดับ คือ

1. Global level

ระดับบนสุด ครอบคลุมทุก database รายละเอียดสำหรับสิทธิ์ของระดับนี้จะถูกเก็บไว้ในตาราง mysql.user

2. Database level

ระดับ database ครอบคลุมเฉพาะใน database อันใดอันหนึ่ง เช่น ความสามารถในการกระทำใดๆ ภายใน database นั้นๆ รายละเอียดสำหรับสิทธิ์ของระดับนี้จะถูกเก็บไว้ในตาราง mysql.db และ mysql.host

3. Table level

ระดับ table ครอบคลุมเฉพาะภายในตารางนั้นๆ เช่น ความสามารถในการกระทำกับ table นั้นๆ สามารถทำการ select, insert, delete, update ได้หรือไม่ เป็นต้น รายละเอียดสำหรับสิทธิ์ของระดับนี้จะถูกเก็บอยู่ในตาราง mysql.tables_priv

4. Column level

ระดับ column ครอบคลุมเฉพาะฟิลด์ใดฟิลด์หนึ่งภายในตารางข้อมูล รายละเอียดสำหรับสิทธิ์ของระดับนี้จะถูกเก็บอยู่ในตาราง mysql.columns_priv

รายละเอียด priv_type มีดังต่อไปนี้

ALL PRIVILEGES	FILE	RELOAD
ALTER	INDEX	SELECT
CREATE	INSERT	SHUTDOWN
DELETE	PROCESS	UPDATE
DROP	USAGE	

- เราสามารถใช้ ALL หรือ ALL PRIVILEGES ได้อย่างใดอย่างหนึ่ง
- USAGE มีความหมายเท่ากับ no privileges โดยใช้กรณีที่เราต้องการ create user และกำหนดให้มีสิทธิ์เป็น no privileges
- การถอนสิทธิ์การ GRANT ของ user ใดๆ ให้ใช้ priv_type เป็น GRANT OPTION

ตัวอย่าง: REVOKE GRANT OPTION ON ... FROM ...;

- การใช้ priv_type สำหรับ table: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, GRANT, INDEX และ ALTER
- การใช้ priv_type สำหรับฟิลด์ SELECT, INSERT, UPDATE
- การกำหนดสิทธิ์ระดับ database ใช้ ON *.*
- สำหรับ user_name สามารถระบุได้ถึงโฮสต์ที่ใช้งาน เช่น mysql@localhost หรือ root@'%'

การรักษาความปลอดภัย

การกำหนดสิทธิ์ของผู้ใช้งาน

หลักการที่ MySQL ใช้ เรียกว่า Access Control Lists (ACLs) จะมีการตรวจสอบสิทธิ์ผู้ใช้งาน ว่าแต่ละคนมีสิทธิ์จะทำอะไรได้บ้าง กับฐานข้อมูลใดบ้าง ซึ่งสิทธิ์เหล่านี้จะถูกบันทึกเอาไว้ในตารางชื่อ mysql

ข้อควรกระทำและไม่ควรกระทำในการป้องกันหรือรักษาความปลอดภัยของข้อมูล

1. ห้ามอนุญาตให้ใครก็ตาม ที่ไม่ใช่ผู้ดูแลระบบ มีสิทธิ์เข้าไปเรียกดูหรือใช้งานตารางชื่อ mysql

เพราะ mysql เป็นแหล่งที่เก็บข้อมูลเกี่ยวกับสิทธิ์ต่าง ๆ

2. อย่างกำหนดสิทธิ์ให้กับผู้ใช้งานใด ๆ มากเกินความจำเป็น

ผู้ใช้แต่ละคนควรจะได้รับสิทธิ์แตกต่างกัน และควรมีสิทธิ์จำกัดเป็นกรณี ๆ ไป เช่น หากเป็นผู้ใช้ทั่ว ๆ ไป ควรจะมีสิทธิ์ใช้งานแค่ฐานข้อมูลใดฐานข้อมูลหนึ่ง หรือ อาจจะมีสิทธิ์แค่ตารางใดตารางหนึ่งในแต่ละฐานข้อมูล

ระดับสิทธิ์การเรียกใช้ข้อมูลมี 4 ระดับคือ

- 2.1 ระดับโฮสต์ (host level)
- 2.2 ระดับฐานข้อมูล (db level)
- 2.3 ระดับตารางข้อมูล (table level)
- 2.4 ระดับคอลัมน์ข้อมูล (column level)

3. ระวังผู้ใช้ที่ใช้ชื่อว่า root

root คือ ผู้สร้างฐานข้อมูล มีสิทธิ์ต่าง ๆ มาก ข้อแนะนำว่าอย่าให้มีผู้ใช้ชื่อ root อยู่ในสารบบของ MySQL โดยเราสามารถกำหนดผู้ใช้นี้ขึ้นมาให้มีสิทธิ์เทียบเท่า root และอีกประการคือ รายชื่อผู้ใช้ของ MySQL กับของ Server ที่ใช้งาน จะเป็นคนละชุดกัน

4. ระวังการตั้งรหัสผ่านของผู้ใช้

การตั้งรหัสผ่านควรหลีกเลี่ยงคำศัพท์ทั่วไปที่สามารถค้นหาได้จากดิกชันนารี, หลีกเลี่ยงชื่อเล่น หรือ ชื่อสถานที่ที่ผู้บุกรุกอาจจะคาดเดาได้

5. หากไม่จำเป็นต้องใช้ Telnet ก็ปิดไปดีกว่า

ในกรณีที่ Server ของเราสามารถรองรับการทำงานของโปรแกรม Telnet แต่ในทางปฏิบัติไม่มีความจำเป็นที่ ต้องอนุญาตให้ใครมาใช้โปรแกรมนี้ ก็ควรยกเลิกการทำงาน

6. อย่าเชื่อมือผู้ใช้นัก

เพราะหากผู้ใช้ผลอไปใช้คำสั่ง Drop database mysql ถ้าเราพัฒนาโปรแกรมขึ้นมาใช้ ควรจะมีระบบ ตรวจสอบคำสั่งนี้ โดยห้ามใช้คำสั่งนี้เด็ดขาด

ระดับสิทธิ์ของผู้ใช้

1. ระดับโฮสต์ (host level) คือ ระดับนี้จะจำกัดสิทธิ์ว่า ผู้ใช้สามารถติดต่อเข้ามายัง MySQL ได้จากที่ ไหนบ้าง หรือจาก โฮสต์อะไร ทั้งนี้เพราะถ้า MySQL ของเราอยู่ในเครื่องคอมพิวเตอร์ที่เชื่อมโยงกับเครื่องอื่น ๆ หรือเชื่อมโยงกับเครือข่าย Internet การติดต่อสื่อสารก็ย่อมกระทำจากที่ไหนก็ได้ในโลก ผ่านทางโพรโทคอล TCP/IP

ลักษณะการตรวจสอบสิทธิการใช้งาน เมื่อเครื่องลูกข่ายที่ร้องขอติดต่อเข้ามา จะต้องแจ้งให้ทราบว่ามาจากไหน ชื่อโฮสต์อะไร หมายเลข IP Address เท่าไร เมื่อ MySQL รับทราบการร้องขอ ก็จะนำข้อมูลที่รับแจ้งไป ตรวจสอบว่าโฮสต์นั้นมีสิทธิหรือไม่หรือมีสิทธิอะไรบ้าง
ลักษณะชื่อโฮสต์ที่สามารถกำหนดได้

ลักษณะชื่อโฮสต์ที่สามารถกำหนดได้	หมายความว่าติดต่อมาจาก...
localhost	เครื่องเดียวกันกับ MySQL server
%	เครื่องไหนก็ได้
%.cc-moph	เครื่องใดก็ตามที่อยู่ใน โดเมน cc-moph
203.157%	เครื่องใดก็ตามที่มี IP address เริ่มจาก 203.157

ตัวอย่าง กำหนดสิทธิให้กับผู้ใช้ที่ชื่อว่า train ก็ระบุในลักษณะ [train@host](#) เช่น อนุญาตให้ train สามารถติดต่อมายัง MySQL ได้จากเครื่องหมายเลข 203.157.4.27 ก็ระบุเป็น [train@203.157.4.27](#) ดังนั้นถ้าติดต่อ MySQL โดยใช้สคริปต์ PHP ก็ต้องเขียนคำสั่งในรูปแบบดังนี้

```
mysql_connect("MySQLHost","train","รหัสผ่าน");
```

คำสั่งกำหนดสิทธิการใช้งาน GRANT มีรูปแบบดังนี้

```
GRANT สิทธิต่าง ๆ ON ชื่อฐานข้อมูล.ชื่อตาราง TO ชื่อผู้ใช้งาน@โฮสต์[IDENTIFILE BY 'รหัสผ่าน']
```

```
GRANT ALL ON *.* TO train@203.157.4.27 IDENTIFILE BY 'aBcXyz';
```

สิทธิต่าง คือ สิทธิ (privilege) ที่จะสามารถทำได้เช่น การเรียกดูข้อมูล (SELECT), การลดข้อมูล(DELECT) หรือ จะระบุสิทธิต่าง ๆ เป็น ALL ก็ได้ซึ่งหมายความว่าสามารถทำได้ทุกอย่าง

2. ระดับฐานข้อมูล (db level)

ทั้งนี้ขึ้นอยู่กับว่าการระบุสิทธิต่าง ๆ ตามหลังคำสั่ง GRANT ถ้าระบุคำว่า ALL ก็หมายความว่าให้สิทธิในการเรียกใช้ทุกคำสั่ง เช่น

```
GRANT SELECT ON *.* TO User1@localhost IDENTIFILE BY 'Password';
```

หมายความว่า อนุญาตให้ user1 สามารถใช้คำสั่ง SELECT สำหรับเรียกดูข้อมูลทุกตารางทุกฐานข้อมูล

3. ระดับตารางข้อมูล (table level)

สิทธิระดับนี้จะสัมพันธ์กับสิทธิระดับฐานข้อมูล คือ ถ้าจะให้สิทธิในการใช้คำสั่งกับทุก ๆ ตารางในฐานข้อมูลก็ระบุด้วยเครื่องหมาย * เช่น

GIS.* หมายถึงอนุญาตให้มีสิทธิใช้คำสั่งกับทุกตารางที่มีอยู่ในฐานข้อมูลชื่อ GIS

```
GRANT SELECT ON db1.table1 TO User1@localhost IDENTIFILE BY 'aBcXyz';
```

หมายความว่าอนุญาตให้ user1 เรียกใช้เฉพาะคำสั่ง SELECT เพื่อดึงข้อมูลจากตารางชื่อ table1 ซึ่งอยู่ในฐานข้อมูล db1 เท่านั้น

4. ระดับคอลัมน์ข้อมูล (Column level)

เป็นสิทธิระดับที่ลดลงไปจากสิทธิระดับอื่น ๆ อีก โดยมีความสัมพันธ์กับสิทธิระดับฐานข้อมูลและระดับตารางด้วย เพียงแต่วิธีการกำหนดสิทธิต้องระบุในตารางที่ชื่อ column_priv ซึ่งตารางนี้อยู่ในฐานข้อมูล mysql อีกที ฉะนั้นจึงต้องใช้คำสั่ง SQL ดังไปตรง ๆ ที่ตาราง column_priv มิใช่กำหนดจากการใช้คำสั่ง GRANT เหมือนกับสิทธิระดับอื่น ๆ

สิทธิต่าง ๆ หรือ **privilege** หมายความว่า เมื่อเราอนุญาตให้ผู้ใช้เข้ามายังฐานข้อมูล, ตารางข้อมูล, หรือเข้ามายัง MySQL ได้แล้ว สามารถใช้คำสั่ง SQL ทำอะไรได้บ้าง

สิทธิต่าง ๆ	การทำงาน	ตัวอย่างการใช้งาน
SELECT	เรียกดูรายการข้อมูล	SELECT * FROM table1
INSERT	เพิ่มรายการข้อมูล	INSERT INTO table1 ...
UPDATE	ปรับปรุงแก้ไขรายการข้อมูล	UPDATE table1 SET ...
DELETE	ลบรายการข้อมูล	DELETE FROM table1
ALTER	แก้ไขโครงสร้างตารางข้อมูล	ALTER TABLE table1 ADD INDEX index1(name)
CREATE	สร้างตารางหรือฐานข้อมูลขึ้นมาใหม่	CREATE DATABASE database1 CREATE TABLE table1
DROP	ลบตารางหรือฐานข้อมูล	DROP DATABASE database1 DROP TABLE table1

สิทธิต่างๆ	การทำงาน	ตัวอย่างการใช้งาน
RELOAD	สั่งให้ server เช่นค่าต่าง ๆ ใหม่	FLUSH HOST (ปลดล็อกในกรณีที่ server มีการล็อก ป้องกันการใช้งานบางอย่าง เช่น ป้องกันมิให้มีผู้ใช้งานพร้อมกันมาเกิดกำหนด)
SHUTDOWN	อนุญาตให้สั่งปิดการทำงานของ server	Mysqldadmin shutdown
PROCESS	ใช้คำสั่งพิเศษกับ server เช่น SHOW	SHOW PROCESSLIST (ขอให้แสดงรายการ โพรเซสที่กำลังทำงานอยู่)
FILE	อ่าน – เขียน ไฟล์ที่ server ได้	LOAD DATA INFILE 'data.txt' INTO TABLE table1 (โหลดข้อมูลจากไฟล์ชื่อ data.txt ไปบันทึกลงในตารางข้อมูล table1)

เมื่อคุณ install MySQL เสร็จ สิ่งแรกที่คุณควรทำ ก็คือการกำหนด password สำหรับ root user โดยวิธีการ ดังนี้

```
shell> mysql -u root
mysql> use mysql;
mysql> UPDATE user SET Password=PASSWORD('new_password') WHERE user='root';
mysql> FLUSH PRIVILEGES;
```

วิธีการกำหนด password

จากตัวอย่างในหัวข้อที่ผ่านมาจะเห็นได้ว่าเมื่อคุณ INSERT หรือ UPDATE password คุณต้องใช้ function PASSWORD() เพื่อเข้ารหัส password เนื่องจากการกระทำ เช่นนี้ ทำให้ password ถูกเก็บลง table ในแบบที่ถูกเข้ารหัส เพื่อความปลอดภัยของข้อมูล หากคุณลืมใช้ function PASSWORD() , password ก็จะถูกเก็บในรูปแบบของตัวหนังสือ (plaintext)

ฐานข้อมูล MySQL

MySQL มองไฟล์เตอร์ใน C:\mysql\data เป็นฐานข้อมูล

สำหรับไฟล์ทั้งหลายที่อยู่ในไฟล์เตอร์ย่อยๆ นั้น ก็คือ ไฟล์ที่อยู่ในฐานข้อมูลนั้น โคนจำแนกชนิดของไฟล์ตามนามสกุล 3 ชนิดได้แก่

MYD หมายถึง ไฟล์ที่ใช้เก็บข้อมูล ไฟล์นี้ก็คือตารางข้อมูลนั่นเอง เช่น **phonebook.MYD** หมายถึง ตารางข้อมูลชื่อ **phonebook**

MYI หมายถึง ไฟล์ดัชนี หรือ index file เพื่อใช้เก็บค่าข้อมูลจากคอลัมน์ที่มีสถานะเป็นคีย์หรือดัชนี (index) และเก็บค่าพอยน์เตอร์ (pointer) ด้วย เวลาค้นหาข้อมูล โปรแกรม MySQL จะค้นหาจากไฟล์นี้แทนที่จะค้นหาจากตาราง (ไฟล์ .MYD) ตรงๆ เพราะไฟล์ดัชนีมีขนาดเล็กกว่า มีวิธีการจัดเก็บข้อมูลที่เอื้อต่อการเรียกค้นหา มากกว่า จึงค้นได้เร็วกว่า

เมื่อได้ค่าที่ต้องการแล้ว MySQL จะอาศัยพอยน์เตอร์อ้างอิงไปยังตำแหน่งแถวรายการข้อมูลที่เกี่ยวข้องกับดัชนี ซึ่งอยู่ในไฟล์ข้อมูลจริงหรือ MYD อีกที

เวลามีการแก้ไขข้อมูล หรือลบข้อมูล รวมทั้งเมื่อมีการเพิ่มรายการข้อมูลใหม่เข้าไป ทำให้ค่าดัชนีมีการเปลี่ยนแปลง ข้อมูลในไฟล์ดัชนีก็จะถูกจัดเรียงลำดับใหม่ตามข้อมูลจริงที่เปลี่ยนไปด้วย

frm หมายถึง ไฟล์ที่เก็บรายละเอียดโครงสร้างของตารางข้อมูล ว่าประกอบด้วยคอลัมน์อะไรบ้าง และเป็นคอลัมน์ชนิดไหน ดังนั้นสมมติว่าเราสร้างตารางข้อมูลขึ้นมาหนึ่งตาราง เช่น ตารางชื่อ **phonebook.MYD** **phonebook.MYI** และ **phonebook.frm** โดยแต่ละไฟล์จะทำหน้าที่ตามที่อธิบายไปแล้ว

Reserved Word:

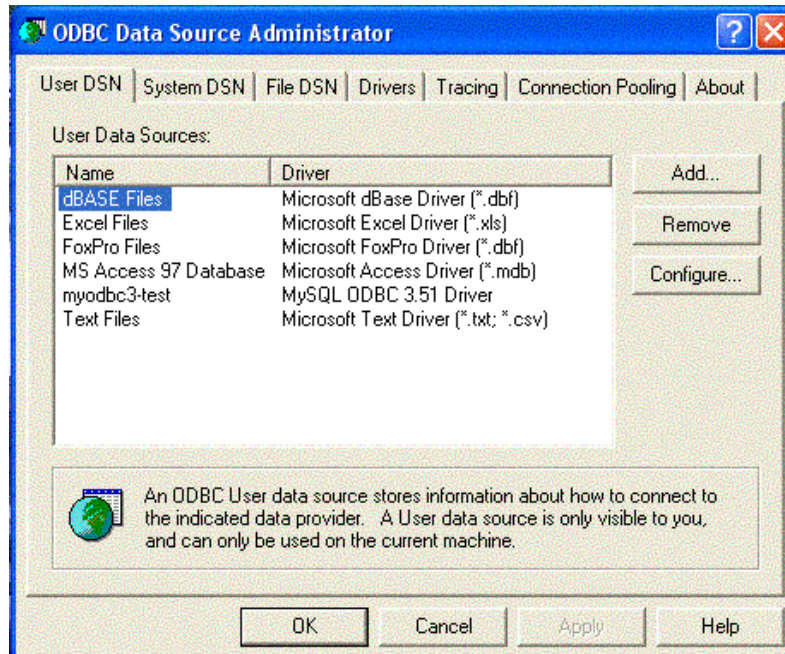
MySQL มีคำที่สงวนไว้ ซึ่งคุณไม่สามารถนำคำเหล่านี้ไปตั้งเป็นชื่อตารางหรือ column ได้

action	add	aggregate	all
alter	after	and	as
asc	avg	avg_row_length	auto_increment
between	bigint	bit	binary
blob	bool	both	by
cascade	case	char	character
change	check	checksum	column
columns	comment	constraint	create
cross	current_date	current_time	date
datetime	day	day_hour	day_minute
day_second	dayofmonth	dayofweek	dayofyear
dec	decimal	default	delayed
delay_key_write	delete	desc	describe
distinct	distinctrow	double	drop
end	else	escape	escaped
enclosed	enum	explain	exists
fields	file	first	float
float4	float8	flush	foreign
from	for	full	function
global	grant	grants	group
having	heap	high_priority	hour
hour_minute	hour_second	hosts	identified
ignore	in	index	infile
inner	insert	insert_id	int
integer	interval	int1	int2
int3	int4	int8	into
if	is	isam	join

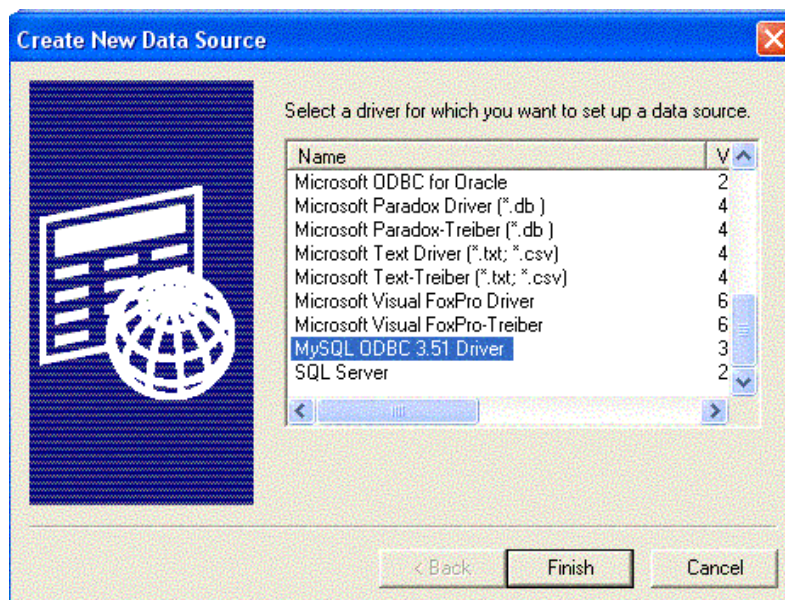
key	keys	kill	last_insert_id
leading	left	length	like
lines	limit	load	local
lock	logs	long	longblob
longtext	low_priority	max	max_rows
match	mediumblob	mediumtext	mediumint
middleint	min_rows	minute	minute_second
modify	month	monthname	myisam
natural	numeric	no	not
null	on	optimize	option
optionally	or	order	outer
outfile	pack_keys	partial	password
precision	primary	procedure	process
processlist	privileges	read	real
references	reload	returns	revoke
rlike	row	rows	second
select	set	show	shutdown
smallint	soname	sql_big_tables	sql_big_selects
sql_log_off	sql_log_update	sql_select_limit	sql_low_priority_updates
sql_small_result	sql_big_result	sql_warnings	straight_join
tables	temporary	terminated	test
then	time	timestamp	tinyblob
tinytext	tinyint	trailing	to
type	use	using	unique
unlock	unsigned	update	usage
values	varchar	variables	varying
varbinary	with	write	when
where	year	year_month	zerofill

การ install MyODBC บน Windows

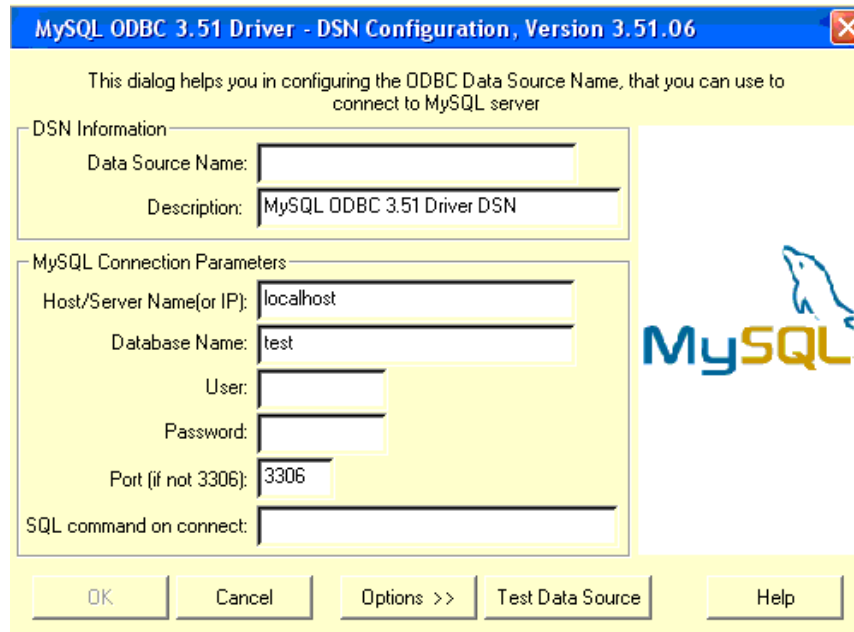
1. double click ที่ file MyODBC-3.51.06.exe เพื่อ install program
2. ไป Data Sources (ODBC)



3. ที่ Tab User DSN คลิกปุ่ม Add เลือก MySQL ODBC 3.51 Driver และกดปุ่ม Finish



4. Windows DSN name : ให้ใส่ชื่อของ ODBC ที่สร้างขึ้น ตั้งชื่อ Data Source Name:



5. Host/Server Name (or IP): ใส่ชื่อเครื่อง หรือ IP ของเครื่องที่เป็น MySQL server
6. Database Name: ใส่ชื่อ database ที่ต้องการติดต่อด้วย
7. User: ใส่ชื่อ user เพื่อ login MySQL server
8. Password: ใส่รหัสผ่าน
9. Port: port ที่ต้องการติดต่อด้วย ไม่จำเป็นต้องใส่ (default = 3306)
10. กดปุ่ม OK จากนั้นจะเห็น Data Sources Windows อีกครั้ง ให้กดปุ่ม OK

การใช้ Microsoft Access เชื่อมต่อกับ MySQL

วิธีการเชื่อมโยงข้อมูลกันระหว่าง Access กับ MySQL จะใช้ ODBC เป็นตัวเชื่อม ซึ่งแบ่งได้เป็น 2 กรณี คือ

1. การ Link table

สามารถดึงข้อมูลจาก MySQL มาดูใน access ได้ ข้อมูลที่อยู่ใน MySQL จะแสดงให้ดูในรูปแบบของข้อมูลใน Access ด้วยวิธีการดังนี้

- สร้าง DSN ที่สำหรับ database ของ MySQL ที่คุณต้องการติดต่อด้วย
- เปิดโปรแกรม Access เลือก database ที่ต้องการจะ link table มาจาก MySQL
- click ที่ file -> get external data -> Link Table...
- เลือก File of type เป็น ODBC Database
- เลือก tab Machine Data Source
- จะเห็น ODBC ที่มีให้เลือก เลือก DSN ที่ต้องการ
- เลือก table ที่ต้องการ Link เข้ามาใน Access

2. การ import/ export table

การ import/export นี้ เปรียบเสมือนการคัดลอกข้อมูลระหว่าง Microsoft Access กับ MySQL ซึ่งวิธีการเรียกใช้ ODBC ก็เหมือนกับวิธีการ Link table