

## ชนิดของข้อมูลใน MySQL

### ข้อมูลตัวเลข:

**TINYINT** ขนาดที่เก็บ 1 ไบต์

UNSIGNED เก็บค่าจำนวนเต็ม 0 ถึง 255

SIGNED เก็บค่าจำนวนเต็ม -128 ถึง 127

**SMALLINT** ขนาดที่เก็บ 2 ไบต์

UNSIGNED เก็บค่าจำนวนเต็ม 0 ถึง 65535

SIGNED เก็บค่าจำนวนเต็ม -32768 ถึง 32767

**MEDIUMINT** ขนาดที่เก็บ 3 ไบต์

UNSIGNED เก็บค่าจำนวนเต็ม 0 ถึง 16777215

SIGNED เก็บค่าจำนวนเต็ม -8388608 ถึง 8388607

**INT หรือ INTEGER** ขนาดที่เก็บ 4 ไบต์

UNSIGNED เก็บค่าจำนวนเต็ม 0 ถึง 4294967295

SIGNED เก็บค่าจำนวนเต็ม -2147483648 ถึง 2147483647

**BIGINT หรือ INTEGER** ขนาดที่เก็บ 8 ไบต์

UNSIGNED เก็บค่าจำนวนเต็ม 0 ถึง 18446744073709551615

SIGNED เก็บค่าจำนวนเต็ม -9223372036854775808 ถึง 9223372036854775807

(ขนาดที่เก็บของ INTEGER จะแปรตามค่าที่บันทึก)

**FLOAT(x)** ถ้า  $0 \leq x \leq 24$  ขนาดที่เก็บ 4 ไบต์ (ดู FLOAT)

ถ้า  $25 \leq x \leq 53$  ขนาดที่เก็บ 8 ไบต์ (ดู DOUBLE)

เก็บค่าจำนวนจริงแบบ IEEE 754

**FLOAT** ขนาดที่เก็บ 4 ไบต์

เก็บค่าจำนวนจริงแบบ IEEE 754

ตั้งแต่  $-3.402823466E+38$  ถึง  $-1.175494351E-38$

และ 0

และ  $1.175494351E-38$  ถึง  $3.402823466E+38$

**DOUBLE หรือ REAL** ขนาดที่เก็บ 8 ไบต์

เก็บค่าจำนวนจริงแบบ IEEE 754

ตั้งแต่  $-1.7976931348623157E+308$  ถึง  $-2.2250738585072014E-308$

และ 0

และ 2.2250738585072014E-308 ถึง 1.7976931348623157E+308

**DECIMAL(m,d)** หรือ **NUMERIC(m,d)** ถ้า  $d = 0$  ขนาดที่เก็บคือ  $m+1$  ไบต์ ถ้า  $d > 0$  ขนาดที่เก็บคือ  $m+2$  ไบต์

เก็บค่าเลขทศนิยมแบบระบุจำนวนหลัก  $m$  ทุกหลักรวมจุดทศนิยม และ  $d$  หลักหลังทศนิยม เช่นถ้าต้องการเก็บค่าให้ได้มากที่สุดเพียง 9999.99 ให้กำหนดเป็น **DECIMAL(7,2)**

### ข้อมูลวันเวลา:

**DATE** ขนาดที่เก็บ 3 ไบต์

เก็บค่าวันที่ในรูปแบบ YYYY-MM-DD

โดยมีค่าตั้งแต่ 1000-01-01 ถึง 9999-12-31

**DATETIME** ขนาดที่เก็บ 8 ไบต์

เก็บค่าวันที่และเวลาในรูปแบบ YYYY-MM-DD HH:mm:SS

โดยมีค่าตั้งแต่ 1000-01-01 00:00:00 ถึง 9999-12-31 23:59:59

**TIMESTAMP(m)** ขนาดที่เก็บ 4 ไบต์

เก็บวันที่และเวลาในรูปแบบ String Timestamp

$m = 14$  หรือไม่กำหนด -> YYYYMMDDHHmmSS

$m = 12$  -> YYMMDDHHmmSS

$m = 10$  -> YYMMDDHHmm

$m = 8$  -> YYYYMMDD

$m = 6$  -> YYMMDD

$m = 4$  -> YYMM

$m = 2$  -> YY

โดยมีค่าตั้งแต่ 1970-01-01 00:00:00 ถึง 2037

**TIME** ขนาดที่เก็บ 3 ไบต์

เก็บค่าเวลาในรูปแบบ HH:mm:SS

โดยมีค่าตั้งแต่ 00:00:00 ถึง 23:59:59

**YEAR(2 หรือ 4)** ขนาดที่เก็บ 1 ไบต์

เก็บค่าปี 2 หรือ 4 หลัก

หากระบุเป็น 2 จะเก็บค่า 70 ถึง 69 หมายถึงปี 1970 ถึง 2069

หากระบุเป็น 4 จะเก็บค่า 1901 ถึง 2155

## **ข้อมูลตัวอักษร:**

**CHAR(m)** ขนาดที่เก็บ m ไบต์ แต่ไม่เกิน 255 ไบต์  
เก็บค่าอักขรตามรหัส ASCII หรือ encoding ที่ใช้

**VARCHAR(m)** ขนาดที่เก็บตามข้อมูลจริง มากที่สุด m ไบต์ แต่ไม่เกิน 255 ไบต์  
เก็บค่าอักขรตามรหัส ASCII หรือ encoding ที่ใช้

CHAR จะเก็บขนาดตายตัว ส่วน VARCHAR จะเก็บตามขนาดของข้อมูลตัวอักษร

## **TINYBLOB หรือ TINYTEXT**

ขนาดที่เก็บตามข้อมูลจริง +1 ไบต์ แต่ไม่เกิน 255 ไบต์  
เก็บค่าอักขรตามรหัส ASCII หรือข้อมูล BINARY

## **BLOB หรือ TEXT**

ขนาดที่เก็บตามข้อมูลจริง +2 ไบต์ แต่ไม่เกิน 65535 ไบต์  
เก็บค่าอักขรตามรหัส ASCII หรือข้อมูล BINARY

## **MEDIUMBLOB หรือ MEDIUMTEXT**

ขนาดที่เก็บตามข้อมูลจริง +3 ไบต์ แต่ไม่เกิน 16777215 ไบต์  
เก็บค่าอักขรตามรหัส ASCII หรือข้อมูล BINARY

## **LOB หรือ LONGTEXT**

ขนาดที่เก็บตามข้อมูลจริง +4 ไบต์ แต่ไม่เกิน 4294967295 ไบต์  
เก็บค่าอักขรตามรหัส ASCII หรือข้อมูล BINARY

## **ENUM('value1','value2',...)**

ขนาดที่เก็บ 1 หรือ 2 ไบต์ ตามจำนวนค่า value ซึ่งกำหนดได้มากที่สุด 65535 ค่า  
เก็บค่าตาม value ที่กำหนด

## **SET('value1','value2',...)**

ขนาดที่เก็บ 1, 2, 3, 4 หรือ 8 ไบต์ ตามจำนวนค่า value ซึ่งกำหนดได้มากที่สุด 64 ค่า  
เก็บค่าตาม value ที่กำหนด

ENUM เก็บค่า value เพียงค่าเดียวต่อหนึ่งแถว

ส่วน SET สามารถเก็บ value ได้หลายๆ ค่าต่อหนึ่งแถว

## คำถาม

1. เป็นตัวอักษรทุกๆไปความยาวไม่เกิน 255 ตัว

ควรใช้ VARCHAR ถ้าหากความยาวของสายอักขระไม่คงที่  
หรือใช้ CHAR ถ้าหากความยาวของสายอักขระคงที่ตายตัว

2. เป็นทั้งตัวอักษรและตัวเลข และมีความยาวมากกว่า 1,000 ตัว

ควรใช้ TEXT, MEDIUMTEXT หรือ LONGTEXT ตามความเป็นไปได้ของความยาวของสายอักขระ

3. เป็นเลขจำนวนเต็ม สามารถคำนวณได้ แต่ไม่สามารถใส่เครื่องหมายจุลภาคได้ เช่น 100000 เป็นต้น

ควรใช้ TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT หรือ INTEGER ตามความเป็นไปได้ของจำนวนที่ใช้

4. เป็นตัวเลขจำนวนเต็ม สามารถคำนวณได้ และใส่เครื่องหมายจุลภาคเก็บเป็นข้อมูลได้ เช่น 100,000 เป็นต้น

ไม่สามารถทำได้ เนื่องจากข้อมูลตัวเลขไม่เก็บเครื่องหมายจุลภาค  
ควรเรียกข้อมูลตัวเลขออกมา แล้วนำไปจัดรูปแบบใส่เครื่องหมายจุลภาคเอง

5. เป็นเลขที่มีจุดทศนิยม สามารถคำนวณได้ และใส่เครื่องหมายจุลภาคเก็บเป็นข้อมูลได้ เช่น 1,000.50 เป็นต้น

ควรใช้ FLOAT, DOUBLE หรือ REAL หากไม่ต้องการสนใจเรื่องจำนวนตำแหน่งทศนิยม  
หรือใช้ DECIMAL หรือ NUMERIC หากต้องการระบุจำนวนหลักทั้งหน้าและหลังจุดทศนิยม  
ส่วนเรื่องเครื่องหมายจุลภาคบอกแล้วในข้อ 4

6. ส่วนอีกคำถามหนึ่งก็คือ ช่อง แอตทริบิวต์ ที่มีตัว binary, unsigned, unsigned zerofill ให้เลือก ไว้ทำอะไรหรือครับ

BINARY เป็นตัวบ่งบอกว่าข้อมูลที่อยู่ภายในเป็นแบบ binary มนุษย์ไม่สามารถอ่านได้  
UNSIGNED เป็นตัวบ่งบอกว่าข้อมูลตัวเลขจะเก็บเพียงค่าบวกและศูนย์  
UNSIGNED ZEROFILL เหมือน UNSIGNED แต่จะเติม 0 ลงไปให้เต็มในหลักที่เหลืออยู่ทางซ้าย