



انواع داده در MySQL

هدف این مقاله بررسی انواع داده ها در پایگاه داده MySQL می باشد. انواع مختلف داده را در دسته بندی های زیر پشتیبانی می کند:

- اعداد (Numeric)
- زمان (Date and Time)
- رشته ها (String)
- داده های فضایی (Spatial)

در این مقاله داده های عددی، زمان و رشته ها را بررسی می کنیم. با توجه به تخصصی بودن داده های فضایی، این نوع داده ها را در مقاله ای جداگانه در آینده بررسی خواهیم کرد.

اعداد (Numeric)

MySQL تمامی انواع داده های عددی استاندارد SQL شامل اعداد صحیح و اعداد تقریبی را پشتیبانی می کند. هر یک از نوع داده های عددی حافظه مشخصی برای ذخیره سازی نیاز دارند. همچنین در نظر داشته باشید که هر کدام از آنها محدوده مشخصی را پشتیبانی می کنند. هر نوع از این داده های عددی را می توان به صورت SIGNED یا UNSIGNED تعریف کرد. در حالت SIGNED این ستون ها از مقادیر منفی پشتیبانی می کنند ولی در حالت UNSIGNED اعداد ذخیره شده باید بزرگتر و یا مساوی صفر باشند. در نظر داشته باشید که نوع داده BOOLEAN برای ذخیره مقادیر TRUE و FALSE همان TINYINT(1) می باشد. جدول زیر اطلاعات کلی نوع داده های عددی را در حالت SIGNED و UNSIGNED نشان می دهد:

Integer اطلاعات کلی نوع داده های عددی			
نوع	حافظه – بایت	حداقل مقدار	حداکثر مقدار
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INTEGER	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

اطلاعات کلی نوع داده های عددی شناور				
نوع	حافظه – بایت	مقدار	نحوه تعریف	
FLOAT	4	از ۰ تا ۲۳ رقم	FLOAT(M,D)	
DOUBLE	8	از ۲۴ تا ۵۳ رقم	DOUBLE(M,D)	
DECIMAL	هر ۹ رقم ۴ بایت به صورت باینری	تا ۶۵ رقم	DECIMAL(M,D)	

در جدول بالا حرف M نشان دهنده تعداد کل رقم های داده و حرف D نشان دهنده تعداد رقم های بعد از ممیز (.) می باشد. به عنوان مثال در ستونی که به صورت FLOAT(7,4) تعریف شده است می توان حداکثر مقدار ۹۹۹,۹۹۹۹ را ذخیره نمود. در صورتی که شما عدد ۹۹۹,۰۰۰۰۹ را به این جدول وارد کنید، عدد وارد شده به صورت ۹۹۹,۰۰۰۱ ذخیره خواهد شد.

برای اطلاعات بیشتر به صفحه [Problems with Floating-Point Values](#) در سایت MySQL مراجعه نمایید.

در نسخه های قبل از ۵,۰,۳ نوع داده DECIMAL به صورت String ذخیره می شود و مقدار رقم هایی که پشتیبانی می کرد مانند نوع داده DOUBLE بود.

نوع داده BIT

این نوع داده برای ذخیره بیت های ۰ و ۱ استفاده می شود. برای تعریف ستونی از این نوع از عبارت BIT(M) استفاده می کنیم که در اینجا M فضایی به طول تقریباً $(M+7)/8$ بایت را برای نگهداری داده ها ایجاد می کند M. می تواند عددی از ۱ تا ۶۴ باشد. برای نشان دادن ۰ و ۱ به صورت بیت از حرف b استفاده

می کنیم. به عنوان مثال 'b'111' و 'b'10000000' که به ترتیب برابر اعداد ۷ و ۱۲۸ می باشند. برای درک بیشتر این موضوع به دستورات و نتایج زیر در خط فرمان MySQL توجه فرمایید:

```
mysql> INSERT INTO t SET b = b'11111111';
mysql> INSERT INTO t SET b = b'1010';
mysql> INSERT INTO t SET b = b'0101';
mysql> SELECT b+0, BIN(b+0), OCT(b+0), HEX(b+0) FROM t;
+-----+-----+-----+-----+
| b+0 | BIN(b+0) | OCT(b+0) | HEX(b+0) |
+-----+-----+-----+-----+
| 255 | 11111111 | 377       | FF       |
| 10  | 1010     | 12        | A        |
| 5   | 101      | 5         | 5        |
+-----+-----+-----+-----+
3 rows in set (0.03 sec)
```

آخرین Query در بالا توابع مختلف برای تبدیل این اعداد را نشان می دهد. عبارت 0+ مقدار دودویی ذخیره شده را به عدد دهدهی متناظر تبدیل می کند.

زمان (Date and Time)

در کل ۵ نوع داده برای ذخیره زمان (ساعت و تاریخ) داریم. در جدول زیر اطلاعات این نوع داده ها را مشاهده می فرمایید:

اطلاعات کلی نوع داده های زمانی (Date and Time)		
نوع	حافظه – بایت	مقدار صفر ("ZERO" VALUE)
DATE	3	'0000-00-00'
TIME	3	'00:00:00'
DATETIME	8	'0000-00-00 00:00:00'
TIMESTAMP	4	'0000-00-00 00:00:00'
YEAR	1	0000

در ادامه به بررسی نوع داده های ذکر شده در بالا خواهیم پرداخت.

نوع داده DATE

این نوع داده فقط برای ذخیره سازی تاریخ استفاده می شود. فرمت پیش فرض این نوع داده 'YYYY-MM-DD' است. بازه ای که توسط DATE پشتیبانی می شود از '1000-01-01' تا '9999-12-31' می باشد.

نوع داده TIME

این نوع داده برای ذخیره سازی زمان (ساعت، دقیقه، ثانیه) استفاده می شود. فرمت پیش فرض این نوع داده 'HH:MM:SS' و یا 'HHH:MM:SS' است. بازه ای که در نوع TIME پشتیبانی می شود از '838:59:59' تا '838:59:59' می باشد. نوع داده TIME فقط برای نمایش زمان روز استفاده نمی شود. برای نمایش تفاوت زمانی دو بازه و یا زمان باقی مانده نیز می تواند از TIME استفاده نمود. به همین دلیل بازه پشتیبانی شده بیشتر از ۲۴ ساعت و یا کمتر از صفر (منفی) می باشد.

برای استفاده از عبارت های مخفف در نوع داده TIME باید دقت نمایید. به عنوان مثال MySQL عبارت '11:12' را به صورت '11:12:00' تفسیر می کند. در صورتی که داده بدون (':') دو نقطه) بیان شود، MySQL آن را از سمت راست تفسیر می کند. به عنوان مثال عبارت '1112' و یا '1112' به معنی '00:11:12' است.

به صورت پیش فرض MySQL داده های خارج از محدوده TIME را در صورتی که با فرمت مناسب وارد شده باشند به نزدیک ترین بازه تبدیل می کند. به عنوان مثال عبارت های '850:00:00' و '850:00:00' به '838:59:59' و '838:59:59' تبدیل خواهند شد. داده هایی که فرمت مناسب نداشته و یا قابل تفسیر برای داده TIME نباشند به عبارت '00:00:00' تبدیل خواهند شد.

نوع داده DATETIME

این نوع داده برای ذخیره سازی زمان به صورت کامل شامل تاریخ و ساعت استفاده می شود. فرمت پیش فرض این نوع داده 'YYYY-MM-DD HH:MM:SS' می باشد. بازه ی پشتیبانی DATETIME از '1000-01-01' تا '9999-12-31 23:59:59' است.

نوع داده TIMESTAMP

این نوع داده همانند DATETIME می باشد. بازه ای که توسط TIMESTAMP پشتیبانی می شود از '1970-01-01 00:00:01' UTC تا '2038-01-19 03:14:07' UTC است. داده های TIMESTAMP هنگام ذخیره شدن از زمان منطقه زمانی (Time Zone) کنونی سیستم به UTC تبدیل و ذخیره می شوند. متناظر با همین امر، هنگام نشان دادن داده های TIMESTAMP، ابتدا داده ها از UTC به منطقه زمانی کنونی سیستم تبدیل شده و بعد نمایش داده می شوند. به همین علت در صورتی که منطقی زمانی سیستم عوض شود، ممکن است داده ها به صورت غیرعادی و یا غلط نمایش داده شوند. نوع داده TIMESTAMP این قابلیت را دارد که مقداردهی اولیه و یا مقداردهی خودکار داشته باشید. برای

مقداردهی اولیه کفایست مقدار پیش فرض فیلد مورد نظر را `CURRENT_TIMESTAMP` قرار دهید.
مانند:

1 `mysql> CREATE TABLE t (ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP);`
در حالت مقداردهی خودکار، مقدار ذخیره شده در فیلد `TIMESTAMP` هنگام بروزرسانی (`UPDATE`) یک ردیف به زمان بروزرسانی تغییر پیدا می کند.

برای کسب اطلاعات بیشتر به صفحه [TIMESTAMP Properties](#) در سایت MySQL مراجعه نمایید.

نوع داده YEAR

این نوع داده برای ذخیره سازی سال با فرمت `YYYY` به کار می رود. شما می توانید سال مورد نظر خود را به صورت دو رقمی (`YEAR(2)`) و یا ۴ رقمی (`YEAR(4)`) تعریف کنید. نوع `YEAR` از سال ۱۹۰۱ تا ۲۱۵۵ و ۰۰۰۰ را پشتیبانی می کند. در صورتی که سال را به صورت ۲ رقمی تعریف کرده باشید، اعداد ۰۰ تا ۶۹ در قرن ۲۱ و اعداد ۷۰ تا ۹۹ در قرن ۲۰ در نظر گرفته می شوند. مثلاً ۴۵ به عنوان ۲۰۴۵ و ۸۰ به صورت ۱۹۸۰ در نظر گرفته می شود. تمامی مقادیر غیر منطقی برای `YEAR` به صورت ۰۰۰۰ ذخیره می شوند.

رشته ها (String)

نوع داده `String` برای نگه داری رشته ها (متون) استفاده می شود. در زیر اطلاعات کلی داده هایی از نوع `String` را مشاهده می فرمایید. در جدول زیر `M` نشان دهنده طول تعریف شده برای داده و `L` نشان دهنده طول واقعی رشته داده شده می باشد.

اطلاعات کلی نوع داده رشته ها (String)		
نوع	مقدار	حافظه – بایت
CHAR(M)	$0 \leq M \leq 255$	$M \times w$ بایت w . تعداد بایت مورد نیاز برای حداکثر طول کاراکترهای تعریف شده در <code>character set</code> جدول می باشد.
BINARY(M)	$0 \leq M \leq 255$	M بایت
VARCHAR(M)	$0 \leq M \leq 65532$	$L + 1$ بایت اگر داده مورد نظر کمتر از ۲۵۵ بایت نیاز داشته باشد. $L + 2$ بایت اگر داده مورد نظر بیشتر از ۲۵۵ بایت نیاز داشته باشد.
VARBINARY(M)	$0 \leq M \leq 65532$	$M + 1$ بایت اگر داده مورد نظر کمتر از ۲۵۵ بایت نیاز داشته باشد. $M + 2$ بایت اگر داده مورد نظر بیشتر از ۲۵۵ بایت نیاز داشته باشد.
TINYBLOB	$L < 2^8$	$L + 1$ بایت
TINYTEXT	$L < 2^8$	$L + 1$ بایت

BLOB	$L < 2^{16}$	$L + 2$ بایت
TEXT	$L < 2^{16}$	$L + 2$ بایت
MEDIUMBLOB	$L < 2^{24}$	$L + 3$ بایت
MEDIUMTEXT	$L < 2^{24}$	$L + 3$ بایت
LOB	$L < 2^{32}$	$L + 4$ بایت
LONGTEXT	$L < 2^{32}$	$L + 4$ بایت
ENUM('v1','v2',...)	حداکثر ۶۵،۵۳۵ عضو	۱ یا ۲ بایت. وابسته به مقدار داده.
SET('v1','v2',...)	حداکثر ۶۴ عضو	۱، ۲، ۳، ۴ یا ۸ بایت وابسته به مجموعه داده شده.

نوع داده CHAR و VARCHAR

CHAR و VARCHAR بسیار شبیه به هم هستند. مهم‌ترین تفاوتی که بین CHAR و VARCHAR وجود دارد نحوه ذخیره سازی و بازیابی اطلاعات است. هنگام تعریف این نوع داده‌ها شما حداکثر تعداد کاراکتری را که می‌خواهید ذخیره شود را مشخص می‌نمایید. به عنوان مثال CHAR(30) برای ذخیره حداکثر ۳۰ کاراکتر تعریف می‌شود.

طول ستونی که از نوع CHAR باشد ثابت است. در صورتی که طول داده کمتر از طول تعریف شده باشد، فضای باقیمانده با فاصله (space) از راست پر می‌شود. هنگام بازیابی اطلاعات این فضای خالی قبل از نمایش حذف می‌شود مگر اینکه حالت **PAD_CHAR_TO_FULL_LENGTH** فعال شده باشد. در نوع داده VARCHAR طول داده ذخیره شده به اندازه مقدار داده وارد شده می‌باشد. به عنوان مثال اگر ستونی به صورت VARCHAR(100) تعریف شده باشد ولی طول رشته داده شده ۵۰ کاراکتر باشد، فقط به مقدار بایت زیر فضا لازم است.

(۵۰ * فضای لازم برای ذخیره هر کاراکتر) + ۲

در صورتی که در تعریف VARCHAR(M) مقدار M بیشتر از ۶۵،۵۳۵ باشد، MySQL به طور خودکار نوع ستون را از VARCHAR به نوع مناسبی مانند MEDIUMTEXT تغییر می‌دهد.

جدول زیر تفاوت‌های فضای مورد نیاز برای ذخیره اطلاعات را در CHAR و VARCHAR نمایش می‌دهد. دقت کنید که در جدول زیر Character Set داده مورد نظر تک بیتی مانند latin1 است.

بررسی تفاوت فضای مورد نیاز برای CHAR(4) و VARCHAR(4)				
فضای مورد نیاز	VARCHAR(4)	فضای مورد نیاز	CHAR(4)	داده
۱ بایت	"	۴ بایت	' '	"
۳ بایت	'ab'	۴ بایت	'ab '	'ab'
۵ بایت	'abcd'	۴ بایت	'abcd'	'abcd'
۵ بایت	'abcd'	۴ بایت	'abcd'	'abcdefgh'

نوع داده BINARY و VARBINARY

این نوع داده ها مانند CHAR و VARCHAR می باشند فقط با این تفاوت که در این نوع داده ها رشته های باینری (صفر و یک) ذخیره می شود. در نتیجه Character set برای این نوع داده ها معنی ندارد. از طرفی مرتب سازی و مقایسه بر اساس مقدار عددی رشته باینری ذخیره شده می باشد.

نوع داده TEXT و BLOB

این نوع داده ها مانند BINARY و VARBINARY می باشند. هر کدام از این داده ها دارای ۴ نوع می باشند که تنها تفاوت آن ها در حداکثر طول داده قابل ذخیره شدن است.

نوع داده ENUM

این نوع داده فقط امکان ذخیره سازی رشته ای را می دهد که از مجموعه تعریف شده باشد. این مجموعه هنگام ساخت جدول تعریف می شود. مانند جدول زیر:

```
CREATE TABLE sizes (
    name ENUM('small', 'medium', 'large')
);
```

در ستون name در جدول بالا فقط می توانید یکی از مقادیر small، medium یا large را ذخیره نمایید. در صورتی که مقداری خارج از مجموعه وارد مجموعه شود، فضای خالی در name ذخیره می شود، نه مقدار وارد شده.

نوع داده SET

این نوع داده مانند ENUM می باشد، با این تفاوت که شما می توانید چند عضو از مجموعه تعریف شده را انتخاب نمایید. توجه نمایید که یا داده های تکراری قبل از ذخیره سازی حذف می شوند. به مثال زیر توجه فرمایید:

```
mysql> CREATE TABLE myset (col SET('a', 'b', 'c', 'd'));
Query OK, 0 rows affected (0.05 sec)

mysql> INSERT INTO myset (col) VALUES ('a,d'), ('d,a'), ('a,d,a'), ('a,d,d'), ('d,a,d');
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> SELECT col FROM myset;
+-----+
| col   |
+-----+
| a,d   |
| a,d   |
| a,d   |
| a,d   |
| a,d   |
+-----+
5 rows in set (0.00 sec)
```

از قراردادن فاصله قبل و یا بعد ',' جدا خودداری نمایید.

چطور نوع داده مناسب را انتخاب نماییم؟

قبل از هر چیز بازه‌ی مورد نیاز برای داده خود را معین نمایید. سپس مختصرترین نوع داده را برای ستون خود انتخاب نمایید. به عنوان مثال برای بازه اعداد ۰ تا ۹۹۹۹۹ نوع MEDIUMINT UNSIGNED بهترین گزینه است.

مورد دیگر انتخاب نوع ستون مناسب با داده شما است. به عنوان مثال هرگز تاریخ شمسی را در پایگاه داده در ستونی از نوع رشته (مانند CHAR) ذخیره ننمایید. بهترین گزینه تبدیل تاریخ به تاریخ میلادی و ذخیره آن در داده‌ای از نوع DATE است.

در هنگام تعریف ستونی برای داده‌های عددی در صورتی که فقط اعداد مثبت را لازم دارید، ستون خود را از نوع UNSIGNED انتخاب نموده و حداقل بازه را انتخاب نمایید. به عنوان مثال اگر ستون عددی شما کلید اصلی و افزایشی خودکار است (PRIMARY KEY and AUTO_INCREMENT) حتما نوع داده خود را UNSIGNED انتخاب نمایید تا حداکثر مقدار کلید اصلی دو برابر شود. نوع INTEGER SIGNED حداکثر عدد ۲۱۴۷۴۸۳۶۴۸ (حدود ۲ میلیارد) و INTEGER UNSIGNED حداکثر عدد ۴۲۹۴۹۶۷۲۹۵ (حدود ۴ میلیارد) را پشتیبانی می‌کند.

روش دیگر استفاده از PROCEDURE ANALYSE به منظور دریافت پیشنهاد خود پایگاه داده است PROCEDURE ANALYSE. با توجه به داده‌های فعلی، پیشنهاد خود را برای بهینه کردن نوع داده ارائه می‌دهد. در نظر داشته باشید که هر چه تعداد داده فعلی شما بیشتر باشد، PROCEDURE

ANALYSE پیشنهاد بهتر و دقیق‌تری می‌دهد. نحوه استفاده از PROCEDURE ANALYSE بسیار ساده است. نحوه نگارش آن به صورت زیر است:

```
mysql> SELECT ... FROM ... WHERE ... PROCEDURE ANALYSE([max_elements,[max_memory]]);

SELECT col1, col2 FROM table1 PROCEDURE ANALYSE(10, 2000);

SELECT * FROM table1 PROCEDURE ANALYSE() \G
```

اسامی مشابه

در پایگاه داده MySQL برخی از انواع داده‌ها با نام‌های مشابه دیگری نیز قابل دسترسی می‌باشند. یکی از دلایل این کار پیاده‌سازی راحت‌تر SQL های پایگاه‌های داده‌ی دیگر است. در زیر این اسامی را مشاهده می‌نمایید.

اسامی مشابه برای نوع داده‌ها در MySQL	
نوع داده	اسم مشابه
INTEGER	INT
DECIMAL	DEC
DECIMAL	FIXED
DECIMAL	NUMERIC
DOUBLE	REAL
TINYINT	BOOLEAN
TINYINT	BOOL
VARCHAR(M)	CHARACTER VARYING(M)
FLOAT	FLOAT4
DOUBLE	FLOAT8
TINYINT	INT1
SMALLINT	INT2
MEDIUMINT	INT3
INT	INT4
BIGINT	INT8
MEDIUMBLOB	LONG VARBINARY
MEDIUMTEXT	LONG VARCHAR
MEDIUMTEXT	LONG
MEDIUMINT	MIDDLEINT

www.7Learn.com

مطالب این مقاله از وبلاگ سعید زبردست گرفته شده است.