

My SQL Column Types

Brief Reference

Taken from: <http://www.flatmysql.com/help/columnntypes.htm>

MySQL stores data in your table in columns. Each column usually contains only a specific class of information. A database that has a table that contains contact information might have the following columns: Name, Address, City, State, Zip, Phone, FAX, Client_Number

Each column has a column type and most column types require that you specify a size. For instance, **varchar(60)** would probably be a good column type and size for the Name column, since it is hard to imagine someone having a name longer than 60 characters. **varchar** is a somewhat generic column type that can store most of the characters that appear on your keyboard.

Here are some common column types and some notes about using them:

Column Name	Maximum Size	Notes
varchar	255 characters	Can store most of the characters that appear on your keyboard. When in doubt, use this column type.
char	255 characters	Should be used only to store characters that tend to have the same width. For instance, varchar(2) would be a good choice for a field that will only contain US state abbreviations, since no state abbreviation is longer than 2 characters. You could also use varchar in this example (and MySQL may convert it to varchar anyway), but selecting char won't necessarily hurt. For columns containing longer and variable length content, don't use char , or MySQL will make your database consume more disk space than is necessary.
int	a whole number less than 2,140,000,000 (2.14 billion)	Should be used to store whole numbers only. A whole number is a number like 8876543 that contains no decimal parts or spaces. Phone numbers like (740) 975-8891 and prices like \$4.24 can't be stored in int columns because they are not whole numbers.
text	65,000 characters	Can store up to 65KB of text consisting of most keyboard characters. Note: your browser may not be able to send 65KB of text via an HTML form.
mediumtext	mediumtext up to 16,000,000	Can store up to 16MB of keyboard character type text. Note: there is no browser that will package and send 16MB via an HTML form, so mediumtext might be overkill for the great

	characters	majority of applications.
float	floating point numbers	Consists of two parts, a whole number and a decimal number. It takes the form float(w,d) where w is the max number of digits that the whole number can contain and d is the max number of digits that the decimal can contain. For example, the largest number that float(4,2) can hold is 9999.99

Detailed Reference

Taken from: http://miketeo.net/ref/mysql/column_type.html

Numeric Column Types

```
? TINYINT[(M)] [UNSIGNED] [ZEROFILL]
? SMALLINT[(M)] [UNSIGNED] [ZEROFILL]
? MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]
? INT[(M)] [UNSIGNED] [ZEROFILL]
? INTEGER[(M)] [UNSIGNED] [ZEROFILL]
? BIGINT[(M)] [UNSIGNED] [ZEROFILL]
? FLOAT(precision) [ZEROFILL]
? FLOAT[(M,D)] [ZEROFILL]
? DOUBLE[(M,D)] [ZEROFILL]
? DOUBLE PRECISION[(M,D)] [ZEROFILL]
? REAL[(M,D)] [ZEROFILL]
? DECIMAL[(M[,D])] [ZEROFILL]
? NUMERIC(M,D) [ZEROFILL]
```

Note that if you specify ZEROFILL for a column, MySQL will automatically add the UNSIGNED attribute to the column.

Type	UNSIGNED	ZEROFILL	Storage	
TINYINT	Y	Y	1 byte	A very small integer. The signed range is -128 to 127. The unsigned range is 0 to 255.
SMALLINT	Y	Y	2 bytes	A small integer. The signed range is -32768 to 32767. The unsigned range is 0 to 65535.
MEDIUMINT	Y	Y	3 bytes	A medium-size integer. The signed range is -8388608 to 8388607. The unsigned range is 0 to 16777215.

INT	Y	Y	4 bytes	A normal-size integer. The signed range is -2147483648 to 2147483647. The unsigned range is 0 to 4294967295.
INTEGER	Y	Y	4 bytes	Synonym for INT.
BIGINT	Y	Y	8 bytes	A large integer. The signed range is -9223372036854775808 to 9223372036854775807. The unsigned range is 0 to 18446744073709551615. Note that all arithmetic is done using signed BIGINT or DOUBLE values, so you shouldn't use unsigned big integers larger than 9223372036854775807 (63 bits) except with bit functions! Note that -, + and * will use BIGINT arithmetic when both arguments are INTEGER values! This means that if you multiply two big integers (or results from functions that return integers) you may get unexpected results if the result is larger than 9223372036854775807.
FLOAT(precision)	N	Y	4 if X<=24, or 8 if 25<=X<=53	A floating-point number. Cannot be unsigned. precision can be <=24 for a single precision floating point number and between 25 and 53 for a double precision floating point number. These types are like the FLOAT and DOUBLE types described immediately below. FLOAT(X) have the same ranges as the corresponding FLOAT and DOUBLE types, but the

				display size and number of decimals is undefined. In MySQL 3.23, this is a true floating point value. In earlier MySQL versions, FLOAT(precision) always has 2 decimals. This syntax is provided for ODBC compatibility.
FLOAT	N	Y	4 bytes	A small (single-precision) floating-point number. Allowable values are -3.402823466E+38 to -1.175494351E-38, 0 and 1.175494351E-38 to 3.402823466E+38. FLOAT without an argument or with an argument of <= 24 stands for a single-precision floating point number.
DOUBLE	N	Y	8 bytes	A normal-size (double-precision) floating-point number. Allowable values are -1.7976931348623157E+308 to -2.2250738585072014E-308, 0 and 2.2250738585072014E-308 to 1.7976931348623157E+308. DOUBLE without an argument or FLOAT(X) where 25 <= X <= 53 stands for a double-precision floating point number.
DOUBLE PRECISION	N	Y	8 bytes	Synonym for DOUBLE.
REAL	N	Y	8 bytes	Synonym for DOUBLE.
DECIMAL	N	Y	M bytes, or D+2 if M<D	An unpacked floating-point number. Behaves like a CHAR column: ``unpacked'' means the number is stored as a string, using one character for each digit of

				the value. The decimal point, and, for negative numbers, the '-' sign is not counted in M. If D is 0, values will have no decimal point or fractional part. The maximum range of DECIMAL values is the same as for DOUBLE, but the actual range for a given DECIMAL column may be constrained by the choice of M and D. If D is left out it's set to 0. If M is left out it's set to 10. Note that in MySQL 3.22 the M argument includes the sign and the decimal point.
NUMERIC	N	Y	M bytes, or D+2 if M<D	Synonym for DECIMAL.

Date & Time Column Types

- ? DATE
- ? DATETIME
- ? TIMESTAMP[(M)]
- ? TIME
- ? YEAR[(2 | 4)]

Type	Storage	
DATE	3 bytes	A date. The supported range is '1000-01-01' to '9999-12-31'.MySQL displays DATE values in 'YYYY-MM-DD' format, but allows you to assign values to DATE columns using either strings or numbers.
DATETIME	8 bytes	A date and time combination. The supported range is '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. MySQL displays DATETIME values in 'YYYY-MM-DD HH:MM:SS' format, but allows you to assign values to DATETIME columns using either strings or numbers.
TIMESTAMP	4 bytes	A timestamp. The range is '1970-01-01 00:00:00' to sometime in the year 2037. MySQL displays TIMESTAMP values in YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD or YYMMDD format, depending on whether M

		is 14 (or missing), 12, 8 or 6, but allows you to assign values to TIMESTAMP columns using either strings or numbers. A TIMESTAMP column is useful for recording the date and time of an INSERT or UPDATE operation because it is automatically set to the date and time of the most recent operation if you don't give it a value yourself. You can also set it to the current date and time by assigning it a NULL value.
TIME	3 bytes	A time. The range is '-838:59:59' to '838:59:59'. MySQL displays TIME values in 'HH:MM:SS' format, but allows you to assign values to TIME columns using either strings or numbers.
YEAR	1 byte	A year in 2- or 4- digit formats (default is 4-digit). The allowable values are 1901 to 2155, and 0000 in the 4 year format and 1970-2069 if you use the 2 digit format (70-69). MySQL displays YEAR values in YYYY format, but allows you to assign values to YEAR columns using either strings or numbers. (The YEAR type is new in MySQL 3.22.)

String Column Types

- ? [NATIONAL] CHAR(M) [BINARY]
- ? [NATIONAL] VARCHAR(M) [BINARY]
- ? TINYBLOB
- ? TINYTEXT
- ? BLOB
- ? TEXT
- ? MEDIUMBLOB
- ? MEDIUMTEXT
- ? LONGBLOB
- ? LONGTEXT
- ? ENUM('value1', 'value2', ...)
- ? SET('value1', 'value2', ...)

Type	Storage	
CHAR	M bytes	A fixed-length string that is always right-padded with spaces to the specified length when stored. The range of M is 1 to 255 characters. Trailing spaces are removed when the value is retrieved. CHAR values are sorted and compared in case-insensitive fashion according to the default character set unless the BINARY keyword is given. NATIONAL CHAR (short form NCHAR) is the ANSI SQL way to define that a CHAR column should use the default CHARACTER set. This is default in MySQL. CHAR is a shorthand for CHARACTER . MySQL allows you to create a column of type CHAR(0) . This is mainly useful when you have to be compliant with some old applications that depend on the

		existence of a column but that do not actually use the value. This is also quite nice when you need a column that only can take 2 values: A CHAR(0), that is not defined as NOT NULL, will only occupy one bit and can only take 2 values: NULL or "".
VARCHAR	L+1 bytes	A variable-length string. Note: Trailing spaces are removed when the value is stored (this differs from the ANSI SQL specification). The range of M is 1 to 255 characters. VARCHAR values are sorted and compared in case-insensitive fashion unless the BINARY keyword is given.
TINYTEXT	L+1 bytes	A BLOB or TEXT column with a maximum length of 255 (2 ⁸ - 1) characters.
TINYBLOB	L+1 bytes	Synonym for TINYTEXT.
BLOB	L+2 bytes	A BLOB or TEXT column with a maximum length of 65535 (2 ¹⁶ - 1) characters.
TEXT	L+2 bytes	Synonym for BLOB.
MEDIUMBLOB	L+3 bytes	A BLOB or TEXT column with a maximum length of 16777215 (2 ²⁴ - 1) characters.
MEDIUMTEXT	L+3 bytes	Synonym for MEDIUMBLOB.
LOB	L+4 bytes	A BLOB or TEXT column with a maximum length of 4294967295 (2 ³² - 1) characters.
LONGTEXT	L+4 bytes	Synonym for LOBBLOB.
ENUM	1 or 2 bytes	An enumeration. A string object that can have only one value, chosen from the list of values 'value1', 'value2', ..., or NULL. An ENUM can have a maximum of 65535 distinct values.
SET	1, 2, 4, or 8 bytes	A set. A string object that can have zero or more values, each of which must be chosen from the list of values 'value1', 'value2', ... A SET can have a maximum of 64 members.