

# SQL and MySQL Data Types

[SQL data types and MySQL](#)

[Column type modifiers in MySQL](#) [Character and binary types in MySQL](#)

[Numeric types in MySQL](#) [DateTime types in MySQL](#) [OpenGIS types in MySQL](#)

[NULL](#) [Column types and indexes](#) [Column types and referential integrity](#)

Databases have tables. Tables have columns. Columns have types and other properties. From those types and properties flow crucial characteristics of your database.

## SQL data types and MySQL

Look in any computer language manual and you will find a list of data types like this:

- *Character stream, or string*: one or more alphanumeric characters likely to be meaningful or printable in text, e.g., a company name or.
- *Binary stream*: a sequence of characters that may be printable or may encode other information for example a digital photograph, an engineering drawing, or a document formatted by a word processor.
- *Number*: a negative or positive numeric value, small or large, rational or irrational, for example 27, -1, 3.14159, 6.28 + 3i. A numeric subtype of special interest is the two-valued numeric called boolean (True/False, Yes/No, Living/Dead, or whatever is appropriate to the problem domain).
- *Datetime*: dates, times, timestamps.

Databases are concerned at the most fundamental level with data types, because every column of every table must be of a defined base type, or of a user-defined type (UDT) derived from a base type. As of version 5, UDTs are not yet available in MySQL.

SQL92 defines a set of base types. Every SQL92 implementation delivers a slightly different subset of these types. In designing your tables, choose types that best fit your requirements in the interests of efficiency and performance. MySQL offers these:

- *Character string*, or string for short: a sequence of characters, as short as someone's initial 'A', or as long as a huge sequence of four billion characters; MySQL has eight string column types including CHAR, VARCHAR, four TEXT types, ENUM and SET.
- *Binary stream* or object: a character stream without optimisation for rendering as text. MySQL has two BINARY and four Binary Large Object (BLOB) types.
- *Numeric*: representation of a number as an integer, a floating point value, or a value to a fixed decimal precision; MySQL has 11 numeric data types: BIT, TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, three FLOATs, DOUBLE, DECIMAL.

- *Datetime*: a value representing the time in milliseconds since a reference date. MySQL has five datetime types: DATE, TIME, DATETIME, TIMESTAMP, and YEAR.
- *OpenGIS*: types for representing spherical geometric values.

## Column type modifiers in MySQL

When you [CREATE](#) or ALTER a column, you may also specify properties the MySQL manual calls *field attributes*, for example UNSIGNED for only positive values, or NULL to accept null values. But *attribute* is a formal SQL synonym for *column*, so using it also for column properties like NULL makes them attributes of attributes. Confusing at best. We refer to these properties as *type modifiers*, because that's exactly what they do.

The general column type modifiers are

- NOT NULL: the column does not accept NULL values, for example `lastname CHAR(20) NOT NULL`,
- NULL: the column accepts NULL values, for example `middlename CHAR(2) NULL`,
- DEFAULT *x*: the default value of the column is *x*, for example `country_code CHAR(2) DEFAULT 'US'`.
- UNSIGNED: numeric type accepts values  $\geq 0$  only.
- AUTO\_INCREMENT: in numeric columns only, automatically assign the next available value to a column in a new row
- *Maximum display width*: specify this by adding a number in parentheses right after the type name, for example `qty INT(4)`. See [Chapter 6](#) (CREATE TABLE) for display width modifier syntax.

To read the rest of this and other chapters, [buy a copy of the book](#)

---

[TOC](#) [Previous](#) [Next](#)

---