**MySQL Data Types Quick Reference Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Type** | **Name** | **Range** | **Attributes** | **Default** |
| Numeric | TINYINT[(M)]  M - *display width*. The weird thing is, though that, for example, if you have a value of 5 digits in a field with a display width of 4 digits, the display width will not cut a digit off.  If the value has less digits than the display width, nothing happens either. So it seems like the display doesn't have any effect in real life. | -128 TO 127 [0 to 255 if UNSIGNED] | AUTO\_INCREMENT, UNSIGNED, ZEROFILL  ZEROFILL is a neat feature that pads values that are less than the specified *display width* with zeros, so that you will always receive a value of the specified length. This is for example useful for invoice ids. | NULL [0 if NOT NULL] |
| Numeric | SMALLINT[(M)] | -32,768 to 32,767 [0 to 65,535] | AUTO\_INCREMENT, UNSIGNED, ZEROFILL | NULL [0 if NOT NULL] |
| Numeric | MEDIUMINT[(M)] | -8,388,608 to 8,388,607 [0 to 16,777,215] | AUTO\_INCREMENT, UNSIGNED, ZEROFILL | NULL [0 if NOT NULL] |
| Numeric | INT[(M)] | -2,147,483,648 to 2,147,483,647 [0 to 4,294,967,295] | AUTO\_INCREMENT, UNSIGNED, ZEROFILL | NULL [0 if NOT NULL] |
| Numeric | BIGINT[(M)] | -/+9,223,372,036,854,775,807 [0 to 18,446,744,073,709,551,615] | AUTO\_INCREMENT, UNSIGNED, ZEROFILL | NULL [0 if NOT NULL] |
| Numeric | FLOAT[(M,D)]  (*M*,*D*)means than values can be stored with up to *M* digits in total, of which *D* digits may be after the decimal point. For example, a column defined as FLOAT(7,4) will look like -999.9999 when displayed | Min=+/-1.175E-38 Max=+/-3.403E+38 | ZEROFILL | NULL [0 if NOT NULL] |
| Numeric | DOUBLE[(M,D)] | Min=+/-2.225E-308 Max=+/-1.800E+308 | ZEROFILL | NULL [0 if NOT NULL] |
| Numeric | DECIMAL(M,D) Stored as string  Used for money as it is more exact that float or double. Float are not exact and can introduce cumulative rounding errors | Max Range = DOUBLE range | ZEROFILL | NULL [0 if NOT NULL] |
| String | CHAR(M) | M=0-255 Characters Right Padded | BINARY | NULL ["" if NOT NULL] |
| String | VARCHAR(M) | M=0-65,535 Characters Trailing spaces removed | BINARY | NULL ["" if NOT NULL] |
| String | TINYBLOB | 0-255 bytes | (case sensitive) | NULL ["" if NOT NULL] |
| String | BLOB (images or other files)  No length specified | 0-65,535 bytes | (case sensitive) | NULL ["" if NOT NULL] |
| String | MEDIUMBLOB | 0-16,777,215 bytes | (case sensitive) | NULL ["" if NOT NULL] |
| String | LONGBLOB | 0-4,294,967,295 bytes | (case sensitive) | NULL ["" if NOT NULL] |
| String | TINYTEXT | 0-255 bytes | (not case sensitive) | NULL ["" if NOT NULL] |
| String | TEXT  No length specified | 0-65,535 bytes | (not case sensitive) | NULL ["" if NOT NULL] |
| String | MEDIUMTEXT | 0-16,777,215 bytes | (not case sensitive) | NULL ["" if NOT NULL] |
| String | LONGTEXT | 0-4,294,967,295 | (not case sensitive) | NULL ["" if NOT NULL] |
| String | ENUM("value1","value2",...) | Column is exactly 1 of values 1-255 values | (not case sensitive) | NULL ["" if NOT NULL] |
| String | SET("value1","value2",...) | Column is 0 or more values in list 1-64 members | (not case sensitive) | NULL ["" if NOT NULL] |
| Date & Time | DATE | "1000-01-01" - "9999-12-31" | (YYYY-MM-DD) | NULL ["0000-00-00" if NOT NULL] |
| Date & Time | TIME | "-838:59:59" - "838:59:59" | (hh:mm:ss) | NULL ["00:00:00" if NOT NULL] |
| Date & Time | DATETIME | "1000-01-01 00:00:00" - "9999-12-31 23:59:59" | (YYYY-MM-DD hh:mm:ss) | NULL ["0000-00-00 00:00:00" if NOT NULL] |
| Date & Time | TIMESTAMP | 19700101000000 - 2037+ | (YYYYMMDDhhmmss) | Current Date & Time |
| Date & Time | YEAR | 1900 - 2155 | (YYYY) | NULL ["0000" if NOT NULL] |
| String | ENUM  Fancy name for a list. Create a list of items from which the value must be selected or it can be null. | fieldName ENUM('Yes', No')  Values must be enclosed within ‘’ even numbers otherwise they are treated as an index (numbers not recommended).  If you insert an invalid value the ‘’ empty string is put in. | Stores up to 65,535 enumerated types. The DEFAULT modifier may be used to specify the default value for this field. |  |

Code to explain **zerofill** and **M** in int datatype.

mysql> create table a ( a tinyint );  
Query OK, 0 rows affected (0.29 sec)  
mysql> show columns from a;  
+-------+------------+------+-----+---------+-------+  
| Field | Type | Null | Key | Default | Extra |  
+-------+------------+------+-----+---------+-------+  
| a | tinyint(4) | YES | | NULL | |  
+-------+------------+------+-----+---------+-------+  
1 row in set (0.26 sec)

mysql> alter table a change a a tinyint(1);  
Query OK, 0 rows affected (0.09 sec)  
Records: 0 Duplicates: 0 Warnings: 0

mysql> insert into a values (100);  
Query OK, 1 row affected (0.00 sec)

mysql> select \* from a;  
+-----+  
| a |  
+-----+  
| 100 |  
+-----+  
1 row in set (0.00 sec)

Some code to better explain what I described.  
mysql> create table b ( b int (4));  
Query OK, 0 rows affected (0.25 sec)

mysql> insert into b values (10000);  
Query OK, 1 row affected (0.00 sec)

mysql> select \* from b;  
+-------+  
| b |  
+-------+  
| 10000 |  
+-------+  
1 row in set (0.00 sec)

mysql> alter table b change b b int(11);  
Query OK, 1 row affected (0.00 sec)  
Records: 1 Duplicates: 0 Warnings: 0

mysql> select \* from b;  
+-------+  
| b |  
+-------+  
| 10000 |  
+-------+  
1 row in set (0.00 sec)

mysql> alter table b change b b int(11) zerofill;  
Query OK, 1 row affected (0.00 sec)  
Records: 1 Duplicates: 0 Warnings: 0

mysql> select \* from b;  
+-------------+  
| b |  
+-------------+  
| 00000010000 |  
+-------------+  
1 row in set (0.00 sec)

mysql> alter table b change b b int(4) zerofill;  
Query OK, 1 row affected (0.08 sec)  
Records: 1 Duplicates: 0 Warnings: 0

mysql> select \* from b;  
+-------+  
| b |  
+-------+  
| 10000 |  
+-------+  
1 row in set (0.00 sec)

mysql> alter table b change b b int(6) zerofill;  
Query OK, 1 row affected (0.01 sec)  
Records: 1 Duplicates: 0 Warnings: 0

mysql> select \* from b;  
+--------+  
| b |  
+--------+  
| 010000 |  
+--------+  
1 row in set (0.00 sec)