### 1．数字类型

***M***指示整数类型的最大显示宽度。最大显示宽度为255.显示宽度与类型可以包含的值范围无关，对于浮点型和定点型，***M***是可以存储的总位数。

如果您ZEROFILL为数字列指定，MySQL会自动将该UNSIGNED 属性添加到列中。

允许UNSIGNED 属性的数字数据类型也允许SIGNED。但是，这些数据类型是默认签名的，所以该 SIGNED属性不起作用。

SERIAL是别名BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT UNIQUE。

当在一个类型的整数值之间使用减法时UNSIGNED，除非[NO\_UNSIGNED\_SUBTRACTION](https://dev.mysql.com/doc/refman/5.7/en/sql-mode.html#sqlmode_no_unsigned_subtraction)启用S​​QL模式，否则结果是无符号的 。

* [BIT[(***M***)]](https://dev.mysql.com/doc/refman/5.7/en/bit-type.html)

一个比特值类型。***M***指示每个值的位数，从1到64.如果***M***省略，则缺省值为1 。

* [TINYINT[(](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[M](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[)] [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)

一个非常小的整数。签署的范围是 -128到127。无符号的范围是0到 255。

* [BOOL](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)， [BOOLEAN](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html)

这些类型是同义词 [TINYINT(1)](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html)。值为零被认为是错误的。非零值被认为是正确的：

mysql> SELECT IF(0, 'true', 'false');

+------------------------+

| IF(0, 'true', 'false') |

+------------------------+

| false |

+------------------------+

mysql> SELECT IF(1, 'true', 'false');

+------------------------+

| IF(1, 'true', 'false') |

+------------------------+

| true |

+------------------------+

mysql> SELECT IF(2, 'true', 'false');

+------------------------+

| IF(2, 'true', 'false') |

+------------------------+

| true |

+------------------------+

mysql> SELECT IF(0 = FALSE, 'true', 'false');

+--------------------------------+

| IF(0 = FALSE, 'true', 'false') |

+--------------------------------+

| true |

+--------------------------------+

mysql> SELECT IF(1 = TRUE, 'true', 'false');

+-------------------------------+

| IF(1 = TRUE, 'true', 'false') |

+-------------------------------+

| true |

+-------------------------------+

mysql> SELECT IF(2 = TRUE, 'true', 'false');

+-------------------------------+

| IF(2 = TRUE, 'true', 'false') |

+-------------------------------+

| false |

+-------------------------------+

mysql> SELECT IF(2 = FALSE, 'true', 'false');

+--------------------------------+

| IF(2 = FALSE, 'true', 'false') |

+--------------------------------+

| false |

+--------------------------------+

* [SMALLINT[(***M***)] [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html)

一个小整数。签署的范围是 -32768到32767。无符号的范围是0到 65535。

* [MEDIUMINT[(](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[M](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[)] [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)

一个中等大小的整数 签署的范围是 -8388608到8388607。无符号的范围是0到 16777215。

* [INT[(](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[M](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[)] [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)

一个正常大小的整数。签署的范围是 -2147483648到 2147483647。无符号的范围是 0到4294967295。

* [INTEGER[(](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[M](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[)] [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)

这是一个同义词 [INT](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html)。

* [BIGINT[(](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[M](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)***[)] [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/integer-types.html" \o "11.2.1整型（精确值） -  INTEGER，INT，SMALLINT，TINYINT，MEDIUMINT，BIGINT)

一个大整数 签署的范围是 -9223372036854775808到 9223372036854775807。无符号的范围是0到 18446744073709551615。

* [DECIMAL[(](https://dev.mysql.com/doc/refman/5.7/en/fixed-point-types.html" \o "11.2.2定点类型（准确值） -  DECIMAL，NUMERIC)***[M](https://dev.mysql.com/doc/refman/5.7/en/fixed-point-types.html" \o "11.2.2定点类型（准确值） -  DECIMAL，NUMERIC)***[[,](https://dev.mysql.com/doc/refman/5.7/en/fixed-point-types.html" \o "11.2.2定点类型（准确值） -  DECIMAL，NUMERIC)***[D](https://dev.mysql.com/doc/refman/5.7/en/fixed-point-types.html" \o "11.2.2定点类型（准确值） -  DECIMAL，NUMERIC)***[])] [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/fixed-point-types.html" \o "11.2.2定点类型（准确值） -  DECIMAL，NUMERIC)

外带“ 精确 ”定点数。 ***M***是总位数（精度），***D***是小数点后的位数（标度）。小数点和（对于负数） -符号不计入 ***M***。如果 ***D***为0，则数值没有小数点或小数部分。（***M***） 的最大位数[DECIMAL](https://dev.mysql.com/doc/refman/5.7/en/fixed-point-types.html" \o "11.2.2定点类型（准确值） -  DECIMAL，NUMERIC)是65.支持的最大小数数（***D***）是30.如果***D***省略，则默认为0.如果***M***省略，则默认值为10。

* [FLOAT[(](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[M](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[,](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[D](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[)] [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)

一个小的（单精度）浮点数。允许值是-3.402823466E+38 对-1.175494351E-38， 0以及1.175494351E-38 对3.402823466E+38。这些是基于IEEE标准的理论极限。实际的范围可能会稍微小一点取决于您的硬件或操作系统。

***M***是总位数，***D***是小数点后面的位数。如果***M*** 和***D***省略，值被存储到硬件允许的限制。单精度浮点数精确到小数点后7位。

* [DOUBLE[(](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[M](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[,](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[D](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[)] [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)

一个正常大小的（双精度）浮点数。允许值是 -1.7976931348623157E+308对 -2.2250738585072014E-308， 0以及 2.2250738585072014E-308对1.7976931348623157E+308。这些是基于IEEE标准的理论极限。实际的范围可能会稍微小一点取决于您的硬件或操作系统。

***M***是总位数，***D***是小数点后面的位数。如果***M*** 和***D***省略，值被存储到硬件允许的限制。双精度浮点数精确到小数点后15位。

* [FLOAT(](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[p](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)***[) [UNSIGNED] [ZEROFILL]](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)

一个浮点数。***p*** 表示以位为单位的精度，但MySQL仅使用该值来确定是使用 [FLOAT](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html)还是 [DOUBLE](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html)用于结果数据类型。如果***p***从0到24，则数据类型将变为[FLOAT](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)无***M***或***D***值。如果 ***p***从25到53，则数据类型将变为[DOUBLE](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)无 ***M***或***D*** 值。结果列的范围与本节前面所述的单精度[FLOAT](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)或双精度[DOUBLE](https://dev.mysql.com/doc/refman/5.7/en/floating-point-types.html" \o "11.2.3浮点类型（近似值） -  FLOAT，DOUBLE)数据类型相同。

2日期和时间类型CREATE TABLE t1 (t TIME(3), dt DATETIME(6));

该***fsp***值如果给出，则必须在0到6的范围内。值为0表示没有小数部分。如果省略，则默认精度为0 。

表中的 任何[TIMESTAMP](https://dev.mysql.com/doc/refman/5.7/en/datetime.html)或 [DATETIME](https://dev.mysql.com/doc/refman/5.7/en/datetime.html)列都可以具有自动初始化和更新属性。

* [DATE](https://dev.mysql.com/doc/refman/5.7/en/datetime.html)

支持的范围是 '1000-01-01'到 '9999-12-31'。MySQL [DATE](https://dev.mysql.com/doc/refman/5.7/en/datetime.html)以'YYYY-MM-DD'格式显示 值 ，但允许[DATE](https://dev.mysql.com/doc/refman/5.7/en/datetime.html" \o "11.3.1 DATE，DATETIME和TIMESTAMP类型) 使用字符串或数字将值分配给列。

* [DATETIME[(](https://dev.mysql.com/doc/refman/5.7/en/datetime.html" \o "11.3.1 DATE，DATETIME和TIMESTAMP类型)***[fsp](https://dev.mysql.com/doc/refman/5.7/en/datetime.html" \o "11.3.1 DATE，DATETIME和TIMESTAMP类型)***[)]](https://dev.mysql.com/doc/refman/5.7/en/datetime.html" \o "11.3.1 DATE，DATETIME和TIMESTAMP类型)

日期和时间组合。支持的范围是 '1000-01-01 00:00:00.000000'到 '9999-12-31 23:59:59.999999'。MySQL [DATETIME](https://dev.mysql.com/doc/refman/5.7/en/datetime.html)以'YYYY-MM-DD HH:MM:SS[.fraction]'格式显示值 ，但允许[DATETIME](https://dev.mysql.com/doc/refman/5.7/en/datetime.html" \o "11.3.1 DATE，DATETIME和TIMESTAMP类型)使用字符串或数字将值分配给 列。

***fsp***可以给出从0到6范围内 的可选值来指定小数秒精度。值为0表示不存在小数部分。如果省略，则默认精度为0。

* [TIMESTAMP[(](https://dev.mysql.com/doc/refman/5.7/en/datetime.html" \o "11.3.1 DATE，DATETIME和TIMESTAMP类型)***[fsp](https://dev.mysql.com/doc/refman/5.7/en/datetime.html" \o "11.3.1 DATE，DATETIME和TIMESTAMP类型)***[)]](https://dev.mysql.com/doc/refman/5.7/en/datetime.html" \o "11.3.1 DATE，DATETIME和TIMESTAMP类型)

时间戳。范围是'1970-01-01 00:00:01.000000'UTC到'2038-01-19 03:14:07.999999'UTC。 [TIMESTAMP](https://dev.mysql.com/doc/refman/5.7/en/datetime.html)值以自纪元（'1970-01-01 00:00:00'UTC）以来的秒数存储。***fsp***可以给出从0到6范围内 的可选值来指定小数秒精度。值为0表示不存在小数部分。如果省略，则默认精度为0。

* [TIME[(***fsp***)]](https://dev.mysql.com/doc/refman/5.7/en/time.html)

一次 范围是'-838:59:59.000000' 到'838:59:59.000000'。MySQL [TIME](https://dev.mysql.com/doc/refman/5.7/en/time.html)以'HH:MM:SS[.fraction]'格式显示 值 ，但允许[TIME](https://dev.mysql.com/doc/refman/5.7/en/time.html" \o "11.3.2时间类型) 使用字符串或数字将值分配给列。

***fsp***可以给出从0到6范围内 的可选值来指定小数秒精度。值为0表示不存在小数部分。如果省略，则默认精度为0。

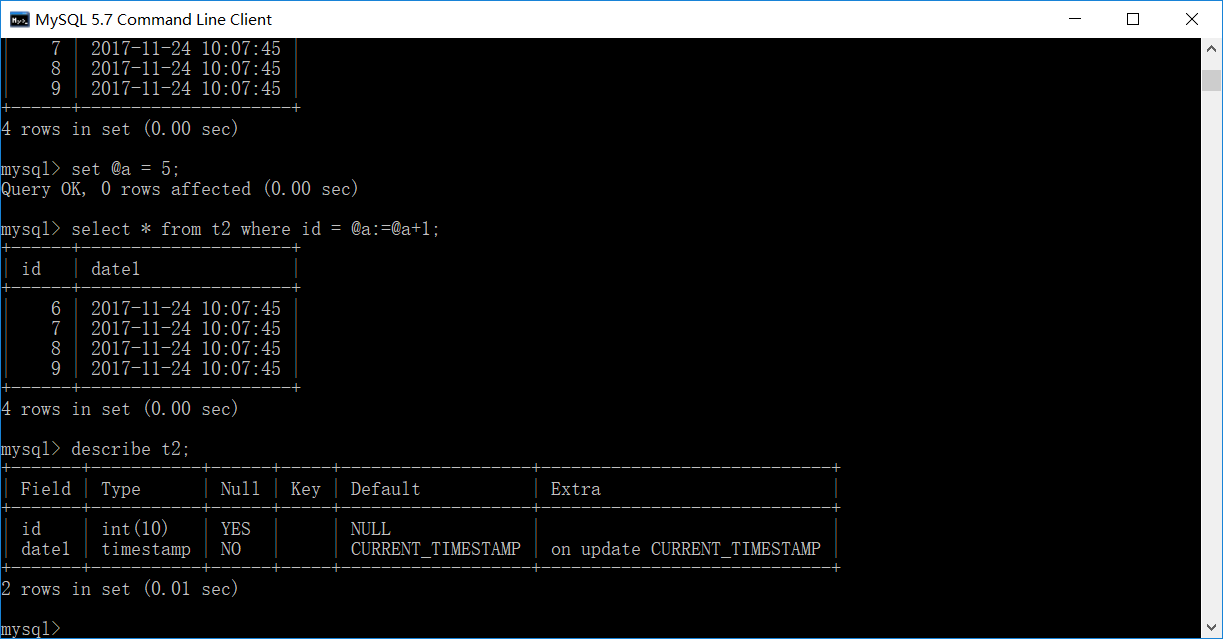
* [YEAR[(4)]](https://dev.mysql.com/doc/refman/5.7/en/year.html" \o "11.3.3年度类型)

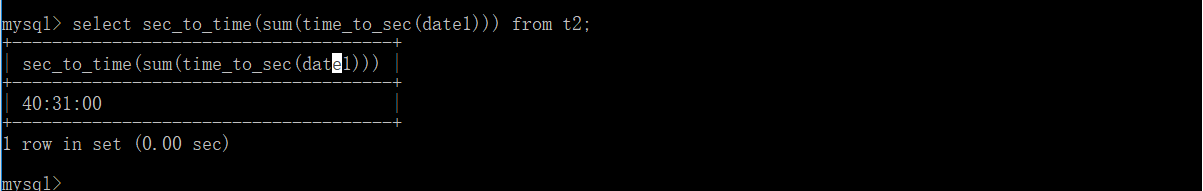
一年四位数的格式。MySQL [YEAR](https://dev.mysql.com/doc/refman/5.7/en/year.html)以YYYY格式显示 值 ，但允许[YEAR](https://dev.mysql.com/doc/refman/5.7/en/year.html" \o "11.3.3年度类型)使用字符串或数字将值分配给列。值显示为 1901到2155和 0000。

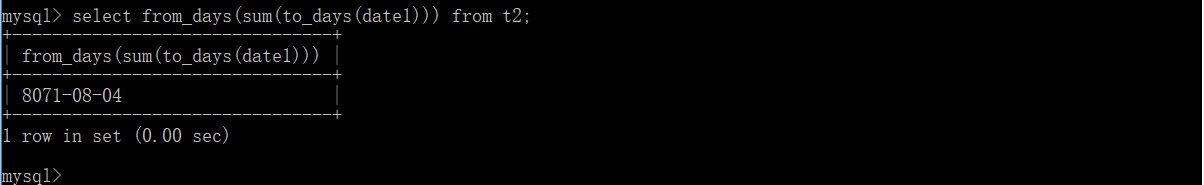
在[SUM()](https://dev.mysql.com/doc/refman/5.7/en/group-by-functions.html" \l "function_sum)和 [AVG()](https://dev.mysql.com/doc/refman/5.7/en/group-by-functions.html#function_avg)聚合函数不具有时间价值的工作。（它们将值转换为数字，在第一个非数字字符后丢失所有内容。）要解决此问题，请转换为数值单位，执行聚合操作，然后转换回临时值。例子：

SELECT SEC\_TO\_TIME(SUM(TIME\_TO\_SEC(*time\_col*))) FROM *tbl\_name*;

SELECT FROM\_DAYS(SUM(TO\_DAYS(*date\_col*))) FROM *tbl\_name*;







### 3字符串类型概述

许多字符串数据类型的列定义可以包括指定字符集或列的排序规则的属性。这些属性适用于 [CHAR](https://dev.mysql.com/doc/refman/5.7/en/char.html)， [VARCHAR](https://dev.mysql.com/doc/refman/5.7/en/char.html)，的 [TEXT](https://dev.mysql.com/doc/refman/5.7/en/blob.html)类型 [ENUM](https://dev.mysql.com/doc/refman/5.7/en/enum.html)和 [SET](https://dev.mysql.com/doc/refman/5.7/en/set.html)数据类型：

* 该CHARACTER SET属性指定字符集，该属性指定字符集COLLATE的排序规则。例如：
* CREATE TABLE t
* (
* c1 VARCHAR(20) CHARACTER SET utf8, //设置字符集
* c2 TEXT CHARACTER SET latin1 COLLATE latin1\_general\_cs //设置为区分大小写排序
* );
* 该BINARY属性是指定表默认字符集和该字符集的二进制（\_bin）排序规则的简写。在这种情况下，比较和排序是基于数字字符代码值。
* 该ASCII属性是速记的 CHARACTER SET latin1。
* 该UNICODE属性是速记的 CHARACTER SET ucs2。
* [NATIONAL] CHAR[(***M***)] [CHARACTER SET ***charset\_name***] [COLLATE ***collation\_name***]

一个固定长度的字符串，在存储时总是用空格填充指定的长度。 ***M***代表字符的列长度。范围***M***是0到255.如果***M***省略，则长度为1。

[CHAR](https://dev.mysql.com/doc/refman/5.7/en/char.html)除非[PAD\_CHAR\_TO\_FULL\_LENGTH](https://dev.mysql.com/doc/refman/5.7/en/sql-mode.html#sqlmode_pad_char_to_full_length) 启用了SQL模式，否则 当检索到值 时删除尾随空格 。

* [NATIONAL] VARCHAR(***M***) [CHARACTER SET ***charset\_name***] [COLLATE ***collation\_name***]

一个可变长度的字符串。***M*** 表示以字符表示的最大列长度。范围***M***是0到65,535。a的有效最大长度 [VARCHAR](https://dev.mysql.com/doc/refman/5.7/en/char.html)取决于最大行大小（65,535字节，在所有列中共享）和使用的字符集。例如， utf8字符每个字符最多可能需要三个字节，因此[VARCHAR](https://dev.mysql.com/doc/refman/5.7/en/char.html" \o "11.4.1 CHAR和VARCHAR类型)使用utf8字符集的 列 最多可以声明为21,844个字符。

* [ENUM('](https://dev.mysql.com/doc/refman/5.7/en/enum.html" \o "11.4.4 ENUM类型)***[value1](https://dev.mysql.com/doc/refman/5.7/en/enum.html" \o "11.4.4 ENUM类型)***[','](https://dev.mysql.com/doc/refman/5.7/en/enum.html" \o "11.4.4 ENUM类型)***[value2](https://dev.mysql.com/doc/refman/5.7/en/enum.html" \o "11.4.4 ENUM类型)***[',...) [CHARACTER SET](https://dev.mysql.com/doc/refman/5.7/en/enum.html" \o "11.4.4 ENUM类型)***[charset\_name](https://dev.mysql.com/doc/refman/5.7/en/enum.html" \o "11.4.4 ENUM类型)***[] [COLLATE](https://dev.mysql.com/doc/refman/5.7/en/enum.html" \o "11.4.4 ENUM类型)***[collation\_name](https://dev.mysql.com/doc/refman/5.7/en/enum.html" \o "11.4.4 ENUM类型)***[]](https://dev.mysql.com/doc/refman/5.7/en/enum.html" \o "11.4.4 ENUM类型) 枚举最大可以有 65535 个成员值。如果将一个数字存储到一个 ENUM 中，数字被当作为一个索引值，并且存储的值是该索引值所对应的枚举成员。(但是，这在 LOAD DATA 将不能工作，因为它视所有的输入均为字符串。) 在一个 ENUM 字符串中存储数字是不明智的，因为它可能会打乱思维。
* SHOW COLUMNS FROM table\_name LIKE enum\_colum

ENUM 是一个字符串对象，其值通常选自一个允许值列表中，该列表在表创建时的列规格说明中被明确地列举。

在下列某些情况下，值也可以是空串("") 或 NULL：

如果将一个无效值插入一个 ENUM (即，一个不在允许值列表中的字符串)，空字符串将作为一个特殊的错误值被插入。事实上，这个字符串有别于一个"普通的"空字符串，因为这个字符串有个数字索引值为 0。

如果一个 ENUM 被声明为 NULL，NULL 也是该列的一个合法值，并且该列的缺省值也将为 NULL 。如果一个 ENUM 被声明为 NOT NULL，该列的缺省值将是该列表所允许值的第一个成员。

每个枚举值均有一个索引值：

在列说明中列表值所允许的成员值被从 1 开始编号。

空字符串错误值的索引值为 0。这就意味着，你可以使用下面所示的 SELECT 语句找出被赋于无效 ENUM值的记录行。

mysql> SELECT \* FROM tbl\_name WHERE enum\_col=0;

NULL 值的索引值为 NULL。

* [SET('](https://dev.mysql.com/doc/refman/5.7/en/set.html" \o "11.4.5 SET类型)***[value1](https://dev.mysql.com/doc/refman/5.7/en/set.html" \o "11.4.5 SET类型)***[','](https://dev.mysql.com/doc/refman/5.7/en/set.html" \o "11.4.5 SET类型)***[value2](https://dev.mysql.com/doc/refman/5.7/en/set.html" \o "11.4.5 SET类型)***[',...) [CHARACTER SET](https://dev.mysql.com/doc/refman/5.7/en/set.html" \o "11.4.5 SET类型)***[charset\_name](https://dev.mysql.com/doc/refman/5.7/en/set.html" \o "11.4.5 SET类型)***[] [COLLATE](https://dev.mysql.com/doc/refman/5.7/en/set.html" \o "11.4.5 SET类型)***[collation\_name](https://dev.mysql.com/doc/refman/5.7/en/set.html" \o "11.4.5 SET类型)***[]](https://dev.mysql.com/doc/refman/5.7/en/set.html" \o "11.4.5 SET类型)

一个字符串对象，该对象可以具有零个或多个值，每个都必须从值的列表中选择 ， ， 值在内部表示为整数。

Set与enume的差别在于 set一次可以取多个值使用，而enum一次只能使用一个值