

MySQL数据类型

整型

整数类型	字节	无符号范围	有符号范围
tinyint	1	0 ~ 255	-128 ~ 127
smallint	2	0 ~ 65535	-32768 ~ 32767
mediumint	3	0 ~ 16777215	-8388608 ~ 8388607
int	4	0 ~ 4294967295	-2147483648 ~ 2147483647
bigint	8	0 ~ 18446744073709551615	-9223372036854775808~9223372036854775808

书写格式: 数据类型(展示宽度)

默认展示宽度:

有符号 无符号

- tinyint 4位 3位
- smallint 6位 5位
- mediumint 9位 8位
- int 11位 10位
- bigint 20位 19位

注意:

1. 这里的 **展示宽度** 与 **存储字节** 无关.
2. 实际插入的 **数据宽度 大于 设置的宽度**, 数据依旧会插入**成功**, 原先设置宽度则失效.
3. 实际插入的 **数据宽度 大于 默认的宽度**, 则会**报错**
4. 实际插入的 **数据值大小 大于 整数范围**, 则会**报错**

例如:

整数：987 占3位
整数：-987 占4位

设定字段'age' 为 tinyint(2)
1. 若插入 age = 10, 则成功
2. 若插入 age = 123, 依旧成功. 看：注意点2
3. 若插入 age = 999999, 则报错. 看：注意点3
4. 若插入 age = 300, 则报错. 看：注意点4

浮点型

浮点类型	字节数	负数范围	非负数范围
float	4	-3.402823466 E +38 ~ -1.175494351 E -38	0 和 1.175494351 E -38 ~ 3.402823466 E +38
double	8	-1.7976931348623157 E +308 ~ -2.2250738585072014 E -308	0 和 2.2250738585072014 E -308 1.7976931348623157 E +308
decimal(M,D)	M+2	同 double 类型	同 double 类型

其中, decimal 被称之为 **定点型**

书写格式: 数据类型(M , D)

- **M** : 精度, **数据的总长度** (小数点不占长度)
- **D** : 标度, **小数位的长度**

注意:

1. 若实际插入的 数据精度 **大于 定义的精度**, 则会**四舍五入**, 使值的精度达到要求
2. decimal定点是以 **字符串形式存储**, 所以 精度比浮点要高
3. 若对**精度要求较高**, 推荐使用**decimal定点**, 因为**浮点容易出现误差**.

日期

日期类型	字节	日期格式	最小值	最大值	零值
datetime	8	YYYY-MM-DD HH:MM:SS	1000-01-01 00:00:00	9999-12-31 23:59:59	0000-00-00 00:00:00
timestamp	4	YYYY-MM-DD HH:MM:SS	1970-01-01 00:00:00	20371231 235959	0000-00-00 00:00:00
timestamp	4	YYYYMMDDHHMMSS	19700101 000000	20371231 235959	00000000000000
date	4	YYYY-MM-DD	1000-01-01	9999-12-31	0000-00-00
time	3	HH:MM:SS	-838:59:59	838:59:59	00:00:00
year	1	YYYY	1901	2155	0000

书写格式: 数据类型

注意:

1. timestamp 具有两种写法, 可任选
2. 若追求**效率**, 有时可以用 int 代替日期类型, 用 **int** 存储 **时间戳**.

字符串

字符串类型	字节	描述
char(M)	M	M 的范围 0 ~ 255 之间的整数
varchar(M)	M	M 的范围 0 ~ 65536 之间的整数
tinyblob		0~255
blob		0~65535
mediumblob		0~167772150
longblob		0~4294967295
tinytext		0~255
text		0~65535
mediumtext		0~167772150
longtext		0~4294967295
varbinary(M)	M	0~M
binary(M)	M	0~M

书写格式: 数据类型(字节长度)

注意:

1. **char(M)** 实际插入的 数据长度 必须 **小于等于** M (长度不会计算**最后面的连续空格**)
2. **varchar(M)** 实际插入的 数据长度 必须 **小于等于** M (长度会计算**最后面的连续空格**)
3. 其余了解即可

其余类型

其余类型使用频率较低, 这里只是简单的提及一下.

MySQL 还有 **enum** 枚举类型, **set** 类型