

1、整形

- tinyint: 1字节, 非常小的整数, 带符号: -128~127, 不带符号: 0~255
 - 1字节: 8位, 11111111
 - 如果是带符号位, 最前面那个表示符号位
 - 1Byte = 8bit(位)
 - 1K = 1024B
 - 1M = 1024K
 - 1G = 1024M
 - 1T = 1024G
- int: 4字节, 标准整形, 带符号: -2147483648~2147483647, 不带符号: 0~4294967295;
 - 默认11, 表示的是显示宽度, 还有比如year(4), 显示的是4位年, 跟字符串是不一样的
 - int(11) 11 代表的并不是长度, 而是字符的显示宽度, 在字段类型为 int 时, 无论你显示宽度设置为多少, int 类型能存储的最大值和最小值永远都是固定的。
 - zerofill 加上可以好理解一点

```
#      zerofill, int不足4位展示的时候补0
create table if not exists int_test(id int(4) zerofill);
desc int_test;
insert into int_test values(11111111);
insert into int_test values(22222222);
insert into int_test values(1);
insert into int_test values(33);
insert into int_test values(444);
insert into int_test values(555);

select * from int_test;

mysql>
mysql> select * from int_test;
+-----+
| id    |
+-----+
| 11111111 |
| 11111111 |
| 22222222 |
| 0001 |
| 0033 |
| 0444 |
| 0555 |
+-----+
7 rows in set (0.00 sec)

mysql>
```

- 其他类型: smallint(2字节)、mediumint(3字节)、bigint(8字节);

类型	存储	最小值	最大值
	(Bytes)	(Signed/Unsigned)	(Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

2、浮点型

- 表示小数, 浮点数只能表示近似值, 不能表示精确值
 - 比如说 1/3 + 1/3 + 1/3 != 1 (有些编程语言里面)
- float: 4字节, 单精度浮点数;
 - 默认的float类型只能存6个数字(包括小数部分和整数部分)。整数或者小数超过6位, 超过的部分就会四舍五入
 - 默认float和float(m)一样 , m<=24时都是默认float类型, 都只能存6个数字(包括小数点前后的位数), 整数超过6位就被四舍五入
- double: 8字节, 双精度浮点数;

- decimal(m, d): m表示整数加小数的总位数, d表示小数位数, 比如decimal(4,2)
 - 21.926
 - 21.92
 - 21.9
 - 21.00
- 区别: float与double存储的都是近似值, decimal存储的是精确值。
 - 在数据库中涉及到money的金额, 一定要用decimal类型

3、字符串类型

- char(m): 最多可以容纳m个字符
- varchar(m): 最多可以容纳m个字符, 如果存储的字符数n少于m时, 实际占用的存储空间是L+1个字节
 - 这里的m是指的字符数, 并不是字节数, 而每个字符占用的字节数与编码格式有关
 - 其中L是n个字符所占用的字节数
 - 多出的1个字节是用来记录实际的存储长度的
 - 比如中文1个字符占用的是3个字节
 - Latin1字符集
 - 1个字符 = 1个字节
 - 1个汉字 = 2个字节
 - utf8
 - 1个字符 = 3个字节
 - 1个汉字 = 3个字节
 - gbk
 - 1个字符 = 2个字节
 - 1个汉字 = 2个字节
 - 还有各种编码格式, 太多。。。
- 其他类型: blob(longblob), text(longtext)
- char和varchar的区别:
 - char固定长度, 每个值占用相同的字符数, 不够的位数mysql会在它的右边用空格补齐
 - varchar是一种可变长度的类型, 每个值占用对应的字符数再加上一个用来记录其长度的字节即L+1
 - 使用时如何选择?
 - 如果数据都有相同的长度, 比如手机号、身份证号码等, 选用varchar会多占用空间, 因为其内部会多占用一位来存储其长度;
 - 如果数据长度不一, 那么选用varchar能节省存储空间
 - 而char不论字符长度都需要占用相同的空间, 即使是空值也不例外

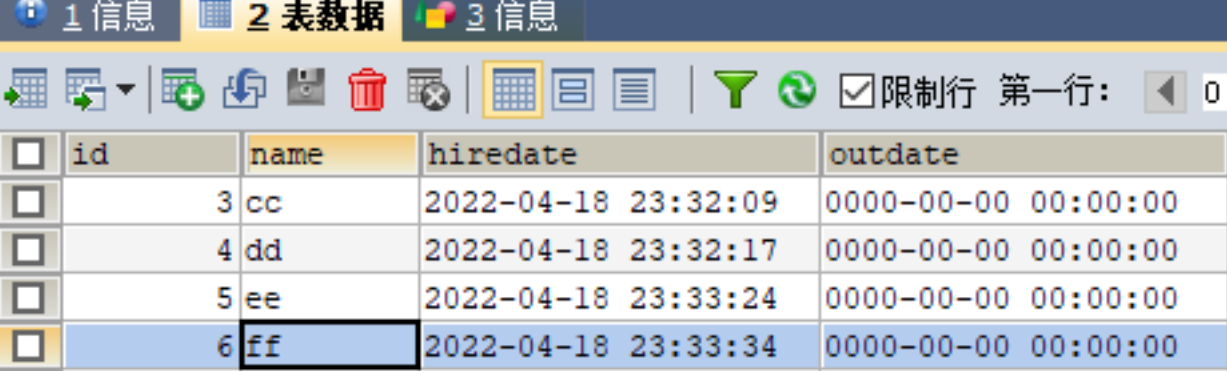
```
drop table if exists user;
create table if not exists user
(id int,
name varchar(12),
age int(4)
) default charset=utf8;

insert into user(id, name,age) values(1,'ab',21);
insert into user(id, name,age) values(2,'唐',21);

select * from user;
```

3、日期时间型

- year, 表示年, 如2021
- date, 日期型, 表示一个日期格式, 如2021-11-11
- time, 时间型, 表示一个时间格式, 如22:10:15
- datetime, 日期时间型, 表示一个日期+时间的组合格式, 如2021-11-11 22:10:15
- timestamp, 时间戳型, 如果某列的数据类型为timestamp且该列为not null, 当插入数据时没有指定该列的值时, 那么这个值会自动取值为当前的日期和时间; 在创建和修改数据行时, 如果没有明确的对timestamp列进行赋值, 则它就会自动取值为当前的日期和时间, 如果表中有多个timestamp列, 只有第一个会自动取值
 - 1970-1-1 00:00:00 从这个时间算中间隔得秒数



4、枚举类型

- enum: 枚举类型
 - 当一个字段的值只能在某个范围内取值时, 比如性别只能取男或者女可以用enum('男', '女')来表示
 - 这个字段在插入数据时只能取男或者女或者不传值
 - 每个枚举值有一个索引, 第一个值的索引为1
 - 枚举类型的好处是mysql在存储此类数据时, 直接转化为数字存储而不是字符串, 可以节省空间, 并且在表的.frm文件中存储"数字-字符串"之间的对应关系