

Unsigned 与 Signed 关键字

Signed:

在默认情况下声明的整型变量都是有符号的类型 (*char* 有点特别), 如果需声明无符号类型的话就需要在类型前加上 *unsigned*。无符号版本和有符号版本的区别就是无符号类型能保存 2 倍于有符号类型的正整数数据, 如 16 位系统中一个 *int* 能存储的数据的范围为 -32768~32767, 而 *unsigned* 能存储的数据范围则是 0~65535。由于在计算机中, 整数是以补码形式存放的。根据最高位的不同, 如果是 1, 有符号数的话就是负数; 如果是无符号数, 则都解释为正数。

整型的每一种都有无符号 (*unsigned*) 和有符号 (*signed*) 两种类型, 在默认情况下声明的整型变量都是有符号的类型 (*char* 有点特别), 如果需声明无符号类型的话就需要在类型前加上 *unsigned*。无符号版本和有符号版本的区别就是有符号类型需要使用一个 bit 来表示数字的正负, 比如 16 位系统中一个 *int* 能存储的数据的范围为 -32768 ~ 32767 (16 位 2 进制的最高位作为符号位 '1' 为负 '0' 为正), 而 *unsigned* 能存储的数据范围则是 0~65535 (这个最高位不用做符号位, 所以是 2 的 16 次方, 一共 65536)。由于在计算机中, 整数是以补码形式存放的。根据最高位的不同, 如果是 1, 有符号数的话就是负数; 如果是无符号数, 则都解释为正数。同时在相同位数的情况下, 所能表达的整数范围变大。另外, *unsigned* 若省略后一个关键字, 大多数编译器都会认为是 *unsigned int*。

Unsigned:

整型的每一种都分为: 无符号 (*unsigned*) 和有符号 (*signed*) 两种类型 (*float* 和 *double* 总是带符号的), 在除 *char* 以外的数据类型中, 默认情况下声明的整型变量都是有符号的类型; *char* 在默认情况下总是无符号的。在除 *char* 以外的数据类型中, 如果需声明无符号类型的话就需要在类型前加上 *unsigned*。无符号版本和有符号版本的区别就是无符号类型能保存 2 倍于有符号类型的正整数数据, 比如 16 位系统中一个 *short* 能存储的数据的范围为 -32768~32767, 而 *unsigned* 能存储的数据范围则是 0~65535。由于在计算机中, 整数是以补码形式存放的。根据最高位的不同, 如果是 1, 有符号数的话就是负数; 如果是无符号数, 则都解释为正数。另外, *unsigned* 若省略后一个关键字, 大多数编译器都会认为是 *unsigned int*。

整型的每一种都有无符号 (*unsigned*) 和有符号 (*signed*) 两种类型, 在默认情况下声明的整型变量都是有符号的类型 (*char* 有点特别), 如果需声明无符号类型的话就需要在类型前加上 *unsigned*。无符号版本和有符号版本的区别就是有符号类型需要使用一个 bit 来表示数字的正负, 比如 16 位系统中一个 *int* 能存储的数据的范围为 -32768 ~ 32767 (16 位 2 进制的最高位作为符号位 '1' 为负 '0' 为正), 而 *unsigned* 能存储的数据范围则是 0~65535 (这个最高位不用做符号位, 所以是 2 的 16 次方, 一共 65536)。由于在计算机中, 整数是以补码形式存放的。根据最高位的不同, 如果是 1, 有符号数的话就是负数; 如果是无符号数, 则都解释为正数。同时在相同位数的情况下, 所能表达的整数范围变大。另外, *unsigned* 若省略后一个关键字, 大多数编译器都会认为是 *unsigned int*。

在 sql 语句中的意义

sql 语句中, 创建一个数据表时

```
create table user{user_id int unsigned}
```

当中的 *unsigned* 表示, 数据项 *user_id* 恒为正整数。

例如如果 tinyint 最大是 127，那 tinyintunsigned 最大可以到 $127 * 2$
unsigned 属性只针对整型，而 binary 属性只用于 char 和 varchar。