



链滴

MySQL 中 TIMESTAMP 问题 笔记

作者: [PeterChu](#)

原文链接: <https://ld246.com/article/1598695234892>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



1. 长度问题。

在一些博文中看到有这样的描述：

TIMESTAMP列类型

TIMESTAMP值可以从1970的某时的开始一直到2037年，精度为一秒，其值作为数字显示。

TIMESTAMP值显示尺寸的格式如下表所示：

:

| 列类型 | 显示格式 |
|---------------|----------------|
| TIMESTAMP(14) | YYYYMMDDHHMMSS |
| TIMESTAMP(12) | YYMMDDHHMMSS |
| TIMESTAMP(10) | YYMMDDHHMM |
| TIMESTAMP(8) | YYYYMMDD |
| TIMESTAMP(6) | YYMMDD |
| TIMESTAMP(4) | YYMM |
| TIMESTAMP(2) | YY |

“完整”TIMESTAMP格式是14位，但TIMESTAMP列也可以用更短的显示尺寸创造最常见的显示尺寸是6、8、12、和14。

你可以在创建表时指定一个任意的显示尺寸，但是定义列长为0或比14大均会被强制定义为列长14。

列长在从1 ~ 13范围的奇数值尺寸均被强制为下一个更大的偶数。

列如：

定义字段长度 强制字段长度

TIMESTAMP(0) -> TIMESTAMP(14)

TIMESTAMP(15)-> TIMESTAMP(14)

TIMESTAMP(1) -> TIMESTAMP(2)

TIMESTAMP(5) -> TIMESTAMP(6)

所有的TIMESTAMP列都有同样的存储大小，

使用被指定的时期时间值的完整精度（14位）存储合法的值不考虑显示尺寸。

不合法的日期，将会被强制为0存储

*这有几个含意： *

- 1、虽然你建表时定义了列TIMESTAMP(8)，但在你进行数据插入与更新时TIMESTAMP列实际上保存了14位的数据（包括年月日时分秒），只不过在你进行查询时MySQL返回给你的是8位的年月日数据。如果你使用ALTER TABLE拓宽一个狭窄的TIMESTAMP列，以前被“隐蔽”的信息将被显示。
- 2、同样，缩小一个TIMESTAMP列不会导致信息失去，除了感觉上值在显示时，较少的信息被显示出。
- 3、尽管TIMESTAMP值被存储为完整精度，直接操作存储值的唯一函数是UNIX_TIMESTAMP(); 由于MySQL返回TIMESTAMP列的列值是进过格式化后的检索的值，这意味着你可能不能使用某些函数来操作TIMESTAMP列（例如HOUR()或SECOND()），除非TIMESTAMP值的相关部分被包含在格式化的值中。例如，一个TIMESTAMP列只有被定义为TIMESTAMP(10)以上时，TIMESTAMP列的HH部分才会被显示，因此在更短的TIMESTAMP值上使用HOUR()会产生一个不可预知的结果。
- 4、不合法TIMESTAMP值被变换到适当类型的“零”值(0000000000000000)。(DATETIME,DATE亦然)

但是，当使用如下语句时：

```
ALTER TABLE myemp ADD COLUMN hiredate1 TIMESTAMP(14);
```

会提示：1426 - Too-big precision 14 specified for 'hiredate1'. Maximum is 6.

因此，MySQL 中的TIMESTAMP 数据长度到底是多少呢？（本地MySQL版本为8.0）

2. 另外，MySQL的timestamp字段可以使用的范围是多少。

timestamp类型的起始时间是1970-01-01 00:00:01 UTC，和时区是关系的。如果我没有理解错的话MySQL将timestamp类型的值保存的时候，会从当前时区转成UTC时间，正好解释了前面1970-01-01 00:00:00或1970-01-01 00:00:01两个值保存时出错的问题了。从当前时区转成UTC时间需要减去『小时』，结果就不在timestamp类型的范围内了。