生产环境中部署着各种版本的MySQL，包括MySQL 5.5/5.6/5.7三个大版本和N个小版本，由于MySQL在向上兼容性较差，导致相同SQL在不同版本上表现各异，下面从几个方面来详细介绍时间戳数据类型。

#### 时间戳数据存取

在MySQL上述三个大版本中，默认时间戳(Timestamp)类型的取值范围为'1970-01-01 00:00:01' UTC 至'2038-01-19 03:14:07' UTC，数据精确到秒级别，该取值范围包含约22亿个数值，因此在MySQL内部使用4个字节INT类型来存放时间戳数据：

1、在存储时间戳数据时，先将本地时区时间转换为UTC时区时间，再将UTC时区时间转换为INT格式的毫秒值(使用UNIX\_TIMESTAMP函数)，然后存放到数据库中。

2、在读取时间戳数据时，先将INT格式的毫秒值转换为UTC时区时间(使用FROM\_UNIXTIME函数)，然后再转换为本地时区时间，最后返回给客户端。

在MySQL 5.6.4及之后版本，可以将时间戳类型数据最高精确微秒(百万分之一秒)，数据类型定义为timestamp(N),N取值范围为0-6，默认为0，如需要精确到毫秒则设置为Timestamp(3)，如需要精确到微秒则设置为timestamp(6)，数据精度提高的代价是其内部存储空间的变大，但仍未改变时间戳类型的最小和最大取值范围。

#### 时间戳字段定义

时间戳字段定义主要影响两类操作：

1. 插入记录时，时间戳字段包含DEFAULT CURRENT\_TIMESTAMP，如插入记录时未指定具体时间数据则将该时间戳字段值设置为当前时间。

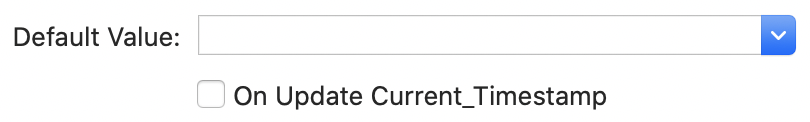
2、更新记录时，时间戳字段包含ON UPDATE CURRENT\_TIMESTAMP，如更新记录时未指定具体时间数据则将该时间戳字段值设置为当前时间。

CURRENT\_TIMESTAMP表示使用CURRENT\_TIMESTAMP()函数来获取当前时间，类似

于NOW()函数

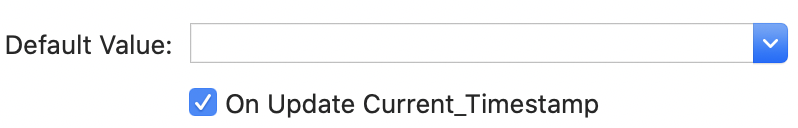
根据上面两类操作，时间戳列可以有四张组合定义，其含义分别为：

1、当字段定义为timestamp，表示该字段在插入和更新时都不会自动设置为当前时间。



2、当字段定义为timestamp DEFAULT CURRENT\_TIMESTAMP，表示该字段仅在插入且未指定值时被赋予当前时间，在更新时且未指定值时不做修改。

3、当字段定义为timestamp ON UPDATE CURRENT\_TIMESTAMP，表示该字段在插入且未指定值时被赋值为"0000-00-00 00:00:00",在更新且未指定值时更新为当前时间。



4、当字段定义为timestamp DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP，表示该字段在插入或更新时未指定值，则被赋值为当前时间。

在MySQL中执行的建表语句和最终表创建语句会存在差异，建议使用

SHOW CREATE TABLE TB\_XXX获取已创建表的建表语句。

#### 时间戳字段在MySQL各版本的使用差异

1、在MySQL 5.5及之前版本中，仅能对一个时间戳字段定义DEFUALT CURRENT\_TIMESTAMP或ON UPDATE CURRENT\_TIMESTAMP，但在MySQL 5.6和MySQL 5.7版本中取消了该限制;

2、在MySQL 5.6版本中参数explicit\_defaults\_for\_timestamp默认值为1，在MySQL 5.7版本中参数explicit\_defaults\_for\_timestamp默认值为0;

3、在MySQL 5.5和MySQL 5.7版本中timestamp类型默认为NOT NULL,但在MySQL 5.6版本中timestamp类型默认为NULL;

4、当建表语句中定于c1 timestamp 时，

在MySQL 5.5中等价于`c1` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP

ON UPDATE CURRENT\_TIMESTAMP;

在MySQL 5.6中等价于`c1` timestamp NULL DEFAULT NULL;

在MySQL 5.7中等价于`c1` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP;

5、当建表语句中c1 timestamp default 0时，

在MySQL 5.5中等价于`c1` timestamp NOT NULL DEFAULT '0000-00-00

00:00:00';

在MySQL 5.6中等价于`c1` timestamp NULL DEFAULT '0000-00-00 00:00:00';

在MySQL 5.7中等价于`c1` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00';

PS1: MySQL 5.6版本和MySQL 5.7版本中主要差异受参数

explicit\_defaults\_for\_timestamp的默认值影响。

PS2：当时间戳列的默认值为'0000-00-00 00:00:00'时，使用“不在时间戳取值范围内”的该默认值并不会产生警告。

#### 时间戳类型引发的异常

当MySQL参数time\_zone=system时，查询timestamp字段会调用系统时区做时区转换，而由于系统时区存在全局锁问题，在多并发大数据量访问时会导致线程上下文频繁切换，CPU使用率暴涨，系统响应变慢设置假死。

#### timestamp类型和datetime类型选择

在部分"数据库指导"文档中，会推荐使用timestamp类型代替datetime字段，其理由是timestamp类型使用4字节，而datetime字段使用8字节，但随着磁盘性能提升和内存成本降低，在实际生产环境中，使用timestamp类型并不会带来太多性能提升，反而可能因timestamp类型的定义和取值范围限制和影响业务使用。

在MySQL 5.6.4及之后版本，可以将时间戳类型(timestamp)数据最高精确微秒，也同样可以将时间类型(datetime)数据最高精确微秒，时间类型(datetime)同样可以获得timestamp类型相同的效果，如将字段定义为 dt1 DATETIME(3) NOT NULL DEFAULT NOW(3) ON UPDATE NOW(3); 时间类型(datetime)的存取范围'1000-01-01 00:00:00.000000' 至 '9999-12-31 23:59:59.999999'，能更好地存放各时间段的数据。

#### timestamp类型使用建议

1、在只关心数据最后更新时间的情况下，建议将时间戳列定义为TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP；

2、在关心创建时间和更新时间的情况下，建议将更新时间设置为时间戳字段，将创建时间定义为DAETIME 或 TIMESTAMP DEFAULT '0000-00-00 00:00:00'，并在插入记录时显式指定创建时间；

3、建议在表中只定义单个时间戳列，并显式定义DEFAULT 和 ON UPDATE属性；

4、虽然在MySQL中可以对时间戳字段赋值或更新，但建议仅在必要的情况下对时间戳列进行显式插入和更新；

5、建议将time\_zone参数设置为system外的值，如中国地区服务器设置为'+8:00'；

6、建议将MySQL线下测试版本和线上生产版本保持一致。