1292 Incorrect datetime value

"org.springframework.dao.DataIntegrityViolationException: ### Error updating database. Cause: com.mysql.jdbc.MysqlDataTruncation: Data truncation: Incorrect

datetime value: '2040-08-11 15:24:52.981'

for column 'updatetime' at row 1 ### The error may involve

com.ins.bcp.banktext.mappers.BankOffsetSumMapper.updateByExampleSelective-Inline

The error occurred while setting parameters ### SQL: update t_bank_offset_sum SET offset_sum_id = ?, updatetime = ?,

fee_status = ?, op_batch_no = ? WHERE (offset_sum_id = ?)

Cause: com.mysql.jdbc.MysqlDataTruncation: Data truncation: Incorrect datetime value: '2040-08-11 15:24:52.981'

update t_bank_offset_sum SET offset_sum_id = ?, updatetime = '2040-08-11 15:24:52.981', fee_status = ?, op_batch_no = ? WHERE (offset_sum_id = ?)

□

□

□

MySQL 日期类型：日期格式、所占存储空间、日期范围 比较。

日期类型	存储空间	日期格式	日期范围
datetime	8 bytes	YYYY-MM-DD HH:MM:SS	1000-01-01 00:00:00 ~ 9999-12-31 23:59:59
timestamp	4 bytes	YYYY-MM-DD HH:MM:SS	1970-01-01 00:00:01 ~ 2038
date	3 bytes	YYYY-MM-DD	1000-01-01 ~ 9999-12-31
year	1 bytes	YYYY	1901 ~ 2155

在 MySQL 中创建表时，对照上面的表格，很容易就能选择到合适自己的数据类型。不过到底是选择 datetime 还是 timestamp，可能会有点犯难。

这两个日期时间类型各有优点：datetime 的日期范围比较大；timestamp 所占存储空间比较小，只是 datetime 的一半。

另外，timestamp 类型的列还有个特性：默认情况下，在 insert, update 数据时，timestamp 列会自动以当前时间（CURRENT_TIMESTAMP）填充/更新。"自动"的意思就是，你不去管它，MySQL 会替你去处理。

timestamp所能存储的时间范围为：'1970-01-01 00:00:01.000000' 到 '2038-01-19 03:14:07.999999'。

datetime所能存储的时间范围为：'1000-01-01 00:00:00.000000' 到 '9999-12-31 23:59:59.999999'。

原因 ： timestamp类型只占4个字节，支持范围小，最大值为2038

参考 ： http://www.jb51.net/article/23966.htm

解决方法 ： 将timestamp类型改为datetime类型，
```
mysql> desc test;
+-------+-----------+------+-----+-------------------+-------+
| Field | Type      | Null | Key | Default           | Extra |
+-------+-----------+------+-----+-------------------+-------+
| t     | timestamp | YES  |     | CURRENT_TIMESTAMP |       |
+-------+-----------+------+-----+-------------------+-------+
1 row in set (0.00 sec)

mysql> select * from test;
+---------------------+
| t                   |
+---------------------+
| 2017-08-18 16:34:19 |
| 2017-08-18 16:34:27 |
| 2017-08-18 16:34:30 |
| 2017-08-18 16:34:31 |
| 2017-08-18 16:34:33 |
+---------------------+
5 rows in set (0.00 sec)
```

```
mysql> alter table test modify column t datetime;
Query OK, 5 rows affected (1.94 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from test;
+---------------------+
| t                   |
+---------------------+
| 2017-08-18 16:34:19 |
| 2017-08-18 16:34:27 |
| 2017-08-18 16:34:30 |
| 2017-08-18 16:34:31 |
| 2017-08-18 16:34:33 |
+---------------------+
5 rows in set (0.00 sec)

mysql> desc test;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| t     | datetime | YES  |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
1 row in set (0.01 sec)

mysql> alter table test alter column t set default now();
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the
right syntax to use near 'now()' at line 1
mysql> select version();
+------------+
| version()  |
+------------+
| 5.7.18-log |
+------------+
1 row in set (0.13 sec)

mysql>
```

但又有问题 ： 默认值没了

解决方法 ：
可以使用trigger来实现此功能。
```
create table test_time (
 idint(11),
 create_time datetime
);
delimiter |
create trigger default_datetime before insert on test_time
  foreach row
    if new.create_time is null then
      set new.create_time = now();
    end if;|
delimiter ;
```