

1. Java.sql.Date 类

a) 继承关系

Class Date

java.lang.Object
java.util.Date

All Implemented Interfaces:

Serializable, Cloneable, Comparable<Date>

已知直接子类:

Date, Time, Timestamp

b) 日期格式

utilDate: Thu Nov 01 11:15:35 CST 2018 | type = java.util.Date

c) 能否直接匹配数据库的 DATETIME 类型

不能, 需要经过转化, 详见 2、3、4。

d) 转化为毫秒 (ms)

utilDate.getTime(): 1541042135041 | type = long

2. 第一种转化方式 SimpleDateFormat 类

a) 实施细节

```
// String ---> DATETIME
SimpleDateFormat sdf = new SimpleDateFormat( pattern: "yyyy-MM-dd HH-mm-ss");
String formatDate = sdf.format(utilDate);
System.out.println("formatDate: " + formatDate + " | type = String");
ps.setString( parameterIndex: 1, formatDate);
```

b) 写入和读取结果

写入: *formatDate: 2018-11-01 11-15-35 | type = String*

读取: *getFormatData: 2018-11-01 11:15:35.0 | type = String*

c) 数据库中的写入结果

2018-11-01 11:15:35

3. 第二种转化方式 java.sql.Date 类

a) 实施细节

```
// java.sql.Date ---> DATETIME
java.sql.Date sqlDate = new java.sql.Date(utlDate.getTime());
System.out.println("sqlDate: " + sqlDate + " | type = java.sql.Date");
ps.setDate( parameterIndex: 2, sqlDate);
```

b) 写入和读取结果

写入: *sqlDate: 2018-11-01 | type = java.sql.Date*

读取: *getSqlData: 2018-11-01 | type = java.sql.Date*

c) 数据库中的写入结果

2018-11-01 00:00:00

4. 第三种转化方式 Timestamp 类

a) 实施细节

```
// java.sql.Timestamp ---> DATETIME
Timestamp timestamp = new Timestamp(utlDate.getTime());
System.out.println("timestamp: " + timestamp + " | type = java.sql.Timestamp");
ps.setTimestamp( parameterIndex: 4, timestamp);
```

b) 写入和读取结果

写入: *timestamp: 2018-11-01 11:15:35.041 | type = java.sql.Timestamp*

读取: *getTimestamp: 2018-11-01 11:15:35.0 | type = java.sql.Timestamp*

c) 数据库中的写入结果

2018-11-01 11:15:35

5. 对 java.sql.Time 类的转化说明

Time 类无法直接写入为 DATETIME 类型。

```
// java.sql.Time ---> DATETIME 不可行
Time time = new Time(utlDate.getTime());
System.out.println("time: " + time + " | type = java.sql.Time");
// ps.setTime(3, time);
```

输出结果: *time: 11:15:35 | type = java.sql.Time*

执行写入报错: *Data truncation: Incorrect datetime value*

6. 总结

针对上述四种写入方式，我们发现，对于数据库中的 DATETIME 类型，只有 2、3、4 三种能够成功写入，其中 2 和 4 能够保持完整的“年月日时分秒”信息，而 3 只会保持“年月日”信息；5 虽然能够输出“时分秒”的信息，但是无法与数据库的 DATETIME 类型匹配。

7. Java 数据类型与 Mysql 数据类型对照表

类型名称	显示长度	数据库类型	JAVA 类型	JDBC 类型 索引 (int)
VARCHAR	L+N	VARCHAR	java.lang.String	12
CHAR	N	CHAR	java.lang.String	1
BLOB	L+N	BLOB	java.lang.byte[]	-4
TEXT	65535	VARCHAR	java.lang.String	-1
INTEGER	4	INTEGER UNSIGNED	java.lang.Long	4
TINYINT	3	TINYINT UNSIGNED	java.lang.Integer	-6
SMALLINT	5	SMALLINT UNSIGNED	java.lang.Integer	5
MEDIUMINT	8	MEDIUMINT UNSIGNED	java.lang.Integer	4
BIT	1	BIT	java.lang.Boolean	-7
BIGINT	20	BIGINT UNSIGNED	java.math.BigInteger	-5
FLOAT	4+8	FLOAT	java.lang.Float	7
DOUBLE	22	DOUBLE	java.lang.Double	8
DECIMAL	11	DECIMAL	java.math.BigDecimal	3
BOOLEAN	1	同 TINYINT		

ID	11	PK (INTEGER UNSIGNED)	java.lang.Long	4
DATE	10	DATE	java.sql.Date	91
TIME	8	TIME	java.sql.Time	92
DATETIME	19	DATETIME	java.sql.Timestamp	93
TIMESTAMP	19	TIMESTAMP	java.sql.Timestamp	93
YEAR	4	YEAR	java.sql.Date	91

注：此信息来源于互联网，并未亲自考证。

8. java 和 mysql 之间的时间日期类型传递

惊奇发现，其实 CSDN 博客上面已经有大佬做出了比较系统的总结。链

接如下：<https://blog.csdn.net/weinianjie1/article/details/6310770>

9. GitHub 源码

欢迎下载。