

JAVA基础：谨慎使用Date和Time类 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_JAVA_E5_9F_BA_E7_A1_80_c104_145335.htm JAVA里提供的日期和时间类，`java.sql.Date`和`java.sql.Time`，只会从数据库里读取某部分值，这有时会导致丢失数据。例如一个包含2002/05/22 5:00:57 PM的字段，读取日期时得到的是2002/05/22，而读取时间时得到的是5:00:57 PM. 你需要了解数据库里存储时间的精度。有些数据库，比如MySQL,精度为毫秒，然而另一些数据库，包括Oracle,存储SQL DATE类型数据时，毫秒部分的数据是不保存的。以下操作中容易出现不易被发现的BUG：获得一个JAVA里的日期对象。从数据库里读取日期试图比较两个日期对象是否相等。如果毫秒部分丢失，本来认为相等的两个日期对象用`Equals`方法可能返回`false`。`java.sql.Timestamp`类比`java.util.Date`类精确度要高。这个类包含了一个`getTime()`方法，但是它不会返回额外精度部分的数据，因此必须使用`getNanos()`方法。有毫微秒部分（即额外精度部分）的数值可能比没有这个部分的数值的大一个毫秒。如果你知道你使用的数据库保存了毫秒部分，可以用下面的代码获得的数值：
`long time=timestamp.getTime() timestamp.getNanos()/1000000.`
毫微秒的部分就是为什么一个`java.sql.Timestamp`对象不等于一个`java.util.Date`对象，而一个`java.util.Date`对象可能等于一个`java.sql.Timestamp`对象的原因。这使得`equals`方法应当保持的对称被打破了。使用`java.sql`包中的时间相关的类是重要的，但是它们也可能导致人为的错误。这是因为，这些类的复杂性虽然已经在`javadoc`里面清楚的描述了，但是由于这些类

看上去很简单而很少有人看这些文档 100Test 下载频道开通，
各类考试题目直接下载。详细请访问 www.100test.com