

Java程序中解决数据库超时与死锁Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/559/2021_2022_Java_E7_A8_8B_E5_BA_8F_c104_559573.htm 简介 每个使用关系型数据库的程序都可能遇到数据死锁或不可用的情况，而这些情况需要在代码中编程来解决.本文主要介绍与数据库事务死锁等情况相关的重试逻辑概念，此外，还会探讨如何避免死锁等问题，文章以DB2(版本9)与Java为例进行讲解。什么是数据库锁定与死锁 锁定(Locking)发生在当一个事务获得对某一资源的“锁”时，这时，其他的事务就不能更改这个资源了，这种机制的存在是为了保证数据一致性.在设计与数据库交互的程序时，必须处理锁与资源不可用的情况。锁定是个比较复杂的概念，仔细说起来可能又需要一大篇，所以在本文中，只把锁定看作是一个临时事件，这意味着如果一个资源被锁定，它总会在以后某个时间被释放。而死锁发生在当多个进程访问同一数据库时，其中每个进程拥有的锁都是其他进程所需的，由此造成每个进程都无法继续下去。 如何避免锁 我们可利用事务型数据库中的隔离级别机制来避免锁的创建，正确地使用隔离级别可使程序处理更多的并发事件(如允许多个用户访问数据)，还能预防像丢失修改 (Lost Update)、读“脏”数据(Dirty Read)、不可重复读(Nonrepeatable Read)及“虚”(Phantom)等问题。 隔离级别 问题现象 丢失修改 读“脏”数据 不可重复读 虚 可重复读 读取稳定性 光标稳定性 未提交的读

隔离级别	丢失修改	读“脏”数据	不可重复读	虚	可重复读	读取稳定性	光标稳定性	未提交的读
只读模式	No	No	No	No	Yes	Yes	Yes	Yes

表1：DB2的隔离级别与其对应的问题现象 在只读模式中，就可以防止锁定发生，而不用那些未提交只读

隔离级别的含糊语句。一条SQL语句当使用了下列命令之一时，就应该考虑只读模式了：1、JOIN 2、SELECT DISTINCT 3、GROUP BY 4、ORDER BY 5、UNION 6、UNION ALL 7、SELECT 8、FOR FETCH ONLY (FOR READ ONLY) 9、SELECT FROM 如果包含上述任一命令，可以说你的SQL语句有歧义性，因此，锁可能就是造成其中资源问题的源头。

另外，以下是一些可降低锁数目的建议：1、将CURRENTDATA设为NO。这条命令告诉DB2模糊光标为只读。2、在适当的时候，尽可能使用User Uncommitted Read(用户未提交的读)。3、尽可能关闭所有光标。4、有一个正确的提交策略。确保程序不再使用资源时就立即释放它。如何处理死锁与超时 在程序中使用重试逻辑，可处理以下三种SQL错误代码：1、904：返回这个代码表示一条SQL语句是因为已达到资源限度而结束的。程序中可提交或回滚更改，并执行重试逻辑。2、911：程序收到这个SQL代码，表示因为没有为锁列表分配足够的内存，现在已达到数据库的最大锁数目。3、912：程序收到这个SQL代码，表示死锁或超时，依照904中的方法来解决。以下是一段Java代码，其捕捉返回的-911、-912、-904代码，并进行重试：

```
for (int i = 0; i < MAX_RETRY_ATTEMPTS; i++) {
    try {
        //等待RETRY_WAIT_TIME
        Thread.sleep(RETRY_WAIT_TIME);
    } catch (InterruptedException e1) {
        //即使休眠被打断，但仍要重试。
        System.out.println("休眠被打断。" + e1);
    }
    //如果是其他错误，就不进行重试。
    System.out.println("有错误发生，错误代码：" + e1.getErrorCode()
        + " SQL状态：" + e1.getSQLState() + "其他信息：" + e1.getMessage());
}
```

上面也可看到，程序对死锁、超时、最大锁数目将会进行MAX_RETRY_ATTEMPTS次重试.其次，当“最大锁数目”的情况发生时 (-911)，程序不必手工进行回滚，因为此时的回滚是自动完成的.最后，无论何时返回-911、-904、-912代码，程序应在下次重试前等待RETRY_WAIT_TIME一段时间。更多优质资料尽在百考试题论坛 百考试题在线题库 java认证更多详细资料 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