非阻塞算法思想在数据库开发中的应用Oracle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_ti2020/558/2021_2022__E9_9D_9E_E9_98_BB_E5_A1_9E_E7_c102_558198.htm 阻塞算法介绍 目前，很多关于并发算法的研究都聚集在非阻塞算法（nonblocking algorithms）上，这种算法使用低层原子化的机器指令取代锁，比如compare-and-swap，从而保证数据在兵法访问下的一致性。非阻塞算法广泛应用于操作系统和JVM的线程和进程调度、垃圾回收以及实现所和其他的并发数据结构。 与基于锁的方案相比，非阻塞算法的设计和实现都要复杂一些，但是它们在可伸缩性和活跃度上占有很大的优势。因为非阻塞算法可以让多个线程在竞争相同资源时不会发生阻塞，所以它能在更精化的层面上调整粒度，并能大大减少开销。进一步而言，它们对死锁和其他活跃度问题具有免疫性。基于锁的算法中，如果一个线程在持有锁的时候休眠，或者停滞不前，那么其它线程就都不能前进了，而非阻塞算法不会受到单个线程失败的影响。在Java 5.0中，使用atomic variable classes，比如AtomicInteger和AtomicReference，能够高效构建非阻赛算法。 非阻塞算法的关键思想就是CAS，CAS是compare and set的缩写，也常被称为lock-free或者wait-free，通过把compare和set两个操作原子化，使得不需要使用锁，但是能够解决并发中的资源争用问题。由于CAS常常是一个回退算法 外循环，所以又被称为spin-lock.由于CAS没有使用锁，线程持续执行，又称为非阻塞算法（non-blocking）。术语不统一，有细微差别，但都差不多表示同一个东西，我都列在这里，方便大家理解。 非阻塞算法的实现通常包括如下部

分：外循环、回退、CAS操作。伪码如下：WHILE (TRUE) // 外循环 { 准备数据 IF CAS_OP() == SUCCESS THEN BREAK. END IF } 非阻塞算法思想在关系数据库开发中的应用 有人说，非阻塞算法这种技术底层框架提供，不需要了解，其实不然，CAS思想可以应用任何地方，包括数据结构、服务接口、数据库应用等等。我这篇文章要讲的内容就是在关系数据库应用中使用CAS思想。 关系数据库数据库提供了"0update T set FState = xx where FState = xx"，执行这样的SQL，会返回一个更新行数，在jdbc或者odbc或者ADO .NET中都可以获得更新行数。上面的SQL，如果更新行数&gt.0，则是更新成功，否则是没有进行任何更新，这是很典型的CAS.可以说，关系数据库 原生支持CAS. 关系数据库中采用事务来确保并发时的原子性，事务实际上就是一种“ 锁 ”。关系数据库中通常有排他锁和共享锁的概念，这有点类似于Java中ReadWriteLock. 需要更新数据时，我们通常使用到关系数据库的排他锁，在Oracle中需要使用SELECT … FOR UPDATE，在Microsoft SQL Server中，使用lock hints. 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