Oracle并发处理机制的简单看法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_ti2020/587/2021_2022_Oracle_E5_B9_B6_E5_c102_587116.htm 在Oracle开发过程中，如果你只是独立地测试你的应用，然后部署，并交给数十个并发用户使用，就很有可能痛苦地遭遇原先未能检测到的并发问题。例如，2个用户同时修改某张订单，首先他们会查询这张订单存在不存在，如果存在，那么修改它的状态。在并发操作中，用户1会很奇怪的发现他的修改丢失了。当然，除此之外，在未能够很好的处理并发问题可能遭遇的情况还有：◆破坏数据的完整性。◆随着用户数的增多，应用的运行速度减慢。◆不能很好地扩缩应用来支持大量用户。为解决这些问题。首先要引入的是ORACLE的锁机制。数据库使用锁（lock）来保证任何给定时刻最多只有一个事务在修改给定的一段数据。实质上讲，正是锁机制才使并发控制成为可能。对ORACLE的锁机制可以查看ORACLE官方文档介绍。以下是对ORACLE锁的一点总结。Oracle只在修改时才对数据加行级锁。正常情况下不会升级到块级锁或表级锁（不过两段提交期间的一段很短的时间内除外，这是一个不常见的操作）。◆如果只是读数据，Oracle绝不会对数据锁定。不会因为简单的读操作在数据行上锁定。◆写入器（writer）不会阻塞读取器（reader）。换种说法：读（read）不会被写（write）阻塞。这一点几乎与其他所有数据库都不一样。在其他数据库中，读往往会被写阻塞。尽管听上去这个特性似乎很不错（一般情况下确实如此），但是，如果你没有充分理解这个思想，而且想通过应用逻辑对应用施加完整性约束，就极

有可能做得不对。◆写入器想写某行数据，但另一个写入器已经锁定了这行数据，此时该写入器才会被阻塞。读取器绝对不会阻塞写入器。 开发人员要尽可能的考虑以上因素。而且还要意识到这些事ORACLE独有的。针对其他数据库，在锁的应用上略有不同。 以DB2为例 1.Oracle通过具有意向锁的多粒度封锁机制进行并发控制，保证数据的一致性。其DML锁（数据锁）分为两个层次（粒度）：即表级和行级。通常的 DML操作在表级获得的只是意向锁（RS或RX），其真正的封锁粒度还是在行级；DB2也是通过具有意向锁的多粒度封锁机制进行并发控制，保证数据的一致性。 其DML锁（数据锁）分为两个层次（粒度）：即表级和行级。通常的DML操作在表级获得的只是意向锁（IS，SIX或IX），其真正的封锁粒度也是在行级；另外，在Oracle数据库中，单纯地读数据（SELECT）并不加锁，这些都提高了系统的并发程度，Oracle强调的是能够"读"到数据，并且能够快速的进行数据读取。而DB2的锁强调的是"读一致性"，进行读数据（SELECT）时会根据不同的隔离级别（RR，RS，CS）而分别加 S，IS，IS锁，只有在使用UR隔离级别时才不加锁。从而保证不同应用程序和用户读取的数据是一致的。 2. 在支持高并发度的同时，DB2和Oracle对锁的操纵机制有所不同：Oracle利用意向锁及数据行上加锁标志位等设计技巧，减小了Oracle维护行级锁的开销，使其在数据库并发控制方面有着一定的优势。而DB2中对每个锁会在锁的内存（locklist）中申请分配一定字节的内存空间，具体是X锁 64字节内存，S锁32字节内存（注：DB2 V8之前是X锁72字节内存而S锁36字节内存）。 3. Oracle数据库中不存在锁升级，而DB2数据库中当数

据库表中行级锁的使用超过locklist*maxlocks会发生锁升级。
4. 在Oracle中当一个session对表进行insert，0update，0delete时候，另外一个session仍然可以从Orace回滚段或者还原表空间中读取该表的前映象（before image）；而在DB2中当一个session对表进行insert，0update，0delete时候，另外一个session仍然在读取该表数据时候会处于lock wait状态，除非使用UR隔离级别可以读取第一个session的未提交的值；所以Oracle同一时刻不同的session有读不一致的现象，而 DB2在同一时刻所有的session都是"读一致"的。 