解析：如何快速掌握SQLServer的锁机制 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/259/2021_2022__E8_A7_A3_E6_9E_90_EF_BC_9A_E5_c98_259380.htm 各种大型数据库所采用的锁的基本理论是一致的，但在具体实现上各有差别。SQL Server更强调由系统来管理锁。在用户有SQL请求时，系统分析请求，自动在满足锁定条件和系统性能之间为数据库加上适当的锁，同时系统在运行期间常常自动进行优化处理，实行动态加锁。对于一般的用户而言，通过系统的自动锁定管理机制基本可以满足使用要求，但如果对数据安全、数据库完整性和一致性有特殊要求，就需要了解SQL Server的锁机制，掌握数据库锁定方法。 锁是数据库中的一个非常重要的概念，它主要用于多用户环境下保证数据库完整性和一致性。 我们知道，多个用户能够同时操纵同一个数据库中的数据，会发生数据不一致现象。即如果没有锁定且多个用户同时访问一个数据库，则当他们的事务同时使用相同的数据时可能会发生问题。这些问题包括：丢失更新、脏读、不可重复读和幻觉读： 1. 当两个或多个事务选择同一行，然后基于最初选定的值更新该行时，会发生丢失更新问题。每个事务都不知道其它事务的存在。最后的更新将重写由其它事务所做的更新，这将导致数据丢失。例如，两个编辑人员制作了同一文档的电子复本。每个编辑人员独立地更改其复本，然后保存更改后的复本，这样就覆盖了原始文档。最后保存其更改复本的编辑人员覆盖了第一个编辑人员所做的更改。如果在第一个编辑人员完成之后第二个编辑人员才能进行更改，则可以避免该问题。 2. 脏读就是指当一个事务正在访

问数据，并且对数据进行了修改，而这种修改还没有提交到数据库中，这时，另外一个事务也访问这个数据，然后使用了这个数据。因为这个数据是还没有提交的数据，那么另外一个事务读到的这个数据是脏数据，依据脏数据所做的操作可能是不正确的。例如，一个编辑人员正在更改电子文档。在更改过程中，另一个编辑人员复制了该文档（该复本包含到目前为止所做的全部更改）并将其分发给预期的用户。此后，第一个编辑人员认为目前所做的更改是错误的，于是删除了所做的编辑并保存了文档。分发给用户的文档包含不再存在的编辑内容，并且这些编辑内容应认为从未存在过。如果在第一个编辑人员确定最终更改前任何人都不能读取更改的文档，则可以避免该问题。 3．不可重复读是指在一个事务内，多次读同一数据。在这个事务还没有结束时，另外一个事务也访问该同一数据。那么，在第一个事务中的两次读数据之间，由于第二个事务的修改，那么第一个事务两次读到的的数据可能是不一样的。这样就发生了在一个事务内两次读到的数据是不一样的，因此称为是不可重复读。例如，一个编辑人员两次读取同一文档，但在两次读取之间，作者重写了该文档。当编辑人员第二次读取文档时，文档已更改。原始读取不可重复。如果只有在作者全部完成编写后编辑人员才可以读取文档，则可以避免该问题。 4．幻觉读是指当事务不是独立执行时发生的一种现象，例如第一个事务对一个表中的数据进行了修改，这种修改涉及到表中的全部数据行。同时，第二个事务也修改这个表中的数据，这种修改是向表中插入一行新数据。那么，以后就会发生操作第一个事务的用户发现表中还有没有修改的数据行，就好象发生了

幻觉一样。例如，一个编辑人员更改作者提交的文档，但当生产部门将其更改内容合并到该文档的主复本时，发现作者已将未编辑的新材料添加到该文档中。如果在编辑人员和生产部门完成对原始文档的处理之前，任何人都不能将新材料添加到文档中，则可以避免该问题。 所以，处理多用户并发访问的方法是加锁。锁是防止其他事务访问指定的资源控制、实现并发控制的一种主要手段。当一个用户锁住数据库中的某个对象时，其他用户就不能再访问该对象。加锁对并发访问的影响体现在锁的粒度上。为了控制锁定的资源，应该首先了解系统的空间管理。在SQL Server 2000系统中，最小的空间管理单位是页，一个页有8K。所有的数据、日志、索引都存放在页上。另外，使用页有一个限制，这就是表中的一行数据必须在同一个页上，不能跨页。页上面的空间管理单位是盘区，一个盘区是8个连续的页。表和索引的最小占用单位是盘区。数据库是由一个或者多个表或者索引组成，即是由多个盘区组成。放在一个表上的锁限制对整个表的并发访问；放在盘区上的锁限制了对整个盘区的访问；放在数据页上的锁限制了对整个数据页的访问；放在行上的锁只限制对该行的并发访问。 SQL Server 2000 具有多粒度锁定，允许一个事务锁定不同类型的的资源。为了使锁定的成本减至最少，SQL Server 自动将资源锁定在适合任务的级别。锁定在较小的粒度（例如行）可以增加并发但需要较大的开销，因为如果锁定了许多行，则需要控制更多的锁。锁定在较大的粒度（例如表）就并发而言是相当昂贵的，因为锁定整个表限制了其它事务对表中任意部分进行访问，但要求的开销较低，因为需要维护的锁较少。SQL Server 可以锁定行、页、扩展盘

区、表、库等资源。 行是可以锁定的最小空间,行级锁占用的数据资源最少，所以在事务的处理过程中，允许其他事务继续操纵同一个表或者同一个页的其他数据，大大降低了其他事务等待处理的时间，提高了系统的并发性。 页级锁是指在事务的操纵过程中，无论事务处理数据的多少，每一次都锁定一页，在这个页上的数据不能被其他事务操纵。在SQL Server 7.0以前，使用的是页级锁。页级锁锁定的资源比行级锁锁定的数据资源多。在页级锁中，即使是一个事务只操纵页上的一行数据，那么该页上的其他数据行也不能被其他事务使用。因此，当使用页级锁时，会出现数据的浪费现象，也就是说，在同一个页上会出现数据被占用却没有使用的现象。在这种现象中，数据的浪费最多不超过一个页上的数据行。 表级锁也是一个非常重要的锁。表级锁是指事务在操纵某一个表的数据时，锁定了这个数据所在的整个表，其他事务不能访问该表中的其他数据。当事务处理的数据量比较大时，一般使用表级锁。表级锁的特点是使用比较少的系统资源，但是却占用比较多的数据资源。与行级锁和页级锁相比，表级锁占用的系统资源例如内存比较少，但是占用的数据资源却是最大。在表级锁时，有可能出现数据的大量浪费现象，因为表级锁锁定整个表，那么其他的事务都不能操纵表中的其他数据。 盘区锁是一种特殊类型的锁，只能用在一些特殊的情况下。簇级锁就是指事务占用一个盘区，这个盘区不能同时被其他事务占用。例如在创建数据库和创建表时，系统分配物理空间时使用这种类型的锁。系统是按照盘区分配空间的。当系统分配空间时，使用盘区锁，防止其他事务同时使用同一个盘区。当系统完成分配空间之后，就不再使

用这种类型的盘区锁。特别是，当涉及到对数据操作的事务时，不使用盘区锁。 数据库级锁是指锁定整个数据库，防止任何用户或者事务对锁定的数据库进行访问。数据库级锁是一种非常特殊的锁，它只是用于数据库的恢复操作过程中。这种等级的锁是一种最高等级的锁，因为它控制整个数据库的操作。只要对数据库进行恢复操作，那么就需要设置数据库为单用户模式，这样系统就能防止其他用户对该数据库进行各种操作。 行级锁是一种最优锁，因为行级锁不可能出现数据既被占用又没有使用的浪费现象。但是，如果用户事务中频繁对某个表中的多条记录操作，将导致对该表的许多记录行都加上了行级锁，数据库系统中锁的数目会急剧增加，这样就加重了系统负荷，影响系统性能。因此，在SQL Server中，还支持锁升级(lock escalation)。所谓锁升级是指调整锁的粒度，将多个低粒度的锁替换成少数的更高粒度的锁，以此来降低系统负荷。在SQL Server中当一个事务中的锁较多，达到锁升级门限时，系统自动将行级锁和页面锁升级为表级锁。特别值得注意的是，在SQL Server中，锁的升级门限以及锁升级是由系统自动来确定的，不需要用户设置。 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