MySQL的四种事务隔离级别

<https://www.cnblogs.com/wyaokai/p/10921323.html>

一、事务的基本要素（ACID）

　　1、原子性（Atomicity）：事务开始后所有操作，要么全部做完，要么全部不做，不可能停滞在中间环节。事务执行过程中出错，会回滚到事务开始前的状态，所有的操作就像没有发生一样。也就是说事务是一个不可分割的整体，就像化学中学过的原子，是物质构成的基本单位。

　　 2、一致性（Consistency）：事务开始前和结束后，数据库的完整性约束没有被破坏 。比如A向B转账，不可能A扣了钱，B却没收到。

　　 3、隔离性（Isolation）：同一时间，只允许一个事务请求同一数据，不同的事务之间彼此没有任何干扰。比如A正在从一张银行卡中取钱，在A取钱的过程结束前，B不能向这张卡转账。

　　 4、持久性（Durability）：事务完成后，事务对数据库的所有更新将被保存到数据库，不能回滚。

二、事务的并发问题

　　1、脏读：事务A读取了事务B更新的数据，然后B回滚操作，那么A读取到的数据是脏数据

　　2、不可重复读：事务 A 多次读取同一数据，事务 B 在事务A多次读取的过程中，对数据作了更新并提交，导致事务A多次读取同一数据时，结果 不一致。

　　3、幻读：系统管理员A将数据库中所有学生的成绩从具体分数改为ABCDE等级，但是系统管理员B就在这个时候插入了一条具体分数的记录，当系统管理员A改结束后发现还有一条记录没有改过来，就好像发生了幻觉一样，这就叫幻读。

　　小结：不可重复读的和幻读很容易混淆，不可重复读侧重于修改，幻读侧重于新增或删除。解决不可重复读的问题只需锁住满足条件的行，解决幻读需要锁表

三、MySQL事务隔离级别

|  |  |  |  |
| --- | --- | --- | --- |
| 事务隔离级别 | 脏读 | 不可重复读 | 幻读 |
| 读未提交（read-uncommitted） | 是 | 是 | 是 |
| ~~不可重复读~~（read-committed） | 否 | 是 | 是 |
| 可重复读（repeatable-read） | 否 | 否 | 是 |
| 串行化（serializable） | 否 | 否 | 否 |

英 /rɪˈpiːtəbl/ 英 /ˈsɪərɪəlaɪzəbl

补充：

　　1、事务隔离级别为读提交时，写数据只会锁住相应的行

　　2、事务隔离级别为可重复读时，如果检索条件有索引（包括主键索引）的时候，默认加锁方式是next-key 锁；如果检索条件没有索引，更新数据时会锁住整张表。一个间隙被事务加了锁，其他事务是不能在这个间隙插入记录的，这样可以防止幻读。

　　3、事务隔离级别为串行化时，读写数据都会锁住整张表

　　4、隔离级别越高，越能保证数据的完整性和一致性，但是对并发性能的影响也越大。

<https://www.cnblogs.com/nightOfStreet/p/12977291.html>

* 先说持久性：我们知道数据的操作会先在内存中完成，那么事务提交后如何保证一定能持久化到磁盘呢

redo log: 事务在提交前对数据的修改会先写到redo log 中，如果返回事务已提交成功，那么表示redo log已经记录完成。redo log 也有缓冲区，redo log的内存缓冲区大小和磁盘扇区的大小512字节一致，不会出现掉电易失的情况。另外redo log记录的是物理变化，体积很小，且redo log 写磁盘是顺序IO，极快~丝滑

redo log 和binlog区别：一个是用于做持久化，另一个用作数据恢复和复制

* 原子性，指的是被事务包裹的一组操作要么全部成功，要么全部失败。不会存在执行了一部分，另一部分不执行的情况

undo log: MySQL使用undo log实现操作回滚。事务开启后执行的命令都会有一条对应反向的逻辑日志计入undo日志文件中（譬如insert 就会有一条delete）。undo log的持久化会被记录在redo log中（利用redo log 速度快的特性）。一旦发生错误或者回滚的时候，利用undo就可以操作回去

事务的持久性和原子性由Redo log和Undo log实现

隔离性和一致性的权衡由锁机制和MVCC实现

<https://www.cnblogs.com/paulwang92115/p/12189487.html>

1. MVCC 是被 MySql 中 事务型存储引擎 InnoDB所支持的。
2. 应对高并发事务，MVCC 比单纯的加行锁更有效，开销更小。
3. MVCC 只在 READ COMMIT 和 REPEATABLE READ两个隔离级别下工作。
4. MVCC 可以使用乐观和悲观来实现。

分析#

InnoDB 存储引擎在数据库没行数据后面添加了三个字段。

6个字节的事务 ID（DB\_TRX\_ID）字段：标记了最新更新这条行记录的 transaction id，每处理一个事务，其值自动 +1。

7个字节的回滚指针（DB\_ROLL\_PTR）字段：指向当前记录的 rollback segment 的 undo log（撤销日志记录），找之前版本的数据就是通过这个指针。

6个字节的 DB\_ROW\_ID 字段：当由 innodb 自动产生聚集索引时，聚集索引包括这个 DB\_ROW\_ID 的值，否则聚集索引中不包括这个值，这个用于索引当中。