

事务

事务是指数据库中的一组操作，要么都成功，要么都失败。事务的支持是在引擎层都实现的，mysql的原生引擎不支持事务，才逐渐被InnoDB取代。

事务隔离性和隔离级别

事务有如下4个特性

- A: 原子性
- C: 一致性
- I: 隔离性
- D: 持久性

数据库上执行多个事务的时候，会出现脏读，不可重复读，幻读等问题。不同事务之间在什么时候能读到数据，这个就是事务的“隔离级别”。mysql的隔离级别如下：

- 读未提交：一个事务还未提交时，它做的修改能被别的事务看到。
- 读提交：一个事务做的修改只有提交后才能被别的事务看到。
- 可重复读：一个事务执行过程中看到的数据总是跟这个事务启动时看到的数据一致。
- 串行化：加锁来隔离不同事务，读会加读锁，写会加写锁，当不同事务执行过程中遇到读写锁冲突时，后启动的事务必须等先启动的事务执行完成。

以上隔离级别逐步解决了多事务执行时会出现的问题，显而易见，事务隔离级别越高，数据库的效率也越低，实际业务需要在效率和安全之间找一个平衡。

事务A	事务B
启动事务 查询得到值1	启动事务
	查询得到值1
	将1改成2
查询得到值V1	
	提交事务B
查询得到值V2	
提交事务A	
查询得到值V3	

如图所示，分析一下不同隔离级别下事务A中的V1,V2,V3的值。

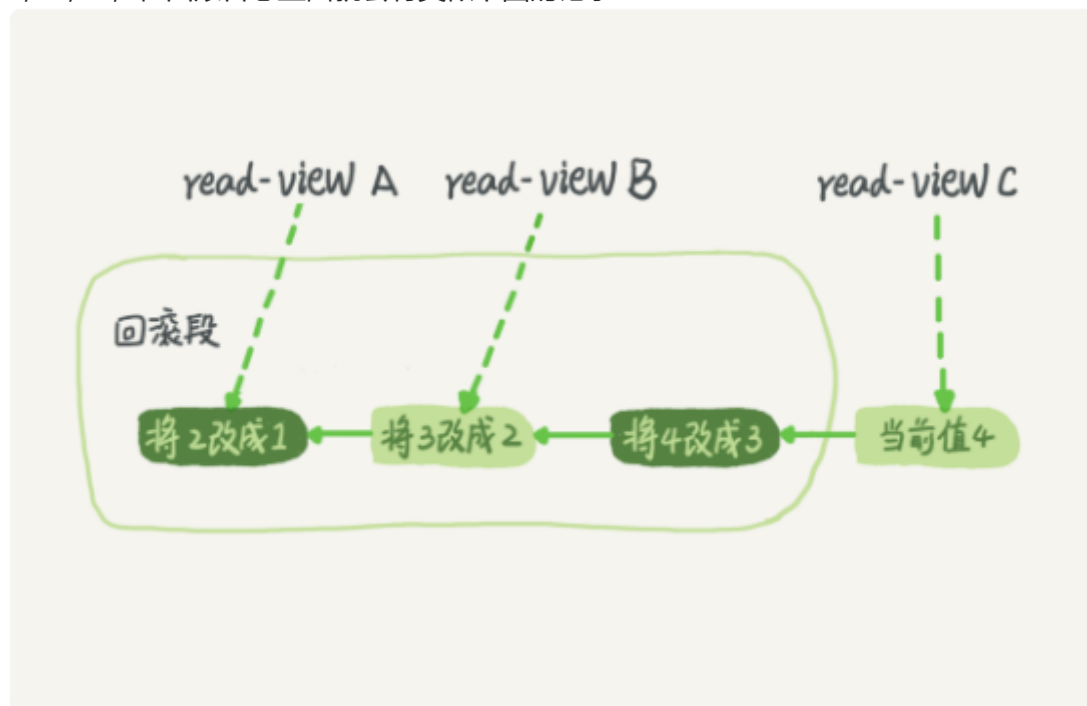
- 读未提交：事务A在事务B未提交时就可以看到事务B所做的修改，所以V1,V2,V3都是2。（假如事务B不提交，回滚了，就会产生脏读）
- 读提交：事务A只能在B提交后才能看到事务B的修改，所以V1是1，V2，V3是2。（在事务A执行期间读到的数据前后是不一致的，就会造成不可重复读。）
- 可重复读：事务A从启动开始读到的数据不会改变直到事务结束，所以V1,V2的值1,V3的值都是2。（假如事务B执行的是insert操作，A事务可能会出现幻读问题。）
- 串行化：事务A启动查询得到的值是1，加上了读锁，事务B需等待事务A提交后才能执行，所以V1,V2的值是1，V3都是2。

数据库是通过视图去处理隔离级别的，读未提交是直接返回记录上的最新值，没有视图；读提交是在每条sql执行时创建的视图；可重复读是在事务开启时创立的视图；串行化是用加锁的方式避免并行访问。

事务隔离的实现

用“可重复度”来具体分析Mysql事务隔离的具体实现。在Mysql中，每条记录在更新的时候都会同时记录一条回滚操作。记录上的最新值都可以通过回滚回到上一个状态。假设一个值从1被改成了

2, 3, 4, 在回滚日志里面就会有类似下图的记录：



如图可以看到，在事务A,B,C启动时mysql的视图记录分别是1，2，4，同一条记录在系统中存在多个版本，这个就是数据库的多版本并发控制(MVCC)。对于视图A来说，要得到1，必须将当前值依次执行图上所有的回滚操作得到。当系统里没有比这个回滚日志更早的read-view时才会删除回滚日志，所以尽量不要使用长事务，会导致回滚日志占据大量存储空间，长事务还会占用锁资源，稍有不慎就会拖垮整个库。