

参考：

MySQL实战45讲

## 前言

提到事务，大家都知道是个重要的话题了，事务一般涉及到ACID，今天我们就要讨论其中的I（Isolation）隔离性。提到隔离性自然要提及隔离级别，有一定基础的应该都知道隔离级别是解决多事务可能出现的脏读，不可重复读，幻读等问题。

## 隔离级别

在讲事务隔离的实现前，我们先介绍下隔离的级别。

读未提交：一个事务未提交前，做的变更已经能被其它事务看见；

读提交：一个事务提交后，做的变更才能被其它事务看见；

可重复读：一个事务执行中看到的数据必须始终保持与启动时看到的一致，当然，未提交前对其它事务是不可见的；

串行化：写加写锁，读加读锁，后一个事务和前一个发生读写锁冲突时必须等前一个提交完。

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	

为了更好的去理解四个隔离级别，以上图两个事务为例，我们看看四个隔离级别的影响。

读未提交：V1=V2=V3=2；

读提交：V1=1，V2=V3=2；

可重复读：V1=V2=1；V3=2；

串行化：V1=V2=1；V3=2；

对于Oracle是默认使用读提交的，我们是建议MySQL也设值默认为读提交的，将启动参数transaction-isolation设值成READ-COMMITTED。

```
mysql> show variables like 'transaction_isolation';
+-----+-----+
| Variable_name | Value               |
+-----+-----+
| transaction_isolation | REPEATABLE-READ |
+-----+-----+
1 row in set, 1 warning (0.02 sec)

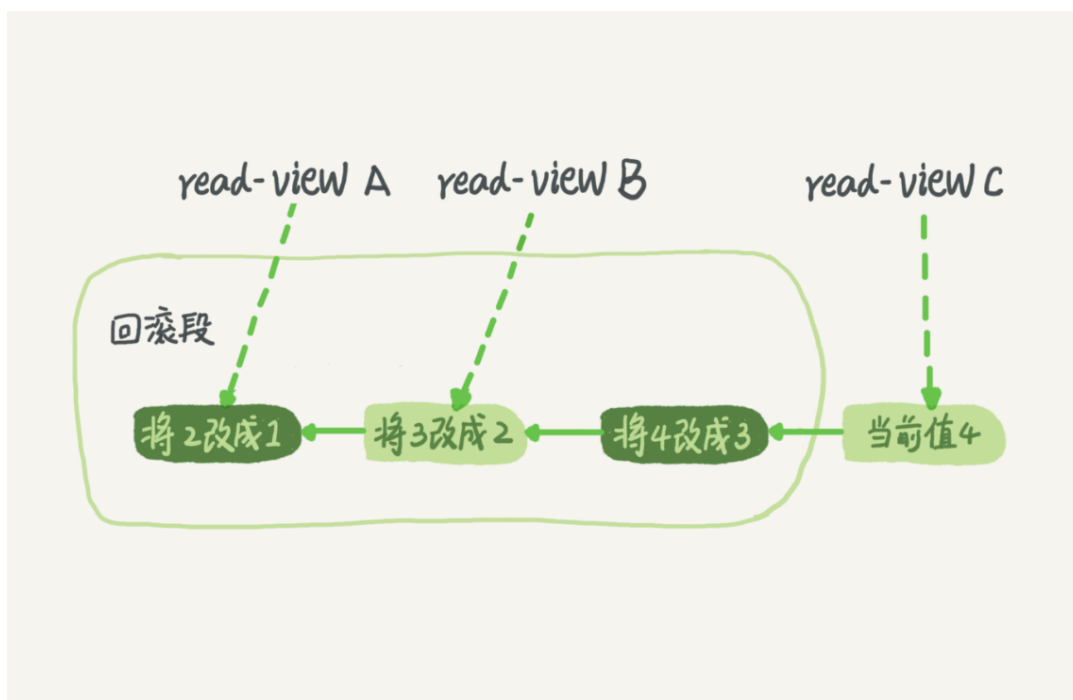
mysql>
```

可以看见MySQL默认是可重复读的，那么就举个可重复读的应用场景吧。一个账务表，校验员开始校验，校验的要求是只校验开始校验前的数据，也就是校验中发生的账务变更不应在本次校验做了，这时候就可以完美运用可重复读了。（提一下，隔离的实现上是通过视图实现的，可重复读可以理解为事务开始就生成了一张静态视图，直到事务结束不会变更）。

## 事务隔离的实现

我们以可重复读来展开叙述这个过程，在MySQL中，每次记录更新时都会同时记录一条回滚记录，因此可以通过回滚，得到前一个状态的值。

假设一个值1被按顺序改成了2，3，4，在回滚日志就会有如下记录。



如图，当前值为4，不同时刻启动的事务产生了A，B，C不同的视图，因此可以认为同一记录在系统中存在了多个版本，这就是数据库的多版本并发控制（MVCC），想要得到视图A的1值，从4依次回滚即可。

当然回滚日志也不可能一直存在，总不能这条记录已经做了亿次级别的变更时第一次的回滚日志还在吧，因此系统会判断，当再没有事务需要用到这些回滚日志时，就会删除。那么，系统如何判断呢，当系统没有比该回滚日志更早的视图时。比如上述的视图A，B已经删除了，那自然相应的回滚日志也可以退休了。

所以这就涉及到一个概念--长事务，这是不建议使用的，为什么呢，事务不结束，视图自然会一直存在，那自然回滚日志也会一直滚雪球，如果过长的话，这占用的空间是无法想象的。（MySQL5.5之前的版本更坑，回滚日志存放ibdata文件，事务提交回滚段被清理，但文件大小不会变）。同时长事务还占用锁资源，严重拖垮整个库。