120 案例实战：千万级数据删除导致的慢查询优化实践（3）

案例实战：千万级数据删除导致的慢查询优化实践（3）

接着119讲的内容往下，我们就得开始要讲解如何为MySQL搭建一套主从复制架构了，会涉及到一些数据库配置的实操步骤

不过在这个之前，其实我们得先解决之前的一个案例的遗留问题，之前在117讲里，我们当时对一个SQL性能优化案例的讲解才到第二讲，最后还留了一个尾巴没解决，就是当时那个慢查询到底是什么原因导致的。

其实说穿了也并不难，这里我们就提一下吧，大家就可以跟之前的117讲衔接上去了。

简单来说，当时经过排查，一直排查到117讲末尾的时候，发现有大量的更新语句在活跃，而且有那种长期活跃的超长事务一直在跑没有结束，结果一问系统负责人，发现他在后台跑了一个定时任务，定时清理数据，结果清理的时候一下子清理了上千万的数据。

这个清理是怎么做的呢？他居然开了一个事务，然后在一个事务里删除上千万数据，导致这个事务一直在运行，所以才看到117讲末尾发现的一些现象。

然后呢，这种长事务的运行会导致一个问题，那就是你删除的时候仅仅只是对数据加了一个删除标记，事实上并没有彻底删除掉。此时你如果跟长事务同时运行的其他事务里在查询，他在查询的时候是可能会把那上千万被标记为删除的数据都扫描一遍的。

因为每次扫描到一批数据，都发现标记为删除了，接着就会再继续往下扫描，所以才导致一些查询语句会那么的慢。

那么可能有人会问了，为什么你启动一个事务，在事务里查询，凭什么就要去扫描之前那个长事务标记为删除状态的上千万的垃圾数据呢？按说那些数据都被删除了，跟你没关系了，你可以不用去扫描他们啊！

这个问题的关键点就在于，那个删除千万级数据的事务是个长事务！

也就是说，当你启动新事务查询的时候，那个删除千万级数据的长事务一直在运行，是活跃的！所以大家还记得我们之前讲解MVCC的时候，提到的一个Read View的概念么？MVCC是如何实现的？不就是基于一个Read View机制来实现的么？

当你启动一个新事务查询的时候，会生成一个Read View，里面包含了当前活跃事务的最大id、最小id和事务id集合，然后他有一个判定规则，具体判定规则大家不记得可以回顾一下当时我们讲过的内容。

总之就是，你的新事务查询的时候，会根据ReadView去判断哪些数据是你可见的，以及你可见的数据版本是哪个版本，因为一个数据有一个版本链条，有的时候你可能可见的仅仅是这个数据的一个历史版本而已。

所以正是因为这个长事务一直在运行还在删除大量的数据，而且这些数据仅仅是标记为删除，实际还没删除，所以此时你新开事务的查询是会读到所有被标记为删除的数据的，就会出现千万级的数据扫描，才会造成慢查询！

针对这个问题，其实大家要知道的一点是，永远不要在业务高峰期去运行那种删除大量数据的语句，因为这可能导致一些正常的SQL都变慢查询，因为那些SQL也许会不断扫描你标记为删除的大量数据，好不容易扫描到一批数据，结果发现是标记为删除的，于是继续扫描下去，导致了慢查询！

所以当时的解决方案也很简单，直接kill那个正在删除千万级数据的长事务，所有SQL很快会恢复正常，从此以后，对于大量数据清理全部放在凌晨去执行，那个时候就没什么人使用系统了，所以查询也很少。

End