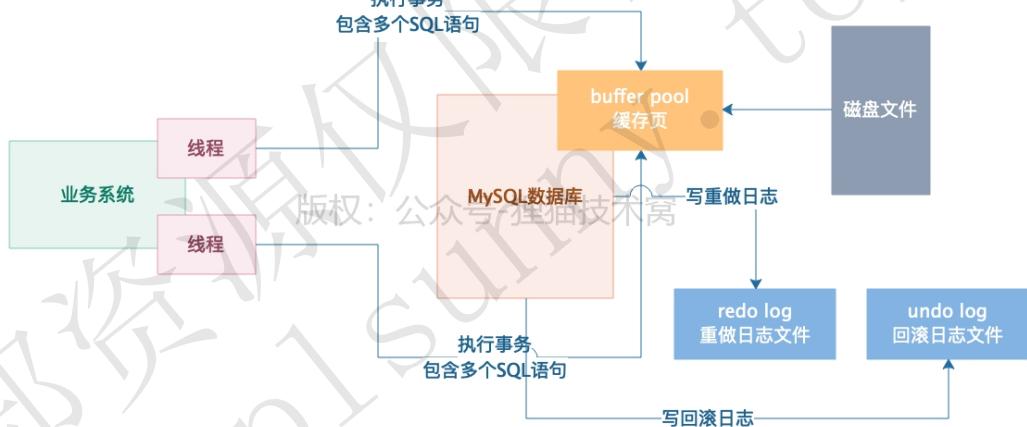


- **如何提问：**每篇文章都有评论区，大家在评论区留言提问
- **如何加入狸猫技术交流群：**
 - 添加微信号：Lvgu0715_（微信名：绿小九），狸猫技术窝的管理员
 - 发送专栏购买截图
 - 2小时内管理员会拉群，人工操作请耐心等待

上一次我们已经讲到，其实对于我们的业务系统去访问数据库而言，他往往都是多个线程并发执行多个事务的，对于数据库而言，他会有多个事务同时执行，可能这多个事务还会同时更新和查询同一条数据，所以这里会有一些问题需要数据库来解决，如下图。



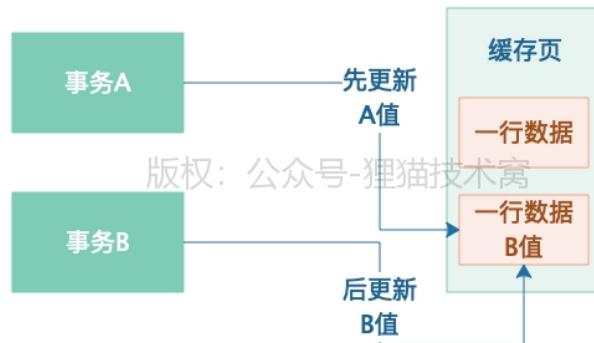
每个事务都会执行各种增删改查的语句，把磁盘上的数据页加载到buffer pool的缓存页里来，然后更新缓存页，记录redo log和undo log，最终提交事务或者是回滚事务，多个事务会并发干上述一系列事情。

所以今天我们要来看看，如果多个事务要是对缓存页里的同一条数据同时进行更新或者查询，此时会产生哪些问题呢？

这里实际上会涉及到**脏写、脏读、不可重复读、幻读**，四种问题。

先看第一种问题，**脏写**

这个脏写的话，他的意思就是说有两个事务，事务A和事务B同时在更新一条数据，事务A先把他更新为A值，事务B紧接着就把他更新为B值，如下图所示



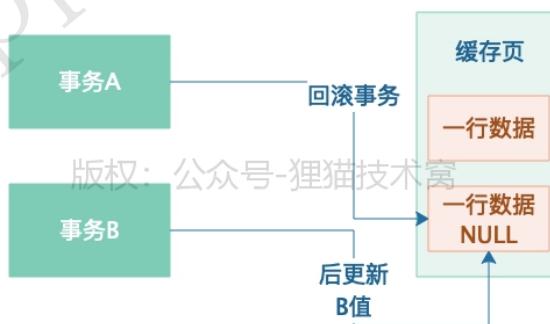
大家可以看到，此时事务B是后更新那行数据的值，所以此时那行数据的值是不是B值？

没错的。而且此时事务A更新之后会记录一条undo log日志，大家应该还记得吧。事务A是先更新的，他在更新之前，这行数据的值为NULL，对吧？

所以此时事务A的undo log日志大概就是：更新之前这行数据的值为NULL，主键为XX

好，那么此时事务B更新完了数据的值为B，结果此时事务A突然回滚了，那么就会用他的undo log日志去回滚。

此时事务A一回滚，直接就会把那行数据的值更新回之前的NULL值！所以此时事务A回滚了，可能看起来这行数据的值就是NULL了，如下图。



然后就尴尬了，事务B一看，我的妈呀，为什么我更新的B值没了？就因为你事务A反悔了就把数据值回滚成NULL了，搞的我更新的B值也没了，这也太坑爹了吧！

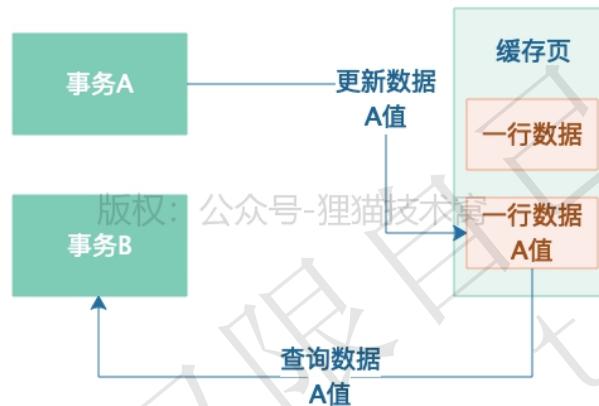
所以对于事务B看到的场景，就是自己明明更新了，结果值却没了，**这就是脏写！**

所谓脏写，就是我刚才明明写了一个数据值，结果过了一会儿却没了！真是莫名其妙。

而他的本质就是事务B去修改了事务A修改过的值，但是此时事务A还没提交，所以事务A随时会回滚，导致事务B修改的值也没了，这就是脏写的定义。

接着我们继续看坑爹的脏读问题

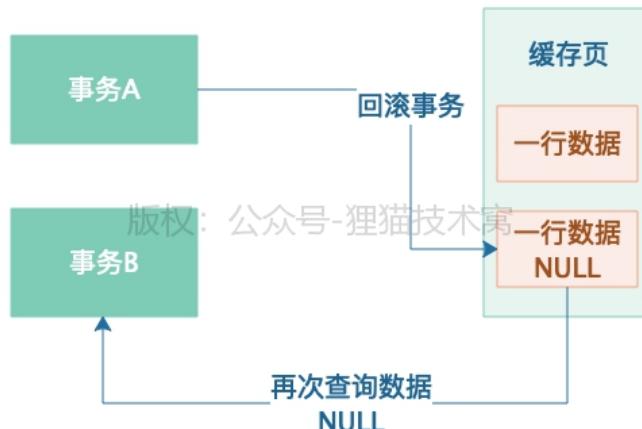
假设事务A更新了一行数据的值为A值，此时事务B去查询了一下这行数据的值，看到的值是不是A值？没错，此时如下图所示。



好，现在事务B可能还挺high的，拿着刚才查询到的A值做各种业务处理。大家知道，每个事务都是业务系统发出的，所以业务系统里的事务B此时肯定会拿到刚查出来的A值在做一些业务处理。

但是接着坑爹的事情发生了，事务A突然回滚了事务，导致他刚才更新的A值没了，此时那行数据的值回滚为NULL值！

然后事务B紧接着此时再次查询那行数据的值，看到的居然此时是NULL值？事务B此时简直欲哭无泪，看下图：



所以这就是坑爹的脏读，他的本质其实就是事务B去查询了事务A修改过的数据，但是此时事务A还没提交，所以事务A随时会回滚导致事务B再次查询就读不到刚才事务A修改的数据了！这就是脏读。

好了，今天我们先初步看一下脏写和脏读两种问题，相信大家有了之前的大量MySQL底层知识原理的铺垫，理解这两个问题，肯定是小case，轻松+愉快的

其实一句话总结，**无论是脏写还是脏读，都是因为一个事务去更新或者查询了另外一个还没提交的事务更新过的数据。**

因为另外一个事务还没提交，所以他随时可能会反悔会回滚，那么必然导致你更新的数据就没了，或者你之前查询到的数据就没了，这就是脏写和脏读两种坑爹场景。

End