

49 一个事务多次查询一条数据读到的都是不同的值，这就是不可重复读？

上一讲我们说完了多个事务并发执行时候，对MySQL的缓存页里的同一行数据同时进行更新或者查询的时候，可能发生的脏写和脏读的问题

我们也理解了，之所以会发生脏写和脏读，最关键的是因为你一个事务写或者查的是人家事务还没提交的时候更新过的数据，所以人家事务随时会反悔回滚，导致你这里有问题。

那么今天我们继续看多个事务并发执行时候的另外两种问题：**一个是不可重复读，一个是幻读**

这两种问题都会奇葩一些，大家仔细看下面的图演示。

先来说说这个不可重复读的问题，这个问题是这样的：假设我们有一个事务A开启了，在这个事务A里会多次对一条数据进行查询

然后呢，另外有两个事务，一个是事务B，一个是事务C，他们俩都是对一条数据进行更新的。

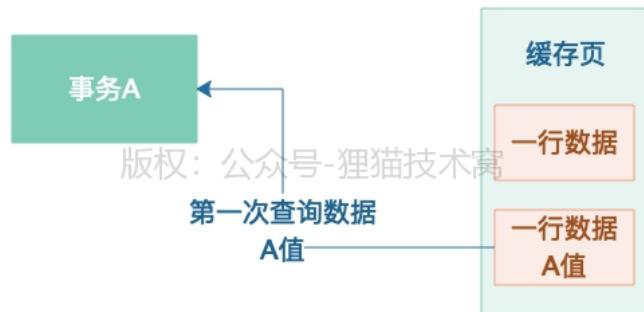
然后我们假设一个前提，就是比如说事务B更新数据之后，如果还没提交，那么事务A是读不到的，必须要事务B提交之后，他修改的值才能被事务A给读取到，其实这种情况下，就是我们首先避免了脏读的发生。

因为脏读的意思就是事务A可以读到事务B修改过还没提交的数据，此时事务B一旦回滚，事务A再次就读不到了，那么此时就会发生脏读问题。

我们现在假设的前提是事务A只能在事务B提交之后读取到他修改的数据，所以此时必然是不会发生脏读的

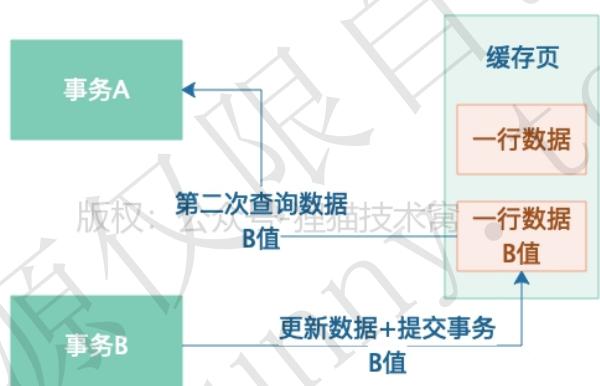
好了，但是你以为没有脏读就万事大吉了吗？绝对不是，此时会有另外一个问题，叫做**不可重复读**

假设缓存页里一条数据原来的值是A值，此时事务A开启之后，第一次查询这条数据，读取到的就是A值，如下图所示。

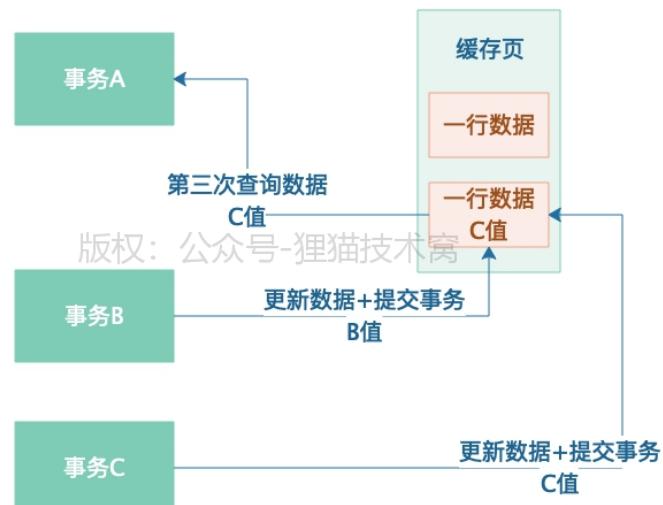


接着事务B更新了那行数据的值为B值，同时事务B立马提交了，然后事务A此时可是还没提交！

大家注意，此时事务A是没提交的，他在事务执行期间第二次查询数据，此时查到的是事务B修改过的值，B值，因为事务B已经提交了，所以事务A可以读到的了，此时如下图所示。



紧接着事务C再次更新数据为C值，并且提交事务了，此时事务A在没提交的情况下，第三次查询数据，查到的值为C值，如下图所示。



好，那么上面的场景有什么问题呢？

其实要说没问题也可以是没问题，毕竟事务B和事务C都提交之后，事务A多次查询查到他们修改的值，是ok的。

但是你要说有问题，也可以是有问题的，就是事务A可能第一次查询到的是A值，那么他可能希望的是在事务执行期间，如果多次查询数据，都是同样的一个A值，他希望这个A值是他重复读取的时候一直可以读到的！他希望这行数据的值是可重复读的！

但是此时，明显A值不是可重复读的，因为事务B和事务C一旦更新了值并且提交了，事务A会读到别的值，所以此时这行数据的值是不可重复读的！此时对于你来说，这个不可重复读的场景，就是一种问题了！

不知道大家看到这里理解了没？如果没理解，反复把这个例子看几遍，理解一下！

上面描述的，其实就是不可重复读的问题，其实这个问题你说是问题是也不一定就是什么大问题，但是说他有问题，确实是有问题的。

因为这取决于你自己想要数据库是什么样子的，如果你希望看到的场景就是不可重复读，也就是事务A在执行期间多次查询一条数据，每次都可以查到其他已经提交的事务修改过的值，那么就是不可重复读的，如果你希望这样子，那也没问题。

但是如果你希望的是，假设你事务A刚开始执行，第一次查询读到的是值A，然后后续你希望事务执行期间，读到的一直都是这个值A，不管其他事务如何更新这个值，哪怕他们都提交了，你就希望你读到的一直是第一次查询到的值A，那么你就是希望可重复读的。

如果你期望的是可重复读，但是数据库表现的是不可重复读，让你事务A执行期间多次查到的值都不一样，都是别的提交过的事务修改过的值，那么此时你就可以认为，数据库有问题，这个问题就是“不可重复读”的问题！

不知道大家听懂这个不可重复读的问题了吗？稍微有点绕口，但是我觉得这篇文章已经解释的很清晰了，请大家反复看2遍，一定会理解可重复读和不可重复读的区别，以及为什么不可重复读会定义为一种数据库的问题呢！

End