

51 SQL标准中对事务的4个隔离级别，都是如何规定的呢？

之前我们给大家讲了数据库中多个事务并发时可能产生的几种问题，包括了脏写、脏读、不可重复读、幻读，几种问题

那么针对这些多事务并发的问题，实际上SQL标准中就规定了事务的几种隔离级别，用来解决这些问题。

注意一下，我们今天讲的这个SQL标准的事务隔离级别，并不是MySQL的事务隔离级别，MySQL在具体实现事务隔离级别的时候会有点差别，这个我们下一次再讲，今天先关注SQL标准是如何规定事务隔离级别的。

在SQL标准中规定了4种事务隔离级别，就是说多个事务并发运行的时候，互相是如何隔离的，从而避免一些事务并发问题

这4种级别包括了：**read uncommitted (读未提交)**，**read committed (读已提交)**，**repeatable read (可重复读)**，**serializable (串行化)**

不同的隔离级别是可以避免不同的事务并发问题的，所以大家一定要对这个事务隔离级别有一个深刻的理解。

第一个**read uncommitted**隔离级别，是不允许发生脏写的

也就是说，不可能两个事务在没提交的情况下更新同一行数据的值，但是在这种隔离级别下，可能发生脏读，不可重复读，幻读。

感觉如何？是不是感觉这种隔离级别让你整个人都感觉不好了！因为脏读的话，就是人家事务在没提交情况下修改的值，居然被你读到了，人家随时可能会回滚的！而且你执行期间多次查询一行数据，可能读到的值都不同，因为别的事务随时会修改值再提交，这个值是不可重复读的！幻读更不用说了，肯定会发生。

说实在的，一般来说，是没有人做系统开发的时候，傻到把事务隔离级别设置为读未提交这个级别的。

第二个是**read committed**隔离级别，这个级别下，不会发生脏写和脏读

也就是说，人家事务没提交的情况下修改的值，你是绝对读不到的！但是呢，可能会发生不可重复读和幻读问题，因为一旦人家事务修改了值然后提交了，你事务是会读到的，所以可能你多次读到的值是不同的！

这里教给大家一个稍微有点骚气的简写名词，就是RC，一般如果你在公司里做开发，有个其他团队的兄弟讨论技术方案的时候，跟你来了句，把事务隔离级别设置成RC！这个时候你不要目瞪口呆，知道是读已提交级别就行了。

你只要记住，这个级别在别的事务已经提交之后读到他们修改过的值就可以了，但是别的事务没提交的时候，绝对不会读到人家修改的值。

第三个是REPEATABLE READ隔离级别，就是可重复读级别

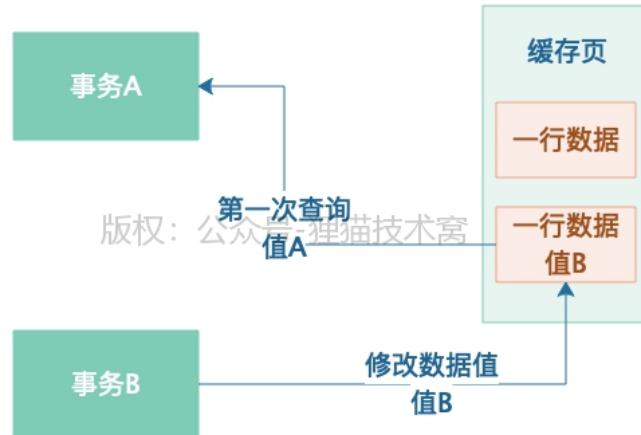
这个级别下，不会发生脏写、脏读和不可重复读的问题，因为你一个事务多次查询一个数据的值，哪怕别的事务修改了这个值还提交了，没用，你不会读到人家提交事务修改过的值，你事务一旦开始，多次查询一个值，会一直读到同一个值！

我们给大家一个图看看这个RR级别的效果，以后记得这个骚气的简写词，就是RR，公司里有兄弟让你把事务设置成RR的时候，你也不要一脸懵逼。

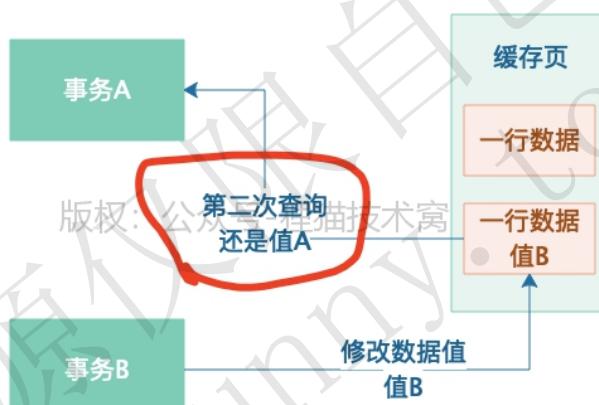
一个事务A，第一次查询一行数据的值是值A，如下图所示。



接着事务B修改了这行数据的值为值B，还提交了，如下图所示。



接着事务A再次查询这行数据的值，读到的还是值A，因为他在事务执行期间，多次读一行数据，绝对读到的都是一样的值，他是允许可重复读的！希望大家理解一下这个概念，可重复读！如下图。



这就是第三种隔离级别，给一个骚气的名字，RR级别，记住了RR级别保证你不会读到人家已经提交的事务修改过的值！但是他还是会幻读的

因为假设你一次SQL是根据条件查询，比如“select * from table where id>10”，第一次查出来10条数据，结果另外一个事务插入了一条数据，下次你可能会查出来11条数据，还是会有幻读问题的！

RR隔离级别，只不过保证对同一行数据的多次查询，你不会读到不一样的值，人家已提交事务修改了这行数据的值，对你也没影响！

最后一个隔离就别，就是serializable级别，这种级别，根本就不允许你多个事务并发执行，只能串行起来执行，先执行事务A提交，然后执行事务B提交，接着执行事务C提交，所以此时你根本不可能有幻读的问题，因为事务压根儿都不并发执行！

但是这种级别一般除非脑子坏了，否则更不可能设置了，因为多个事务串行，那数据库恨不能一秒并发就只有几十了，性能会极差的。

好了，今天就把SQL标准里的四种隔离级别讲完了，**大家一定要记住非常骚气的RC和RR级别**，因为平时比较常见的就是用RC和RR两种隔离级别。

下次我们继续讲MySQL里是如何支持事务隔离级别的，同时在Spring的事务注解里是如何设置事务隔离级别的。

End