

52 MySQL是如何支持4种事务隔离级别的？Spring事务注解是如何设置的？

**MySQL是如何支持4种事务隔离级别的？Spring事务注解是如何设置的？**

---

上次我们讲完了SQL标准下的4种事务隔离级别，平时比较多用的就是RC和RR两种级别，那么在MySQL中也是支持那4种隔离级别的，基本的语义都是差不多的

但是要注意的一点是，MySQL默认设置的事务隔离级别，都是RR级别的，而且MySQL的RR级别是可以避免幻读发生的。

这点是MySQL的RR级别的语义跟SQL标准的RR级别不同的，毕竟SQL标准里规定RR级别是可以发生幻读的，但是MySQL的RR级别避免了！

也就是说，MySQL里执行的事务，默认情况下不会发生脏写、脏读、不可重复读和幻读的问题，事务的执行都是并行的，大家互相不会影响，我不会读到你没提交事务修改的值，即使你修改了值还提交了，我也不可能读到的，即使你插入了一行值还提交了，我也不可能读到的，总之，事务之间互相都完全不影响！

当然，要做到这么神奇和牛叉的效果，MySQL是下了苦功夫的，后续我们接着就要讲解MySQL里的MVCC机制，就是多版本并发控制隔离机制，依托这个MVCC机制，就能让RR级别避免不可重复读和幻读的问题。

然后给大家说一下，假设你要修改MySQL的默认事务隔离级别，是下面的命令，可以设置级别为不同的level，level的值可以是REPEATABLE READ，READ COMMITTED，READ UNCOMMITTED，SERIALIZABLE几种级别。

```
SET [GLOBAL|SESSION] TRANSACTION ISOLATION LEVEL level;
```

但是一般来说，真的其实不用修改这个级别，就用默认的RR其实就特别好，保证你每个事务跑的时候都没人干扰，何乐而不为呢？

另外，给大家说一下，假设你在开发业务系统的时候，比如用Spring里的@Transactional注解来做事务这块，假设某个事务你就是有点手痒痒，就想给弄成RC级别，你就想读到人家已经提交事务修改过的值，好，那么没问题。

在@Transactional注解里是有一个isolation参数的，里面是可以设置事务隔离级别的，具体的设置方式如下：

`@Transactional(isolation=Isolation.DEFAULT)`, 然后默认的就是DEFAULT值, 这个就是MySQL默认支持什么隔离级别就是什么隔离级别。

那MySQL默认是RR级别, 自然你开发的业务系统的事务也都是RR级别的了。

但是你可以手动改成`Isolation.READ_UNCOMMITTED`级别, 此时你就可以读到人家没提交事务修改的值了, 够坑的! 估计一般没人自己坑自己吧!

也可以改成`Isolation.READ_COMMITTED`, `Isolation.REPEATABLE_READ`, `Isolation.SERIALIZABLE`几个级别, 都是可以的。

但是再次提醒, 其实默认的RR隔离机制挺好的, 真的没必要去修改, 除非你一定要在你的事务执行期间多次查询的时候, 必须要查到别的已提交事务修改过的最新值, 那么此时你的业务有这个要求, 你就把Spring的事务注解里的隔离级别设置为`Isolation.READ_COMMITTED`级别, 偶尔可能也是有这种需求的。

好了, 事务的并发问题以及事务隔离级别, 我们迄今为止已经剖析的很透彻了, 接下来就开始讲解MVCC机制, 透彻剖析MySQL是怎么实现牛叉的RR级别的, 怎么就能让事务互相之间彻底隔离开来呢?

**End**