数据库死锁检测对原理

这种情况就是死锁

发生死锁有两种方法解决

1.直接进入等待，直到超时。这个超时时间可以通过参数innodb\_lock\_wait\_timeout来设置

2.发起死锁检测，发现死锁后，主动回滚死锁链条中的某一个事务，让其他事务得以执行。

将参数innodb\_deadlock\_detect设置为on，表示开启这个逻辑

在innodb中，innodb\_lock\_wait\_timeout的值默认是50s，以为着如果使用第一种方法，

第一个被锁住的线程要过50s才会超时退出，然后其他线程才有可能继续执行。

对于在线服务来说，这个等待时间往往是无法接受的。

但是我们又不能把这个时间设置的很小，比如1s，如果这个时候不是死锁，而是正常的锁等待呢

这样就会造成很多误伤,所以我们还是使用死锁检测好一些，好在innodb\_deadlock\_detect默认就是on

其实死锁检测也是会占用很多cpu资源的，当事务被锁住的时候，就要看看它锁依赖的线程有没有被别人锁住

****1.4 回滚机制****

检测到死锁之后，选择插入更新或者删除的行数最少的事务回滚，基于 INFORMATION\_SCHEMA.INNODB\_TRX 表中的 trx\_weight 字段来判断。如果插入更新或者删除的行数一样则回滚后面执行的那条事务。