图文 59 对MySQL锁机制再深入一步，共享锁和独占锁到底是什么？

754 人次阅读 2020-04-24 07:00:00

详情 评论

对 MySQL 锁机制再深入一步，共享锁和独占锁到底是什么?

- **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

今天我们来稍微深入的讲一下MySQL里的共享锁和独占锁这两个概念，上次我们都讲过了，其实多个事务同时更新一行数据，此时都会加锁，然后都会排队等待，必须一个事务执行完毕了，提交了，释放了锁，才能唤醒别的事务继续执行。

那么在这多个事务运行的时候，他们加的是什么锁呢?

其实是X锁，也就是Exclude独占锁，当有一个事务加了独占锁之后，此时其他事务再要更新这行数据，都是要加独占锁的，但是只能生成独占锁在后面等待。

那么这个时候我想问大家一个问题，当有人在更新数据的时候，其他的事务可以读取这行数据吗？默认情况下需要加锁吗?

答案是：不用

因为默认情况下，有人在更新数据的时候，然后你要去读取这行数据，直接默认就是开启mvcc机制的。

也就是说，此时对一行数据的读和写两个操作默认是不会加锁互斥的，因为MySQL设计mvcc机制就是为了解决这个问题，避免频繁加锁互斥。

此时你读取数据，完全可以根据你的ReadView，去在undo log版本链条里找一个你能读取的版本，完全不用去顾虑别人在不在更新。

就算你真的等他更新完毕了还提交了，基于mvcc机制你也读不到他更新的值啊！因为ReadView机制是不允许的，所以你默认情况下的读，完全不需要加锁，不需要去care其他事务的更新加锁问题，直接基于mvcc机制读某个快照就可以了。

那么假设万一要是你在执行查询操作的时候，就是想要加锁呢?

那也是ok的，MySQL首先支持一种共享锁，就是S锁，这个共享锁的语法如下：select * from table lock in share mode，你在一个查询语句后面加上lock in share mode，意思就是查询的时候对一行数据加共享锁。

如果此时有别的事务在更新这行数据，已经加了独占锁了，此时你的共享锁能加吗?

当然不行了，共享锁和独占锁是互斥的！此时你这个查询就只能等着了。

那么如果你先加了共享锁，然后别人来更新要加独占锁行吗？当然不行了，此时锁是互斥的，他只能等待。

那么如果你在加共享锁的时候，别人也加共享锁呢？此时是可以的，你们俩都是可以加共享锁的，共享锁和共享锁是不会互斥的。

所以这里可以先看出一个规律，就是更新数据的时候必然加独占锁，独占锁和独占锁是互斥的，此时别人不能更新；但是此时你要查询，默认是不加锁的，走mvcc机制读快照版本，但是你查询是可以手动加共享锁的，共享锁和独占锁是互斥的，但是共享锁和共享锁是不互斥的，如下规律。

锁类型	独占锁	共享锁
独占锁	互斥	互斥
共享锁	互斥	不互斥

不过说实话，一般开发业务系统的时候，其实你查询主动加共享锁，这种情况较为少见，数据库的行锁是实用功能，但是一般不会在数据库层面做复杂的手动加锁操作，反而会用基于redis/zookeeper的分布式锁来控制业务系统的锁逻辑。

另外就是，查询操作还能加互斥锁，他的方法是：select * from table for update。

这个意思就是，我查出来数据以后还要更新，此时我加独占锁了，其他闲杂人等，都不要更新这个数据了。

一旦你查询的时候加了独占锁，此时在你事务提交之前，任何人都不能更新数据了，只能你在本事务里更新数据，等你提交了，别人再更新数据。

这一讲内容，就是给大家讲了默认情况下更新数据的独占锁，默认情况下查询数据的mvcc机制读快照，然后通过查询加共享锁和独占锁的方式，共享锁和独占锁之间的互斥规则，大家都理解了就好。

End

狸猫技术窝精品专栏及课程推荐：

- 《从零开始带你成为消息中间件实战高手》
- 《21天互联网Java进阶面试训练营》（分布式篇）
- 《互联网Java工程师面试突击》（第1季）
- 《互联网Java工程师面试突击》（第3季）
- 《从零开始带你成为JVM实战高手》