

61 表锁和行锁互相之间的关系以及互斥规则是什么呢？

表锁和行锁互相之间的关系以及互斥规则是什么呢？

今天我们接着讲，MySQL里是如何加表锁的。这个MySQL的表锁，其实是极为鸡肋的一个东西，几乎一般很少会用到，表锁分为两种，一种就是表锁，一种是表级的意向锁，我们分别来看看。

首先说表锁，这个表锁，可以用如下语法来加：

LOCK TABLES xxx READ：这是加表级共享锁

LOCK TABLES xxx WRITE：这是加表级独占锁

其实一般来讲，几乎没人会用这两个语法去加表锁，这不是纯属没事儿找事儿么，所以才说表锁特别的鸡肋。

还有就是有另外两个情况会加表级锁。如果有事务在表里执行增删改操作，那在行级会加独占锁，此时其实同时会在表级加一个意向独占锁；如果有事务在表里执行查询操作，那么会在表级加一个意向共享锁。

其实平时我们操作数据库，比较常见的两种表锁，反而是更新和查询操作加的意向独占锁和意向共享锁，但是这个意向独占锁和意向共享锁，大家暂时可以当他是透明的就可以了，因为两种意向锁根本不会互斥。

为啥呢？因为假设有一个事务要在表里更新id=10的一行数据，在表上加了一个意向独占锁，此时另外一个事务要在表里更新id=20的一行数据，也会在表上加一个意向独占锁，你觉得这两把锁应该互斥吗？

明显是不应该互斥的啊，因为他们俩更新的都是表里不同的数据，你让他们俩在表上加的意向独占锁互斥干什么呢？所以意向锁之间是根本不会互斥的。

同理，假设一个事务要更新表里的数据，在表级加了一个意向独占锁，另外一个事务要在表里读取数据，在表级加了一个意向共享锁，此时你觉得表级的意向独占锁和意向共享锁应该互斥吗？

当然不应该了！一个人要更新数据，一个人要读取数据，俩人在表上加的意向锁，凭什么要互斥？没天理啊！

所以说，大家有没有发现一点，这个所谓的表级的意向独占锁和意向共享锁，似乎是跟脱了裤子放屁一样，多此一举？

但是我们接下来就要给大家讲讲，手动加表级共享锁和独占锁，以及更新和查询的时候自动在表级加的意向共享锁和意向独占锁，他们之间反而是有一定的互斥关系，关系如下表所示。

锁类型	独占锁	意向独占锁	共享锁	意向共享锁
独占锁	互斥	互斥	互斥	互斥
意向独占锁	互斥	不互斥	互斥	不互斥
共享锁	互斥	互斥	不互斥	不互斥
意向共享锁	互斥	不互斥	不互斥	不互斥

大家看看上面表格，仔细看一下，上面说的是在表上面手动加的独占锁和共享锁，以及更新数据和查询数据默认自动加的意向独占锁和意向共享锁，他们互相之间的互斥关系，大家一看就明白了。

其实更新数据自动加的表级意向独占锁，会跟你用 `LOCK TABLES xxx WRITE` 手动加的表级独占锁是互斥的，所以说，假设你手动加了表级独占锁，此时任何人都不能执行更新操作了！

或者你用 `LOCK TABLES xxx WRITE` 手动加了表级共享锁，此时任何人也不能执行更新操作了，因为更新就要加意向独占锁，此时是跟你手动加的表级共享锁，是互斥的！

具体其他实例就不举了，大家看看上面的表格就知道了，你如果手动加了表级的共享锁或者独占锁，此时是会阻塞掉其他事务的一些正常的读写操作的，因为跟他们自动加的意向锁都是互斥的。

但是说实话，这一讲也就是给你讲明白这个表级锁如何加的，如何互斥的，其实一般来说，根本就不会手动加表级锁，所以一般来说读写操作自动加的表级意向锁，互相之间绝对不会互斥。

一般来讲，都是对同一行数据的更新操作加的行级独占锁是互斥，跟读操作都是不互斥的，读操作默认都是走mvcc机制读快照版本的！

End