# innodb的锁模式和锁类型

## 锁模式

MySQL InnoDB一共有四种锁：共享锁（读锁，S锁）、排他锁（写锁，X锁）、意向共享锁（IS锁）和意向排他锁（IX锁）。其中共享锁与排他锁属于行级锁，另外两个意向锁属于表级锁。

* **共享锁（S）**：允许一个事务去读一行，阻止其他事务获得相同的数据集的排他锁。就是我读的时候，你可以读，但是不能写。
* **排他锁（X）**：允许获得排他锁的事务更新数据，但是阻止其他事务获得相同数据集的共享锁和排他锁。我写的时候，你不能读也不能写。
* **意向共享锁（IS）**：表示事务准备给数据行加入共享锁，也就是说一个数据行加共享锁前必须先取得该表的IS锁
* **意向排他锁（IX）**：类似上面，表示事务准备给数据行加入排他锁，说明事务在一个数据行加排他锁前必须先取得该表的IX锁。

意向锁是InnoDB自动加的，不需要用户干预。对于insert、update、delete，InnoDB会自动给涉及的数据加排他锁（X）；对于一般的Select语句，InnoDB不会加任何锁，事务可以通过以下语句给显示加共享锁或排他锁。

共享锁：select \* from table\_name where .....lock in share mode

排他锁：select \* from table\_name where .....for update

1.共享锁(s)和排他锁(X)

2.意向锁，intention locks

意向共享锁(IS)和意向排他锁(IX)

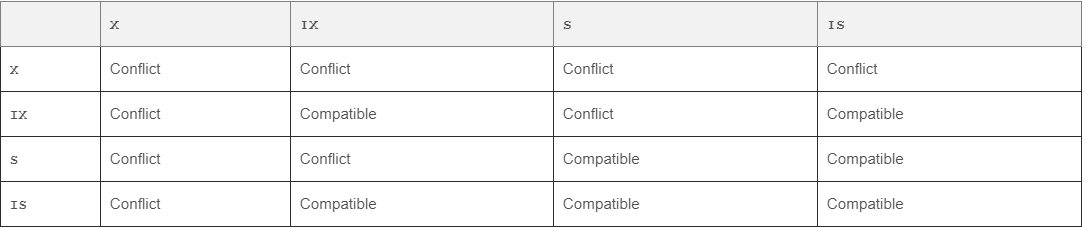
表明了一种意向。

SELECT ... LOCK IN SHARE MODE 加IS锁；

SELECT ... FOR UPDATE 加IX锁。

在一个事务可以得到S锁之前，一定要先获取IS锁或更强级别的锁。

在一个事务可以获得X锁之前，一定要先获得IX锁。



当一个事务请求的锁模式与当前的锁兼容，InnoDB就将请求的锁授予该事务；反之如果请求不兼容，则该事务就等待锁释放。

## 锁类型

innodb的锁是作用于**索引**上的，因此理解innodb的锁，要先理解innode的索引实现B+树。

**1.行级锁 record locks**

行级锁只存在于有索引的行上。

SELECT c1 FROM t WHERE c1 = 10 FOR UPDATE; 锁住C1等于10的索引，此时不能在插入，修改或删除c1=10的行。

**2.区间锁 gap locks**

区间锁是锁住两个索引的区间。

SELECT c1 FROM t WHERE c1 BETWEEN 10 and 20 FOR UPDATE;

加区间锁，锁住这个区间。则C1=15此时插入不进去。

当使用唯一索引则区间锁不存在， SELECT \* FROM child WHERE id = 100;id为主键，则只有行级锁。

值得注意的是，区间锁允许冲突，比如一个事务获取一个区间的S锁的同时另个一事务还可以获取得这个区间的X锁。

区间锁只能阻止其他事务往区间中插入元素，因此X的区间锁和S的区间锁的作用相同。

禁用区间锁:事务的隔离级别设为提交读或开启系统变量 innodb\_locks\_unsafe\_for\_binlog。则区间锁被禁用。只有外键约束检查和重复主键约束检查才可以使用区间锁。

**3.间隙锁next-key locks**

间隙锁是行级锁和区间锁的结合。当一个有索引的行，被间隙锁锁住，则表示这一行被行级锁锁住，前这一行前面的区间被区间锁锁住。

假设一个索引包含10，11，13，20四个值，那么在这个索引的上的next-key锁可以能锁住的区间为以下四个之一。

(negative infinity, 10]

(10,11]

(11,13]

(13,20]

(20,positive infinity)

4.插入意向锁

5.自增锁

自增锁是表锁。如果一个表有一个自增的列，则插入数据的时候，事务需要获得表锁。