

所有当前读加排他锁，都有哪些是当前读？

Select for update

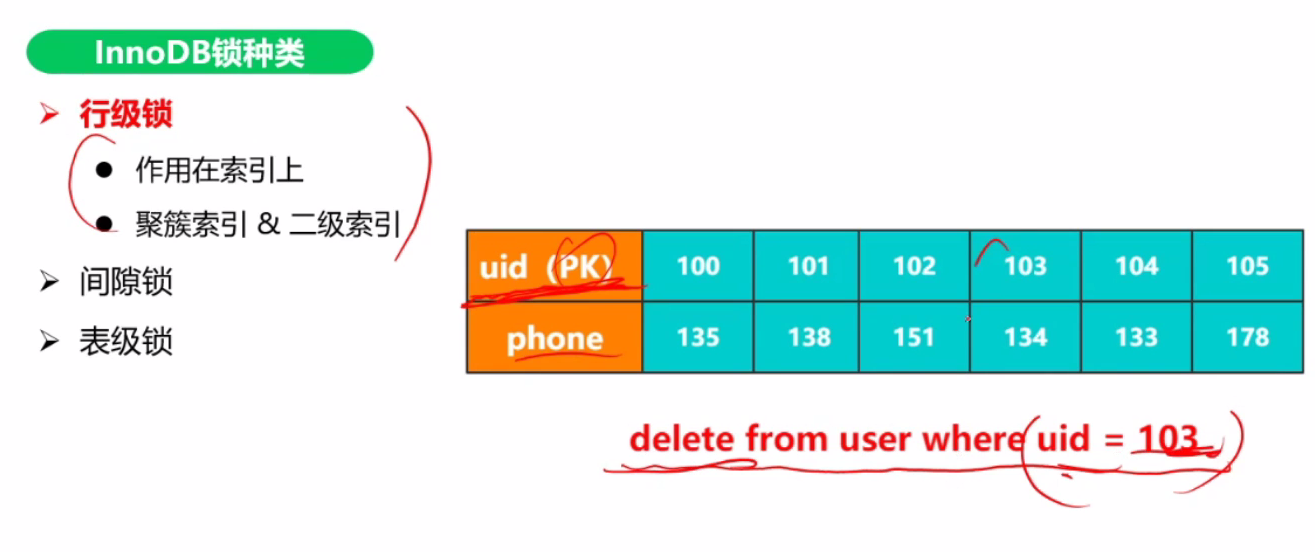
Updata

Delete

排他锁只能被一个事务获取。允许获得锁的事务修改数据

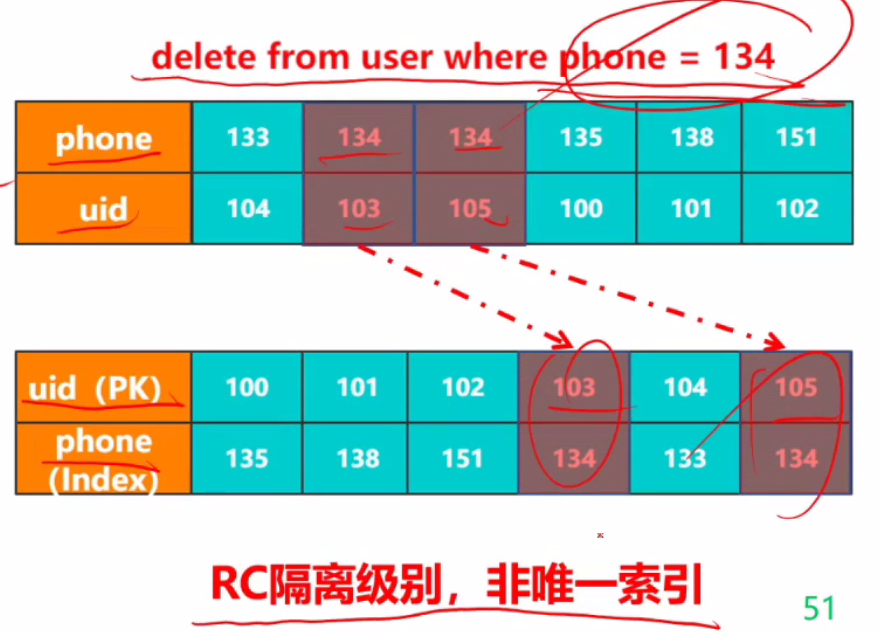
行级锁

红色语句当前读，所以加排他锁，作用在索引上（聚簇索引或者二级索引）而不是在行上.



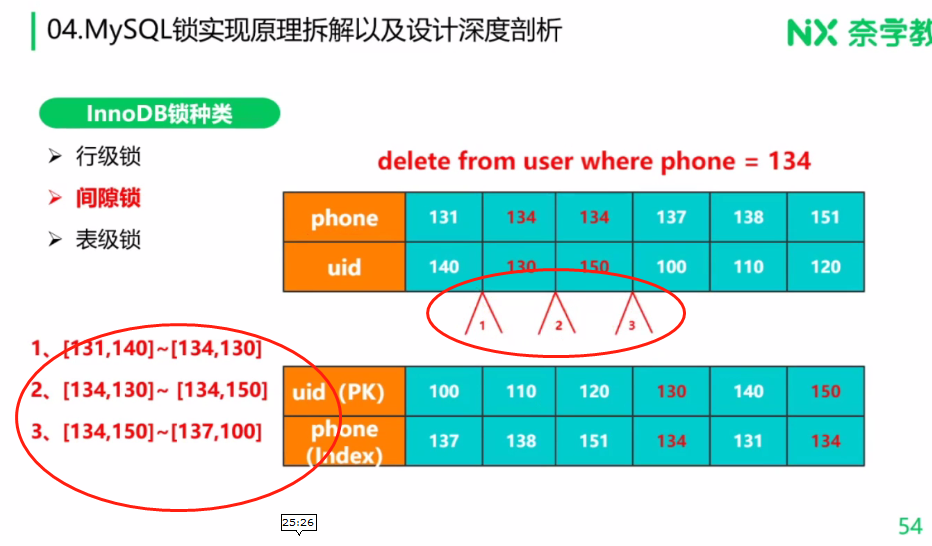
具体情况具体分析

Rc隔离级别，非唯一索引和唯一索引没啥区别：



重点：RR隔离级别非唯一索引的加锁情况, 区别就是引入了gab锁（解决幻读）

Rc 非唯一索引不会加gap锁，所以不会解决幻读问题



上图的引申出的问题：

Insert 语句 phone=134，可以insert，因为他的phone不是唯一索引

再执行delete后，他将返回3条记录，

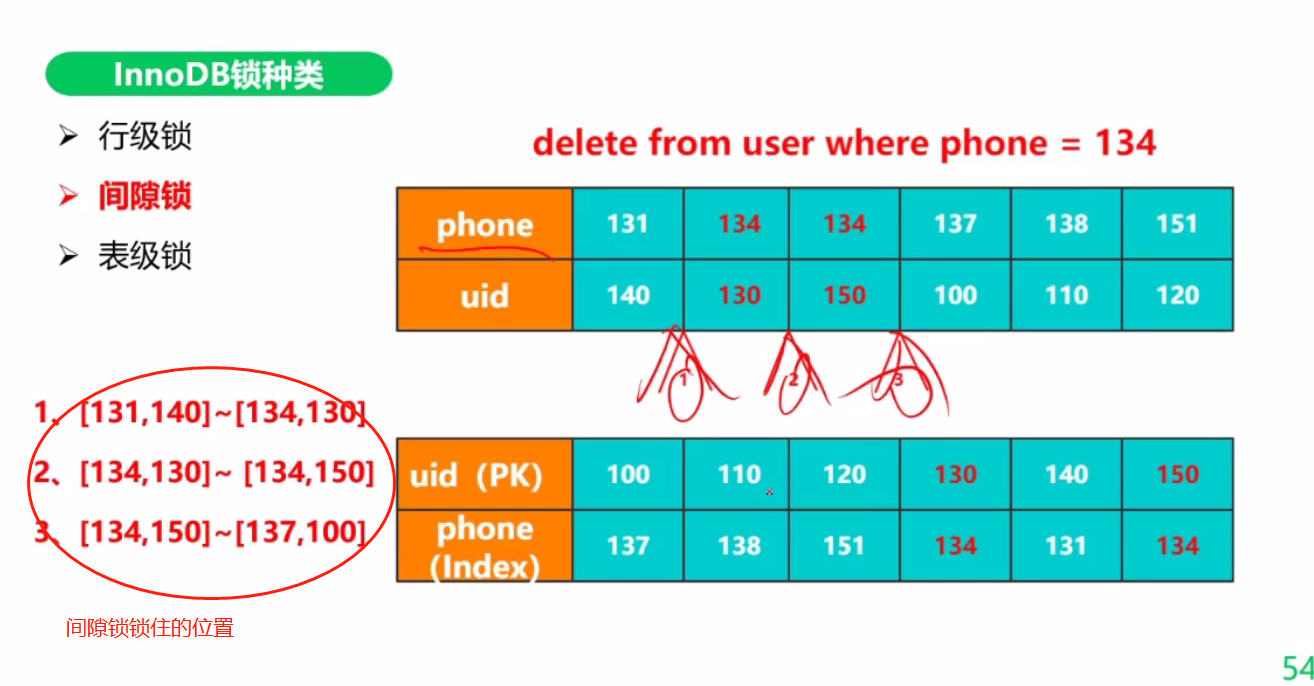
然后再insert语句，

他又返回不同的记录数，**就是每次执行，返回的当前读都不是同样的。**

Gap（间隙）锁的意义（解决幻读）：

**所以，RR级别非唯一索引下，innodb需要加间隙锁**

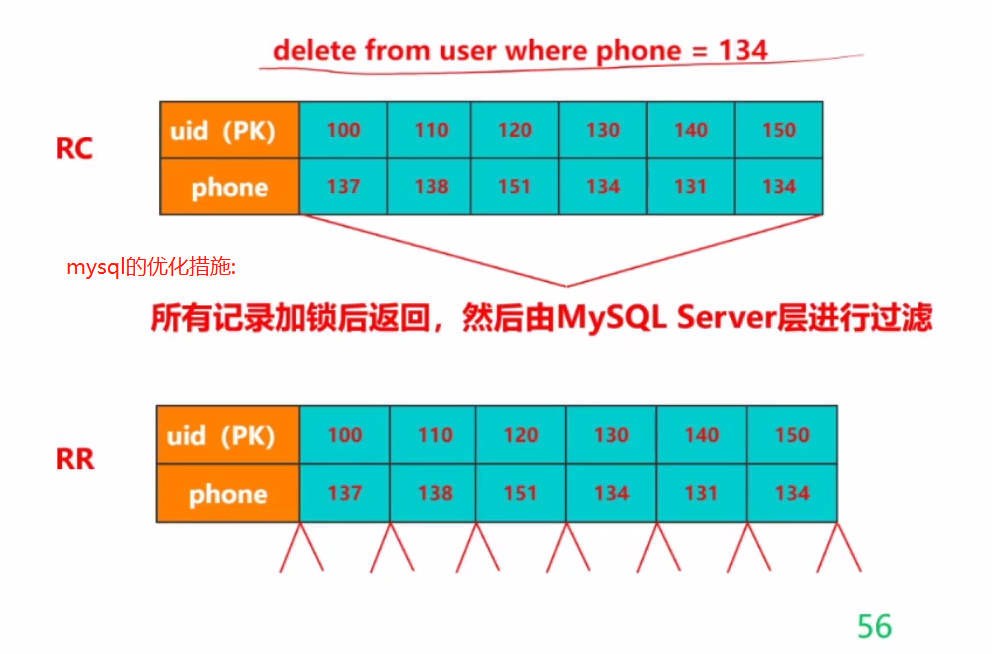
**保证两次当前读返回一致的记录，两次当前读之间，其他的事务不会插入新的满足条件的记录**



表级锁

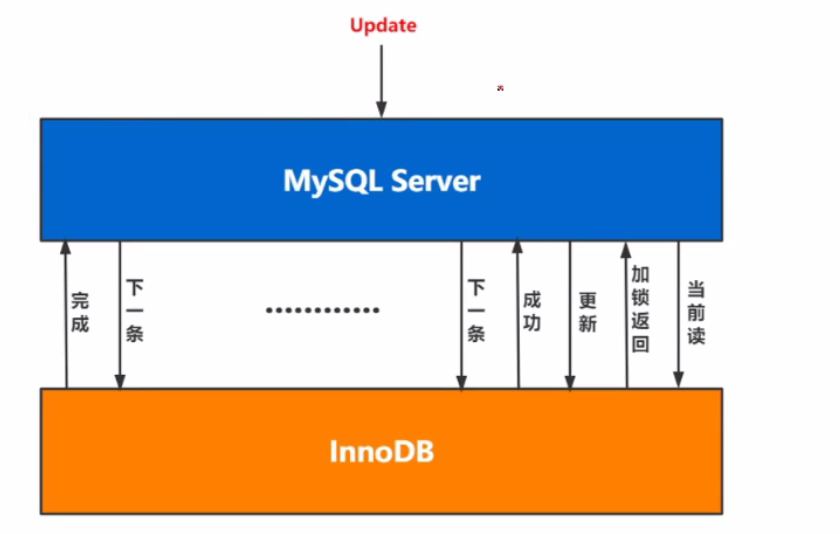
有lock tables（改字段啊，）有元数据锁（meta data lock， MDL）

以下这种情况，全表扫描会导致表锁



尽量避免，虽然有优化（所有记录加锁，但是最终由sever层进行过滤）。

加锁过程



多个事务并发加锁，

事务1 加锁1

随后到事务2加锁2

再然后事务1加了另外的锁3



如下图

第1步，事务t1加了锁，锁住了120记录。

第2步，事务t2加了锁，锁住了130记录。

第3步，事务t1加t2的锁，因为t2已经上锁，所以等待锁释放。

第4步，事务t2加了t1的锁，因为t1已经上锁，所以等待。

所以就产生了死锁的现象