问题起源

Mysql对行锁存在大bug，行锁是加在索引上的。所以如果没有索引，或者意外原因引起索引失效，或者mysql对sql优化器决定不使用索引对情况下，那么就会锁表，导致一个用户更新自己数据，其间所有用户都不能更新自己的数据了。。

其他数据库对索引pgsql mssql oracle等都是把行锁加在数据行上对，这才是行锁的正确方式。。

数据库连接池导致的连接复用对锁表又是雪上加霜，不能及时释放连接，导致表锁持续，很可能造成死锁状态..所有对用户都不能更新自己的数据了。。

# ****二、间隙锁（Next-Key锁）****

　　当用范围条件而不是相等条件检索数据，并请求共享或者排它锁的时候，**InnoDB会给符合条件的已有数据记录的索引项加锁；对于不在范围内的但并不存在的记录，叫做“间隙(GAP)”，InnoDB也会对这个间隙加锁，这就是所谓的间隙锁**。

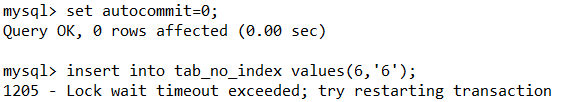
　　如：select \* from where id>100 for update 对id大于100的数据对加锁，但是此时数据中id只有1,2….100,101，不仅对存在的101的记录加锁，还会对大于101不存在的数据的间隙加锁。

　　 此外，对使用相等条件请求给一个不存在的记录加锁，InnoDB也会使用间隙锁，如下：

　　Session\_1：对不存在的id=6的记录加锁

IMG_256

　　Session\_2：插入id=6的记录，也会出现锁等待



# 三、****什么时候使用表锁？****

　　对于InnoDB表，在绝大部分情况下都应该使用行锁，因为事务和行锁往往是我们之所以选择InnoDB表的理由，但在个别情况下也使用表级锁；

　　1）事务需要更新大部分或全部数据，表又比较大，如果使用默认的行锁，不仅这个事务执行效率低，而且可能造成其他事务长时间等待和锁冲突；

　　2）事务涉及多个表，比较复杂，很可能引起死锁，造成大量事务回滚。

# 解决方法汇总

## 实现页锁 对数据分页锁定缩小锁定范围。。会大概率减少锁定

比如分为100页数据，则分别加锁加在100个页上，每个页面内锁定对概率就降低到百分之一类。。

## 实现真正行锁

在MySQL机制下，实现行锁，需要拆分表格数据。。。表锁行锁一体化。有多少用户，就要建立多少表。。虽然mysql没有限制库表数量，但大量表格可能影响些问题。。

## 不使用innodo存储引擎，使用其他存储引擎如pgsql msslq oralce等存储引擎

不过对视图join和存储过程sp会带来些问题，fdw外部表挂载功能或许可以缓解此类问题，要不就要代码实现join