在并发访问时，要安全的修改同一行数据。要安全的修改某行数据，就要保证一个线程在修改时其他线程无法更新这条记录。

在这种情况下可以考虑用悲观锁。悲观锁设定每次修改都会产生冲突。本质就是当前只有一个线程执行操作，结束了再去唤醒其他线程进行处理。
Mysql中的有两种方法：select…for update或lock in share mode。

事物并发性的理解：
事物并发性，粗略的理解就是单位时间内能够执行的事物数量，常见的单位是
TPS（transactions per second）
那在数据量和业务操作量一定的情况下，常见的提高食物并发性主要考虑的有哪几点呢？
1.提高服务器的处理能力，让事务的处理时间变短。
这样不仅加快了这个事务的执行时间，也降低了其他等待改事务执行时间。

2.尽量将事务涉及到的sql操作语句控制在合理范围，换句话说就是不要让一个事务包含的操作太多或者太少
在业务繁忙的情况下，如果单个事务操作的表或者行数据太多，其他的事务可能都在等待该事务commit或者rollback，这样会导致整体上的TPS降低。

3.在操作的时候，尽量控制锁的粒度，能用小的锁粒度就尽量用小的锁粒度，用完锁资源后要记得立即释放，避免后面的事务等待。
但有时候，由于业务的需要，或者为了保证数据一致性的时候，必须要增加锁的粒度。

https://www.cnblogs.com/crazylqy/p/7614245.html（详细看这部分代码）
select * from lockt where col2 = 20 for update，语句会将col2= 20这个索引的入口给锁住，那么事务2虽然看到了所有的数据，但是想去修改col2=20的行数据时候，事务1只能说“不可能也不允许”

总结：这就是select for update的使用场景，为了避免自己看到的数据并不是数据库存储的最新数据并且看到的数据只能由自己修改，需要用for update

select lock in share mode理解

如果看到了前面的select *** for update，就可以很好的理解select lock in share mode， in share mode子句的作用就是将查找到的数据加上一个share锁，这个就是表示其他的事务只能对这些数据进行简单的select操作，并不能够进行DML操作。
那in share mode和for update有什么区别了

lock in share mode 没有for update那么霸道，但是当两个事务都对某行加上了share锁后，
因为事务1和事务2都对该行加上了share锁，事务1以为就只有自己一个人上了s锁，所以当事务一想修改的时候发现没法修改，这种情况下，事务1需要使用for update子句来进行约束，而不是使用for share来使用。