

## 解决超卖

超卖场景：

不同用户在读请求的时候，发现商品库存足够，然后同时发起请求，进行秒杀操作，减库存，导致库存减为负数。

最简单的方法，更新数据库减库存的时候，进行库存限制条件，在reduceStock(GoodsVo goodsvo)这个方法里，sql要多加一个stock\_count > 0即：

```
//stock_count>0的时候才去更新，数据库本身会有锁，那么就不会在数据库中同时多个线程更新一条记录，使用数据库特性来保证超卖的问题
@Update("update miaosha_goods set stock_count=stock_count-1 where goods_id=#{goodsId} and stock_count>0")
public void reduceStock(MiaoshaGoods goods);
```

也可以对读操作加上显式锁（select ... for update）这样一来用户1在进行读操作时用户2就需要排队等待了，但是如果商品很热门并发量很高那么效率就会大大的下降。

1. 在数据库的秒杀订单里加一个唯一的索引,把用户和商品一起建了个唯一索引,唯一索引不可以有重复索引值