

# 优惠券系统设计 Q&A

## 1. 发券接口也有并发问题么？

有的。

## 2. 是不是还有一个case，就是支付成功了，后面用户又取消订单了。

是的，需要修改券状态，核验券是否过期或重置过期时间。

## 3. 请问这个和2 phase commit是不是相似？如果我们用基于MQ的Saga pattern可以么？

TCC和Saga或者Two Phase Commit怎么比较？

是有点相似，但三阶段提交是为了解决两阶段提交协议的缺点而设计的。

分布式事务的四种模式可以参考：<https://zhuanlan.zhihu.com/p/78599954>

## 4. 老师能总结一下这几张表分别存在哪种数据库中么？

可以都用关系型数据库。

数据库的选择可以参考：<https://juejin.cn/post/6844904057925009415>

## 5. 分布式事务在自己学习时候，有没有材料或者主要关注点？

分布式事务第十六章会讲。

参考资料：

1. [https://docs.oracle.com/cd/B19306\\_01/server.102/b14231/ds\\_txns.htm](https://docs.oracle.com/cd/B19306_01/server.102/b14231/ds_txns.htm)

2. <https://tech.ebayinc.com/engineering/grit-a-protocol-for-distributed-transactions-across-microservices/>

分布式事务框架 seata

1. <https://hackernoon.com/fescar-a-distributed-transaction-solution-open-sourced-by-alibaba-f70c9b4c72a1>

2. <https://github.com/seata/seata>

## 6. 老师，如果我们在确认订单之后把优惠券标记为已使用的，这样其他设备也看不到，cancel的时候再标记为未使用，这样有什么坏处？逻辑上冻结状态是更合理一些的。请问技术上，这样做为何不可以。

TCC这部分还是有疑惑，老师并没有回答清楚这个问题。如果在下订单的时候直接“已使用”可能造成多个thread都想access这个优惠券的话，那用TCC的话，也可能多个thread同时尝试冻结呀？所以无论直接“已使用”，还是用tcc冻结，都需要变成线性的访问数据库，才能避免超卖。所以还是不明白为什么需要tcc。

如果要增加实时显示已使用券的需求，如果没有冻结状态，是没有办法正确显示已使用券的数量的，因为用“已使用”表示券“已使用”和“冻结”两种状态。

在下订单的时候就将订单状态改为“已使用”并不会造成多个 thread 都 access 这个优惠券。

#### 7. 还有老师能不能再讲一下，秒杀系统为什么库存要设置为1.5倍？

秒杀系统的 Redis 内库存设置1.5倍主要是为了防止 MQ 部分消息投递失败导致少卖。