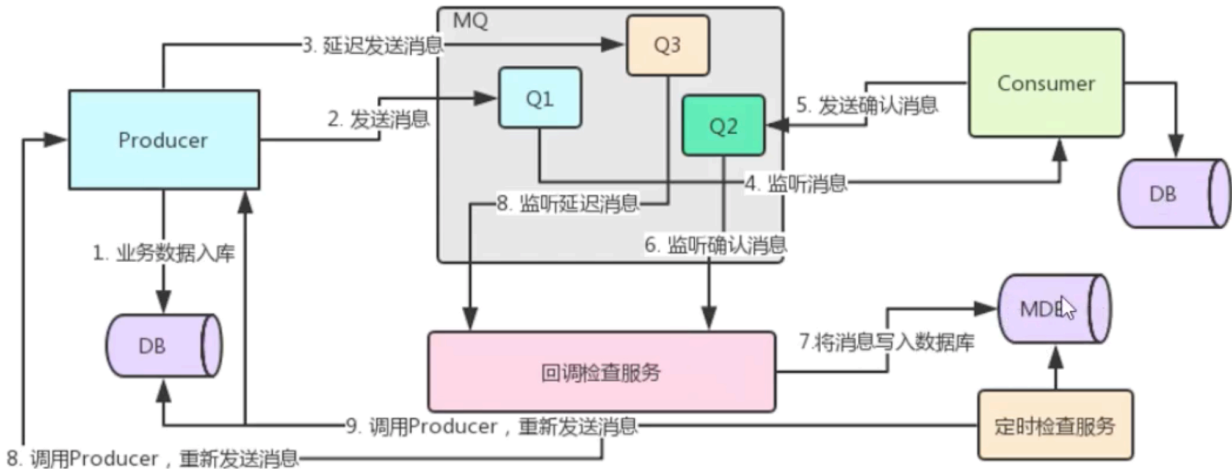


100

## RabbitMQ应用问题

- 3

## 2.1 消息可靠性保障--消息补偿



## 幂等性 处理重复消费的问题

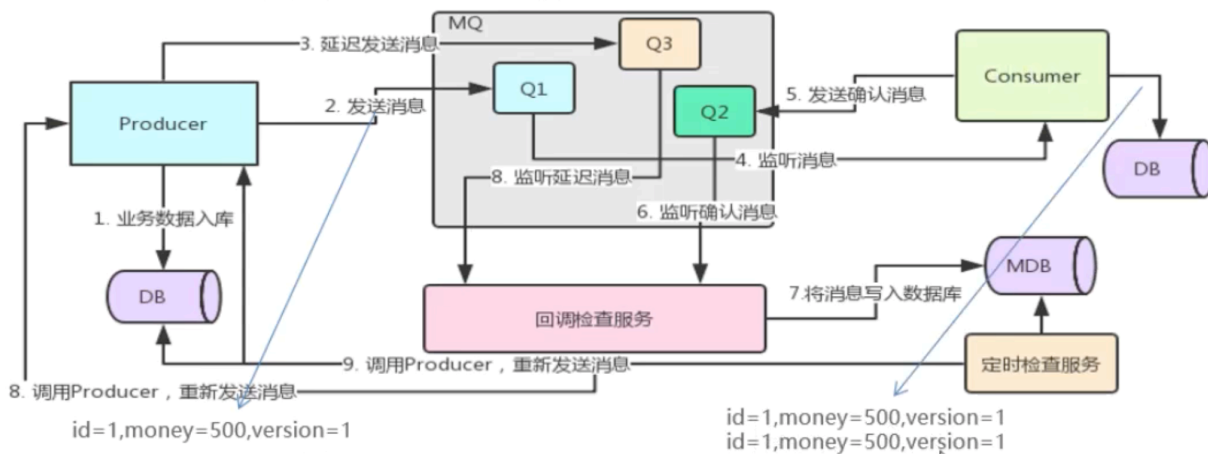
## ■ 2. RabbitMQ 应用问题

### 2.2 消息幂等性保障

幂等性指一次和多次请求某一个资源，对于资源本身应该具有同样的结果。也就是说，其任意多次执行对资源本身所产生的影响均与一次执行的影响相同。

在MQ中指，消费多条相同的消息，得到与消费该消息一次相同的结果。

#### 2.2 消息幂等性保障--乐观锁机制





60元钱小精灵

幂等和消息补偿这里好像有问题：

1. 幂等：数据库里乐观锁是多线程更新同一个值时用的，那个是没问题

但课件中说producer 在消息中直接提供version，这就很奇怪

producer怎么可能自己定义money字段的version？除非producer还得从consumer的DB中查询一遍，这不很奇怪吗

除非是在课件的示意图之外，还有一张独立的money表，那么producer每次发消息前先从这张表中查一次money 和version，再通过消息发给consumer，才说得通

但这样一来就感觉有点多此一举，等于说producer负责select查询，而consumer负责update更新，这也太尬了吧，正常来说，所有的money的计算和修改都由同一个服务完成的，没道理分两个服务，也就说前一个服务不应该去负责money的计算，而是应该直接把整个订单作为消息发给money服务，由money服务根据订单自己去select 和update，此时再利用数据库自带的乐观锁机制才说得通

而“消息链路中的幂等”例如常用的token等手段，它们都是给消息本身做唯一标识的，而不是给最终的money做标识（给money做version标识主要是解决数据库多线程的读写幂等，用这个来解决消息幂等也太尬了）

2. 消息补偿：这个图虽然说没错，也确实是那么用的，但有一说一，作为教程也太敷衍了

这个图本身的逻辑稍微花点时间就能理解，但问题是教程真正要教会的是：为什么要这样设计架构？这个问题是最关键的，反而只字未提

事实上为了实现“消息补偿”做法不止一种，例如最容易想到的“消息打标”，但为什么偏偏最终选择了上面这个复杂的方式，教程完全没说明，这种就跟抄答案一样的教学方式也太敷衍了我

2021-03-24 17:47 1 回复



60元钱小精灵

另外，关于“消息幂等”（不是数据库中锁）反而一个是“预生成id”和“token”2种方式比较好用一点，例如微信支付里，每次支付前都会自动生成一个“预支付订单”的东西，虽然没有仔细去细究，但应该就是通过预先生成id来实现“消息幂等”，而阿里的支付中貌似没有“预支付订单”的东西存在，估计就是使用第二种即token的方式

2021-03-24 17:58 回复



60元钱小精灵

反正目前来看，我甚至觉得这个讲师在说这个问题的时候，可能自己都没真正搞懂当然也有可能是我搞错了，坐等打脸

2021-03-24 17:50 回复