

# 优惠券系统设计

WX: study322 其他均为翻录倒卖  
Coupon System

主讲人 南帝老师

Message Queue  
分布式事务 表单设计 Touch System 索引  
SQL Table Design 优惠券 消息队列 API Ratelimit  
缓存预热 过期券通知  
Distributed Transaction Coupon  
触达系统 Expired conpon notification

# Scenario 场景

设计一个优惠券系统 Coupon System

优惠券的种类

优惠券系统的核心流程





我的淘宝

首页 账户设置 消息

我的交易

我的购物车

已买到的宝贝

我的拍卖

机票酒店保险

我的彩票

我的游戏

我的理财

购买过的店铺

我的收藏

我的积分

我的优惠信息

红包

优惠券

店铺会员卡专区

评价管理

退款维权

我的足迹

优惠卡券

店铺名称

购物券 1 网店优惠券 9 生活服务卡 0 支付宝红包 税值折扣 0 免息券

优惠金额

¥20 全店通用

使用条件: 满20.01  
有效时间: 2020.11.01 至 2020.11.03  
发行店铺: adidas官方旗舰店

199.00

399.00

229.00

¥20 全店通用

使用条件: 满20.01  
有效时间: 2020.11.01 至 2020.11.03  
发行店铺: adidas官方旗舰店

199.00

399.00

229.00

¥60 全店通用

使用条件: 满499.00  
有效时间: 2020.11.01 至 2020.11.03  
发行店铺: adidas官方旗舰店

199.00

399.00

229.00

¥20 全店通用

使用条件: 满20.01  
有效时间: 2020.11.01 至 2020.11.03  
发行店铺: adidas官方旗舰店

199.00

399.00

229.00

¥20 全店通用

使用条件: 满20.01  
有效时间: 2020.10.12 至 2020.11.11  
发行店铺: 大麦网官方旗舰店

120.00

80.00

80.00

¥10 全店通用

使用条件: 满199.00  
有效时间: 2020.11.01 至 2020.11.11  
发行店铺: 宝洁官方旗舰店

125.70

199.00

209.70

¥100 全店通用

使用条件: 满100.01  
有效时间: 2020.07.20 至 2021.01.16  
发行店铺: EWDHK男女鞋品牌店

269.00

249.00

259.00

¥100 全店通用

使用条件: 满100.01  
有效时间: 2020.07.20 至 2021.01.16  
发行店铺: EWDHK男女鞋品牌店

269.00

249.00

259.00

店铺名称

优惠券

Today's Deals

Coupons

Renewed Deals

Outlet

Warehouse Deals

Digital Deals

Woot! Deals

amazon

Amazon Coupons

Deliver to Hong Kong

Today's Deals

Customer Service

Gift Cards

Registry

Sell

Your Coupons

Recently Clipped

Popular Coupons

Most Popular

Subscribe & Save

Prime Pantry

Health & Personal Care

Baby & Child Care

Vitamins & Dietary Supplements

Household Supplies

Health Care

Personal Care

Grocery & Gourmet

Breakfast Foods

Beverages & Coffee

Snack Food

Chocolate

Gluten-Free

Baking Goods & Cooking Supplies

Pay with SNAP EBT

Beauty

Makeup

Skin Care

Hair Care

Tools & Accessories

Other Categories

Automotive

Amazon Coupons

Most Popular Coupons

Save up to \$1.50

\$1.50 OFF ONE Select Pampers Diapers

Clip Coupon

Save up to 10%

10% OFF on Frozen Vegetables & Sides

Clip Coupon

Save up to \$4.00

\$4 OFF ONE Cascade Platinum Plus

Clip Coupon

Discount on first order.

Save up to \$4.00

\$4 OFF ONE Gillette Blade Refill (4 ct or

Clip Coupon



九章算法  
帮助更多中国工程师找工作

课程 旗舰课 1对1私教 免费课 领扣题解 成功案例 更多...

下载 APP 邀请有礼 消息 任务 我的课程

我的课程

我的提问

学分商城

优惠券

个人资料

帐号设置

③ 可用学分: 2160分 学分说明

课程直减券 直减券

¥137/\$20

【限时兑换】全场课程通用券

③ 550学分

仅限兑换一次

课程折扣券 折扣券

100%OFF

【限时】《递归四讲》课程兑换券

③ 550学分

仅限兑换一次

课程折扣券 折扣券

100%OFF

【限时】《海量数据处理算法和面试题全集》课程兑换券

③ 550学分

仅限兑换一次

课程直减券 直减券

¥200/\$30

直播+互动课程直减优惠券

③ 2000学分

立即兑换

LintCodeVIP-7天 折扣券

100%OFF

领扣7天VIP

③ 550学分

立即兑换

LintCodeVIP-30天 折扣券

100%OFF

领扣30天VIP

③ 2000学分

立即兑换



VX: study322 其他均为翻录倒卖







**发券的方式：**同步发送 or 异步发送



VX: study322 其他均为翻录倒卖

**谁能领：**所有用户 or 指定的用户  
**领取上限：**一个优惠券最多能领取多少张？  
**领取方式：**用户主动领取 or 自动发放被动领取



**作用范围：**商品、商户、类目  
**计算方式：**是否互斥、是否达到门槛等

## 商家侧

Coupon  
创建优惠券

Coupon  
发送优惠券

## 用户侧

Coupon  
领取优惠券

VX: study322 其他均为翻录倒卖

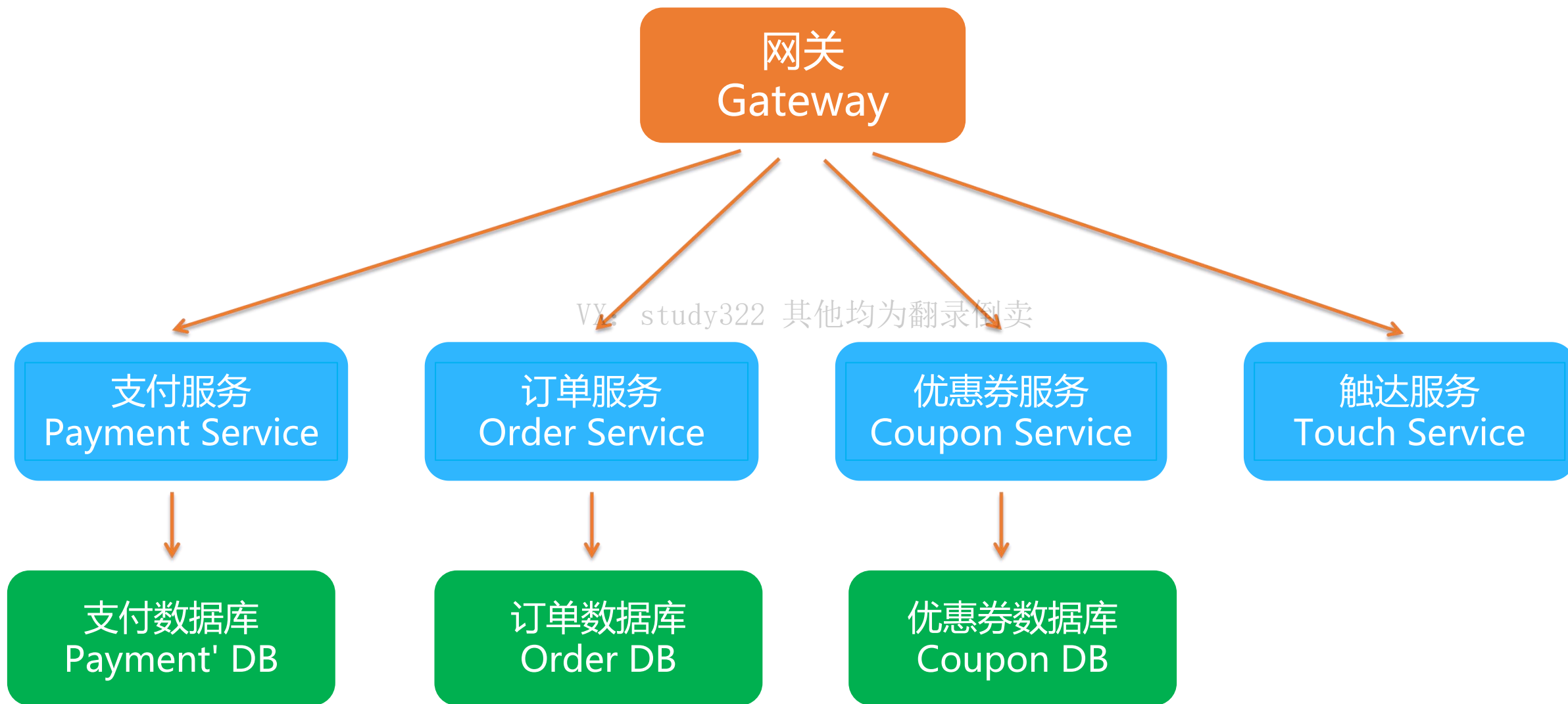
下单

Coupon  
使用优惠券

支付

# Service服务

了解优惠券系统 Coupon System 相关服务  
优惠券系统设计难点



01

Distributed Transaction

券的分布式事务，使用券的过程会出现的分布式问题分析

02

如何防止超发

03

VX: study322 其他均为翻录倒卖

如何大批量给用户发券

04

如何限制券的使用条件

05

如何防止用户重复领券



# Storage存储

VX: study322 其他均为翻录倒卖

## 模型的设计

### 优惠券系统 Coupon System 模型定义

### 优惠券系统的难点精讲

coupon\_batch

## 1. 券批次 (券模板)

Coupon  
指一批优惠券的抽象、模板，包含优惠券的大部分属性。

例：商家创建了一批优惠券，共1000张，使用时间为2020-11-11 00:00:00 ~ 2020-11-11 23:59:59，规定只有化妆品类目商品才能使用，满100减50。

## 2. 券

Coupon  
发放到用户的一个实体，已与用户绑定。

例：将某批次的优惠券中的一张发送给某个用户，此时优惠券属于用户。

## 3. 规则

优惠券的使用有规则和条件限制，比如满100减50券，需要达到门槛金额 100元才能使用



券批次表 coupon\_batch Table

字段 column	类型 type	解释 explain	例子 example
batch_id	INTEGER	批次ID	1111
batch_name	INTEGER	批次名称	“双十一发送券”
coupon_name	VARCHAR	券名称	“劳斯莱斯5元代金券”
rule_id	INTEGER	规则，外键	1010
total_count	INTEGER	总数量	10000
assign_count	INTEGER	已发放券数量	5000
used_count	INTEGER	已使用数量	2500

规则表 rule Table

字段 column	类型 type	解释 explain	例子 example
rule_id	INTEGER	规则id, 自增	1010
name	VARCHAR	名称	"满减规则"
type	INTEGER	优惠券类型, 0—满减, 1—折扣	0
rule_content	BLOB	规则内容	{ threshold: 100    // 使用门槛 amount: 10        // 优惠金额 ..... }

规则内容

```
{
  threshold: 5.01           // 使用门槛
  amount: 5                 // 优惠金额
  use_range: 3              // 使用范围, 0—全场, 1—商家, 2—类别, 3—商品
  commodity_id: 10         // 商品 id
  receive_count: 1         // 每个用户可以领取的数量
  is_mutex: true            // 是否互斥, true 表示互斥, false 表示不互斥
  receive_started_at: 2020-11-1 00:08:00 // 领取开始时间
  receive_ended_at: 2020-11-6 00:08:00   // 领取结束时间
  use_started_at: 2020-11-1 00:00:00      // 使用开始时间
  use_ended_at: 2020-11-11 11:59:59      // 使用结束时间
}
```



优惠券表 coupon Table

字段 column	类型 type	解释 explain	例子 example
coupon_id	INTEGER	券ID, 主键	66889
user_id	INTEGER	用户id	1001
batch_id	INTEGER	批次ID	1111
status	INTEGER	状态, 0—未使用, 1—已使用, 2—已过期, 3—冻结	1
order_id	VARCHAR	对应订单ID	12168257647382907847
received_time	DATETIME	领取时间	2020-11-07 00:12:48
validate_time	DATETIME	有效日期	2020-11-18 00:01:48
used_time	DATETIME	使用时间	2020-11-11 00:01:48

## 1. 新建规则

```
INSERT INTO rule (name, type, rule_content)
VALUES( "满减规则" , 0, '{
    threshold: 100
    amount: 10
    .....
}');
```

VX: study322 其他均为翻录倒卖



商家建券

## 2. 新建优惠券批次

```
INSERT INTO coupon_batch (coupon_name, rule_id, total_count )
VALUES( "劳斯莱斯5元代金券" , 1010, 10000);
```



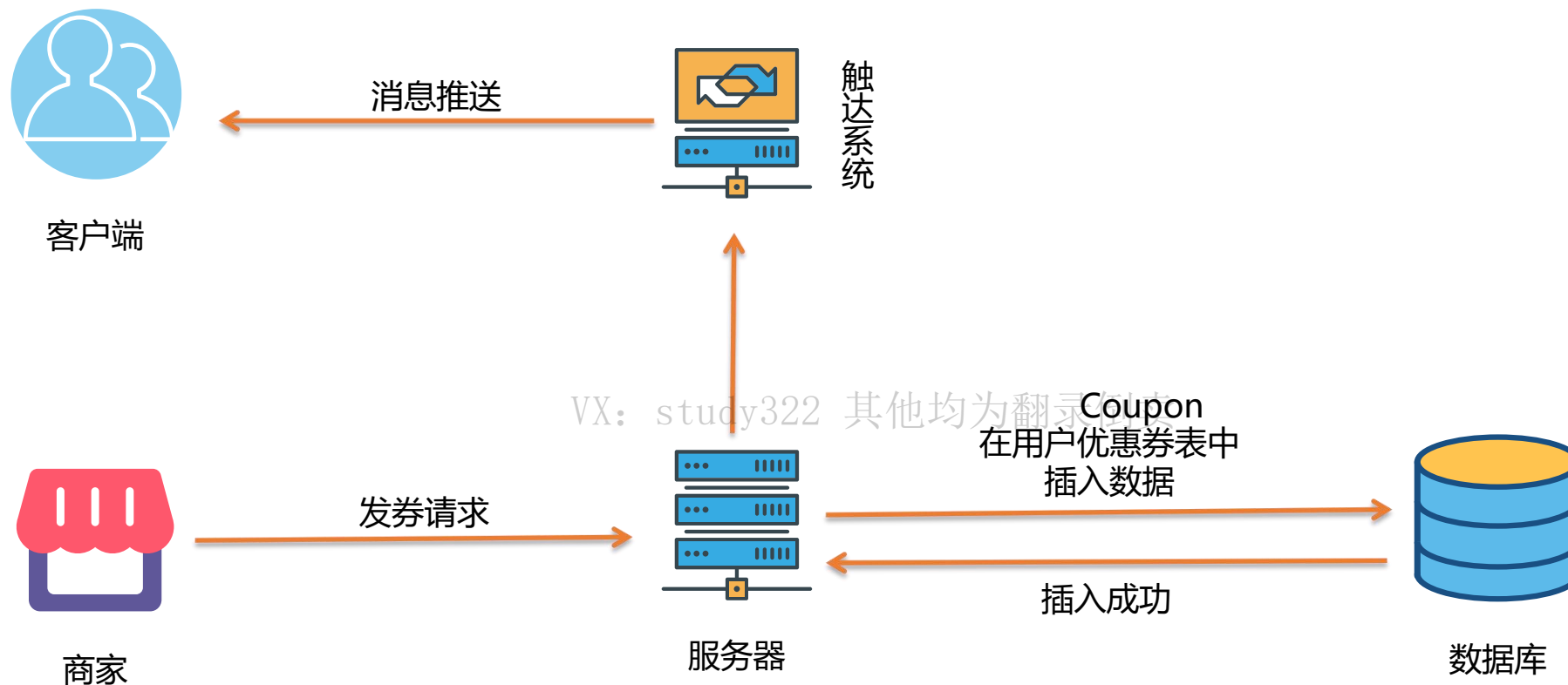
商家

发券

VX: study322 其他均为翻录倒卖



用户



如何给大量用户发券?

asynchronous  
异步发送





## 信息表 message Table

字段 column	类型 type	解释 explain	例子 example
id	INTEGER	信息id	1111
send_id	INTEGER	发送者id	
rec_id	INTEGER	接受者id	
content	VARCHAR	站内信内容	
is_read	INTEGER	是否已读	
send_time	DATETIME	发送时间	

我们先考虑用户量很少的情况，如果商家要给所有人发站内信，则先遍历用户表，再按照用户表中的所有用户依次将站内信插入到 message 表中。这样，如果有100个用户，则群发一条站内信要执行100个插入操作。

## 如果系统的用户数量增加到上万人呢？

这样发一条站内信，就要重复插入上万条数据。而且这上万条数据的 content 是一样的，假设一条站内信占 100 个字节，发一次站内信就要消耗十几 MB。因此我们可以将原来的表拆成两个表。

## 信息表 message Table

字段 column	类型 type	解释 explain	例子 example
id	INTEGER	信息id	1111
send_id	INTEGER	发送者id	
rec_id	INTEGER	接受者id	
message_id	INTEGER	外键, 信息内容id	
is_read	INTEGER	是否已读	

## 信息内容表 message\_content Table

字段 column	类型 type	解释 explain	例子 example
id	INTEGER	信息内容id	1111
content	VARCHAR	内容	
send_time	DATETIME	发送时间	

在发一封站内信的时候，执行两步操作。

在 message\_content 表中，  
插入站内信的内容

01

VX: study322 其他均为翻录倒卖

02

在 message 表中给所有的用户插入一条记录，标识有一封站内信。

## 如果系统的用户数量增加到上千万人呢？

- 这里会有非活跃用户的问题，假设注册用户一千万，其中活跃用户只占其中的20%。
- 如果采用上面拆分成两个表的情况，发一封“站内信”，那得执行一千万个插入操作。可能剩下的80%用户基本都不会再登录了，我们只需要对其中20%的用户插入数据即可。

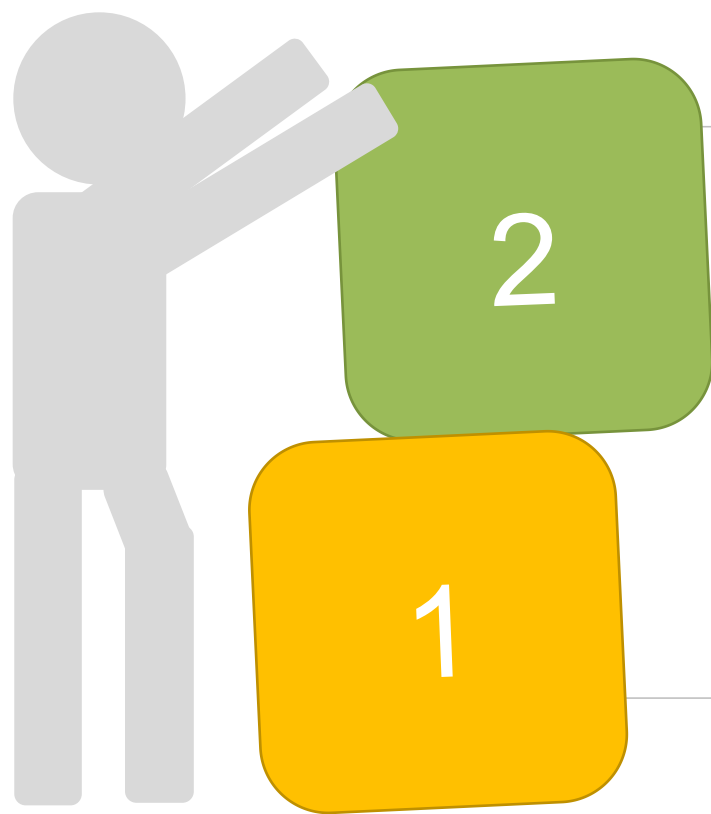


## 信息表 message Table

字段 column	类型 type	解释 explain	例子 example
id	INTEGER	信息id	1111
rec_id	INTEGER	接受者id	
message_id	INTEGER	外键, 信息内容id	
is_read	INTEGER	是否已读	

## 信息内容表 message\_content Table

字段 column	类型 type	解释 explain	例子 example
id	INTEGER	信息内容id	1111
send_id	INTEGER	发送者id	
content	VARCHAR	内容	
send_time	DATETIME	发送时间	



### 系统

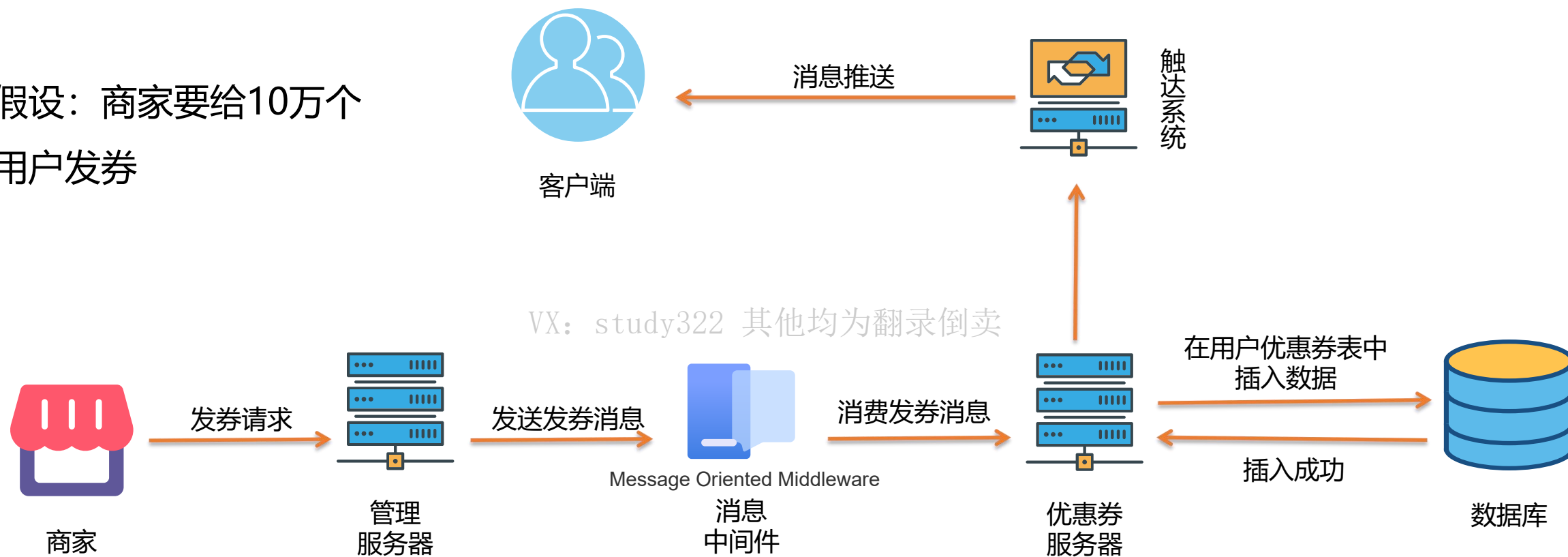
发站内信的时候，只在 `message_content` 插入站内信的主体内容，`message` 里不插入记录。

VX: study322 其他均为翻录倒卖

### 用户

用户登录后，首先查询 `message_content` 中的那些没有在 `message` 中有记录的数据，表示是未读的站内信。在查阅站内信的内容时，再将相关的记录插入到 `message` 中。

假设：商家要给10万个  
用户发券



这样还会有什么问题?

重复消费的问题，会引发超发

1. 运营提供满足条件的用户文件，上传到发券管理后台并选择要发送的优惠券

2. 管理服务器根据用户ID和券批次ID生成消息，发送到消息中间件中

Message Oriented Middleware

3. 优惠券服务器消费消息

VX: study322 其他均为翻录倒卖

```
INSERT INTO coupon (user_id, coupon_id, batch_id)
VALUES(1001, 66889, 1111);
```



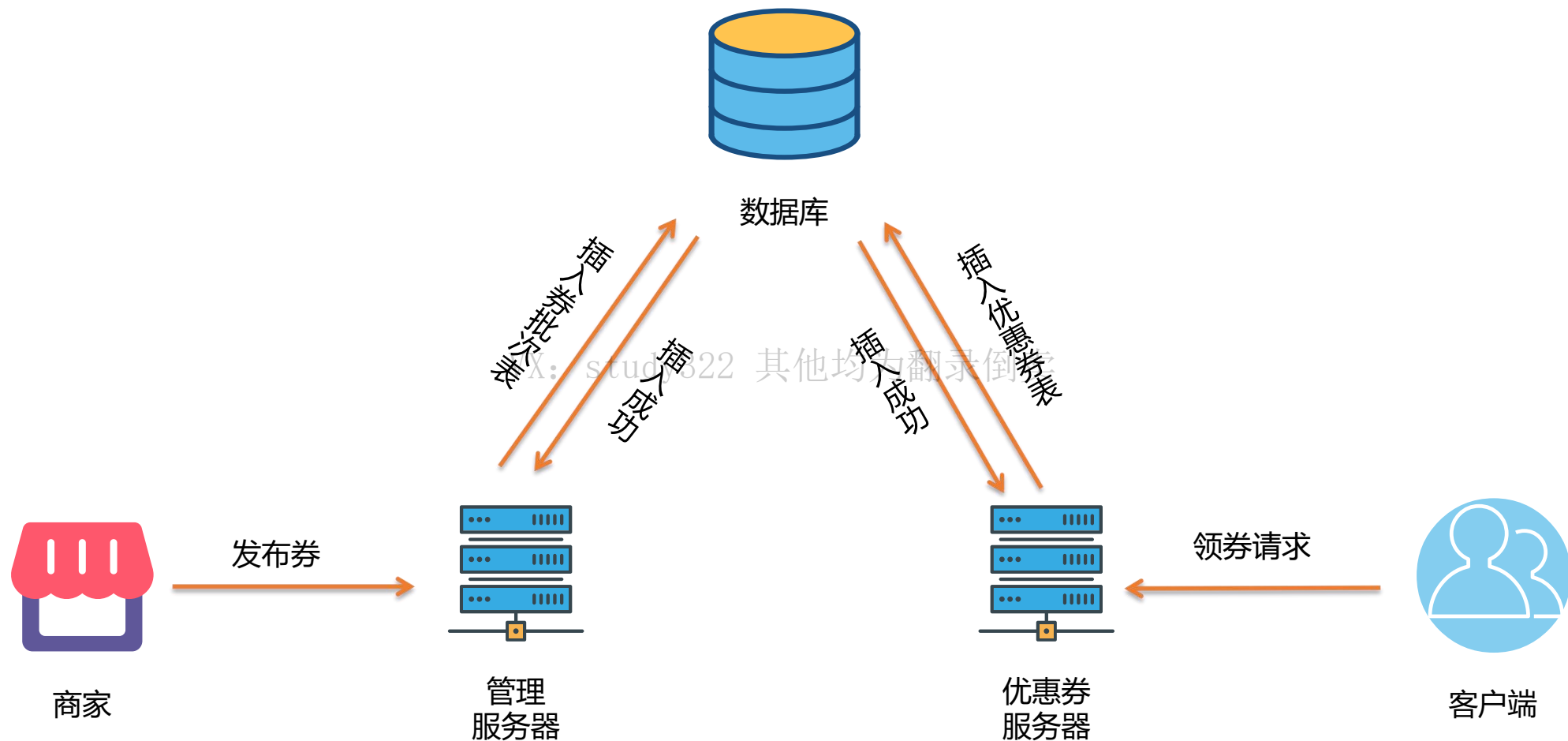
```
UPDATE coupon_batch SET total_count = total_count - 1, assign_count =
assign_count + 1
```

```
WHERE batch_id = 1111 AND total_count > 0;
```



商家发券





## 1. 校验优惠券余量

```
SELECT total_count FROM coupon_batch  
WHERE batch_id = 1111;
```



其实这里也会有因为并发导致的超领问题，大家可以思考一下，这里是怎么防止超领的？



## 2. 新增优惠券用户表，扣减余量

study322 其他均为翻录倒卖

```
INSERT INTO coupon (user_id, coupon_id, batch_id)  
VALUES(1001, 66889, 1111);
```



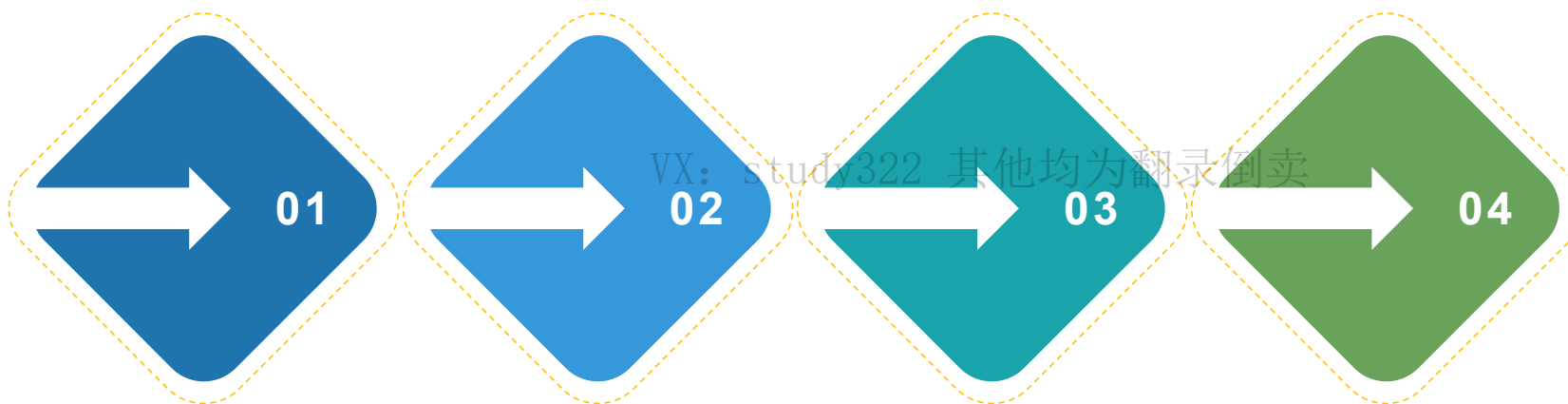
用户领券

```
UPDATE coupon_batch SET total_count = total_count - 1, assign_count =  
assign_count + 1  
WHERE batch_id = 1111 AND total_count > 0;
```

在用户领券的过程中，其实也会出现类似秒杀的情况，大家可以回想一下在秒杀场景下会有哪些问题已经我们是如何解决的？



用户领券



问题

高并发导致数据库崩溃

措施

缓存预热

问题

超高并发导致缓存放行量大  
使数据库崩溃

措施

message queue  
消息队列异步处理



## 如何防止用户重复领取或多领? Redis数据校验

### 1. 在领券前先查缓存

语法: SISMEMBER KEY VALUE

作用: 判断成员元素是否是集合的成员。

实例: SISMEMBER batch\_id:1111:user\_id 1001

VX: study322 其他均为翻录倒卖

### 2. 领券

### 3. 领券后更新缓存

语法: SADD KEY VALUE1.....VALUEN

作用: 将一个或多个成员元素加入到集合中, 已经存在于集合的成员元素将被忽略。

实例: SADD batch\_id:1111:user\_id 1001

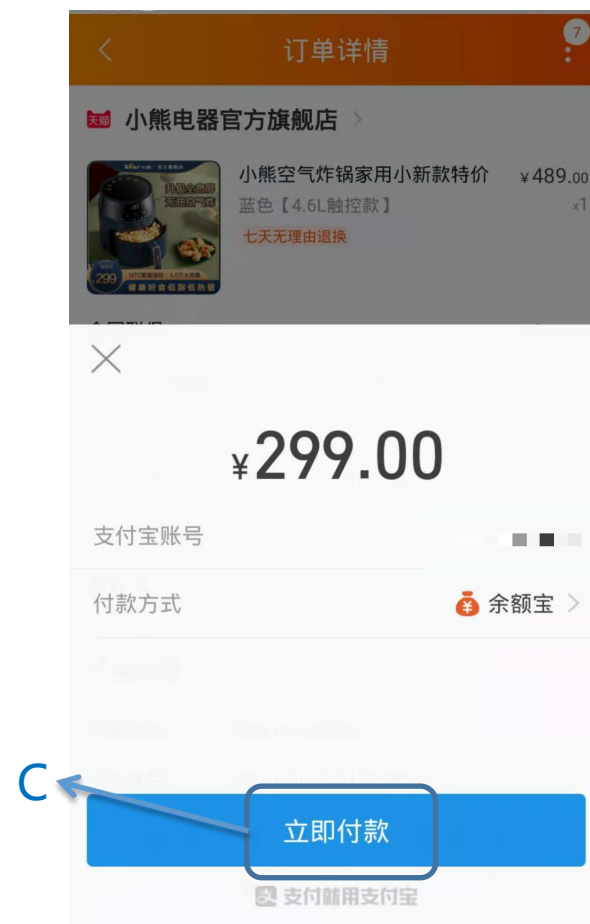
# 什么时候对优惠券使用规则进行校验?

A. 确认订单



B. 提交订单

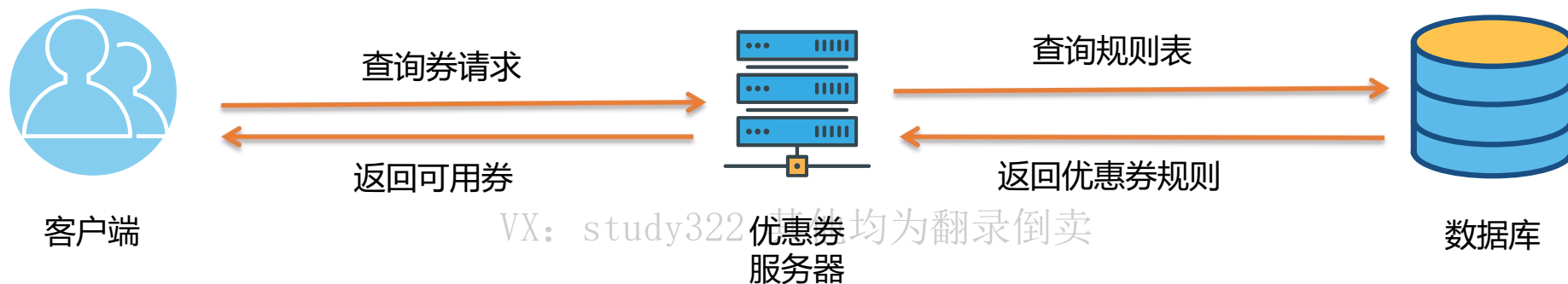
C. 立即付款



确认订单页，对优惠券进行校验

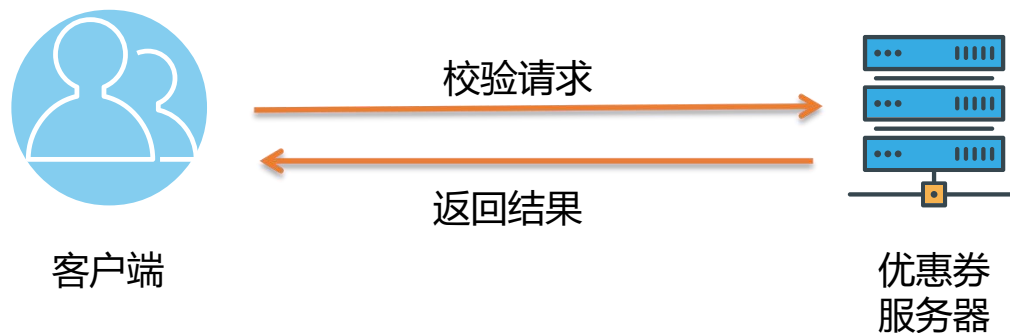


## 返回可用券



VX: study322 其他均为翻录倒卖

## 选择可用券



## 1. 返回可用券

```
SELECT batch_id FROM conpou  
WHERE user_id = 1001 AND status = 0;
```

```
SELECT rule_id FROM coupon_batch  
WHERE batch_id = 1111;
```

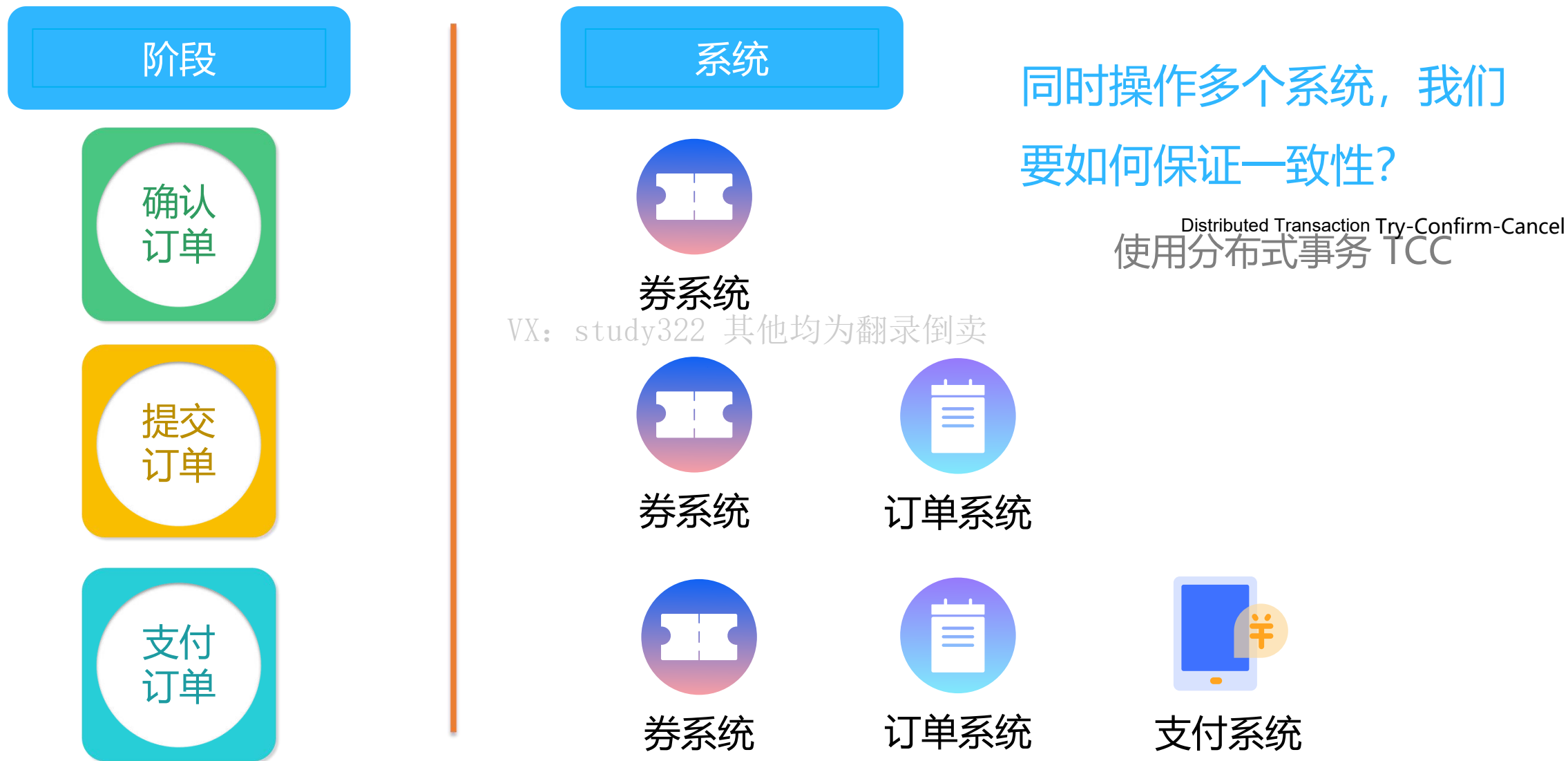
VX: study322 其他均为翻录倒卖

```
SELECT name, type, rule_content FROM rule  
WHERE rule_id = 1010;
```

## 2. 选择可用券，并返回结果



用户用券



优惠券操作记录表  
Coupon\_opt\_record Table

字段 column	类型 type	解释 explain	例子 example
user_id	INTEGER	用户id	1001
coupon_id	INTEGER	优惠券id	1111
operating	INTEGER	操作, 0—锁定, 1—核销, 2—解锁	0
operated_at	DATETIME	操作时间	2020-11-11 00:01:48

TCC是Try-Confirm-Cancel的简称，是Distributed Transaction目前分布式事务主流解决方案

## TCC 实现阶段一：Try

对资源进行冻结，预留业务资源

在创建订单时，将优惠券状态改为 “冻结”

## TCC 实现阶段二：Confirm

VX: study322 其他均为翻录倒卖

确认执行业务操作，做真正的提交，将第一步Try中冻结的资源，真正扣减

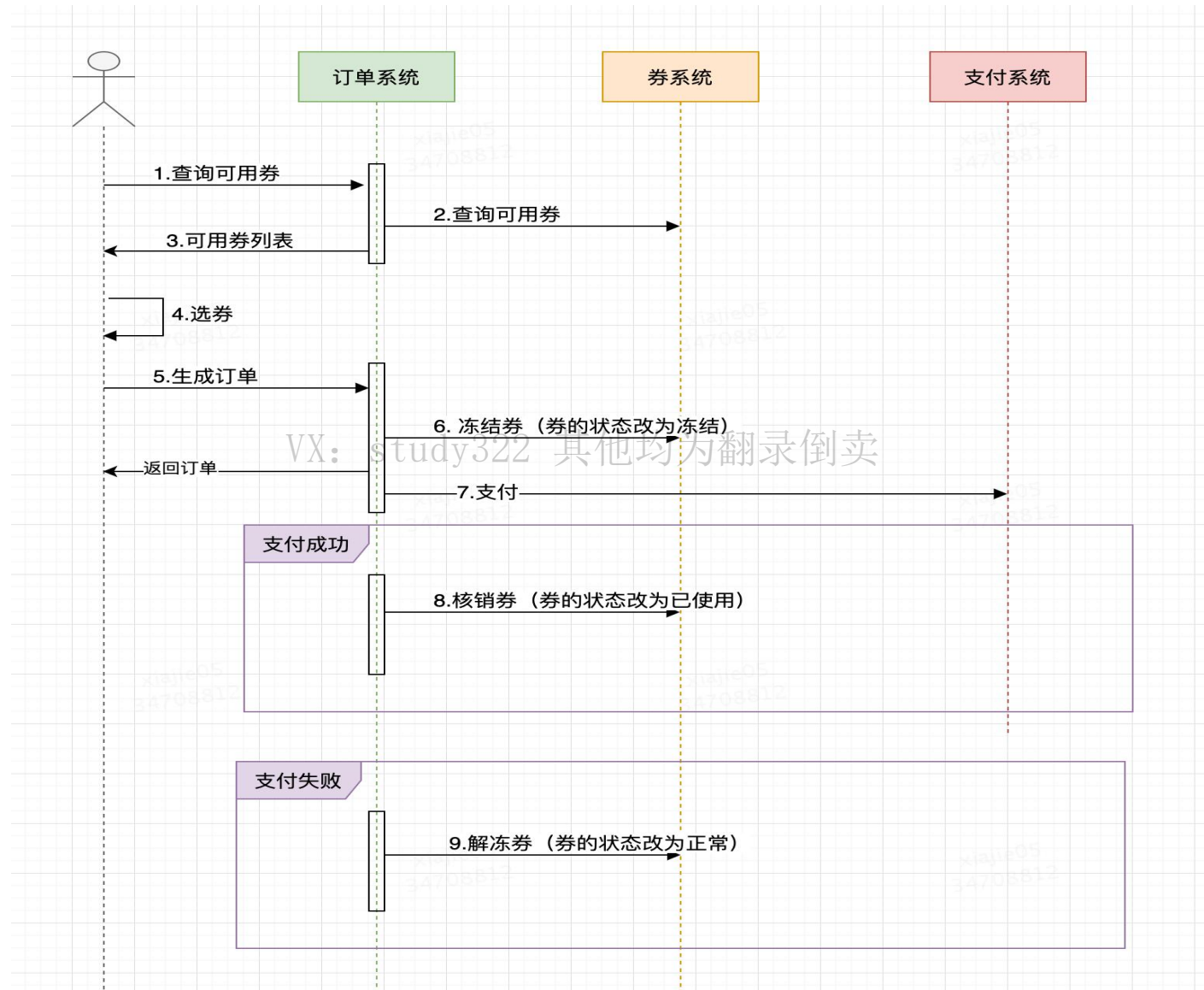
在订单支付成功，将优惠券状态改为 “已使用”

## TCC 实现阶段三：Cancel

取消执行业务操作，取消Try阶段预留的业务资源

在支付失败/超时，或者订单关闭情况，将优惠券状态改为 “未使用”





# Scale 扩展

VX: study322 其他均为翻录倒卖

快过期券提醒

数据库层面优化

发券接口限流保护

## 定时扫描券表

1

缺点：扫描的数据量太大，随着历史数据越来越多，会影响线上主业务，会出现慢 SQL。

## 新增通知表

优点：扫描的数据量小，效率高  
删除无用的已通知的数据记录

3

## 延时消息

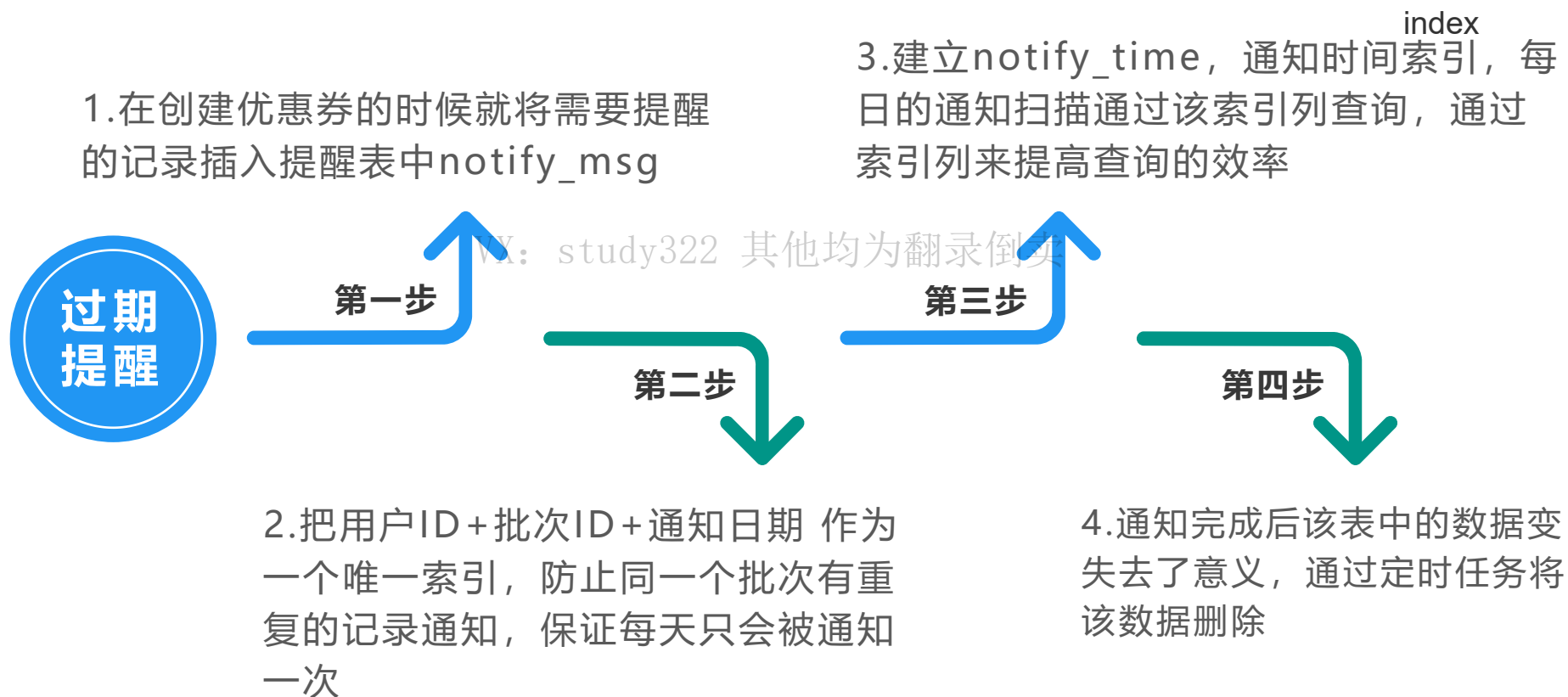
缺点：有些券的有效时间太长了(30天)以上，有可能造成大量 MQ 积压

2

VX: study322 其他均为翻录倒卖

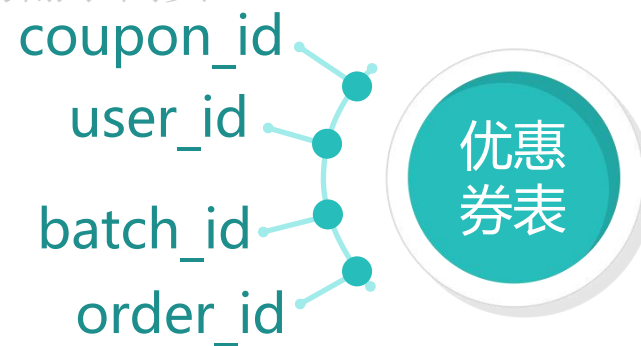
通知信息表 Notify\_msg Table

字段 column	类型 type	解释 explain	例子 example
id	INTEGER	自增主键	1100
coupon_id	BIGINT	券id	1000
user_id	BIGINT	用户id	1001
notify_day	VARCHAR	需要进行通知的日期	"2020-11-11"
notify_type	INTEGER	通知类型，1过期提醒	1
notify_time	TIMESTAMP	通知的时间，在该时间戳所在天内通知	1605059346
status	INTEGER	通知状态，0初始状态 1成功 2失败	0





VX: study322 其他均为翻录倒卖



<sup>current limiting</sup>  
前端限流

点击一次后，按钮  
短时间内置灰

抢券

抢券

<sup>current limiting</sup>  
后端限流

部分请求直接  
跳转到「繁忙页」

VX: study322 其他均为翻墙软件