

1.2、特殊提成	2
1.1 1、业务相关	3
1.1.1 1) 业务背景	4
1.1.2 2) 产品框架	7
1.1.3 3) 数据口径	9
1.1.4 4) 需求开发周期	10
1.2 2、系统架构设计	11
1.2.1 1) 提成方案业务模型	12
1.2.2 2) 数仓调度架构模型	17
1.2.2.1 调度模板格式	19
1.3 3、日常需求	22
1.3.1 2021.Q4	23
1.3.1.1 20211216_特殊提成七期-多品进店	24
1.3.2 2022.Q2	31
1.3.2.1 特殊提成8期	32
1.3.2.1.1 前端交互接口变更	33
1.3.2.2 特殊提成9期	36
1.3.2.2.1 数仓逻辑修改梳理	37
1.3.3 2023.Q1	41
1.3.3.1 2023_02-特殊提成日常	42
1.3.3.2 大BD接入特殊提成	46
1.4 4、资损防控	47

2、特殊提成

1、业务相关

1) 业务背景

- 原始产品需求
- 业务现状
 - 薪资数据产品化范围
 - 销售薪资构成
 - 薪资汇总流程及发放业务流程
 - 特殊提成业务流程
- 需求背景
 - 业务痛点

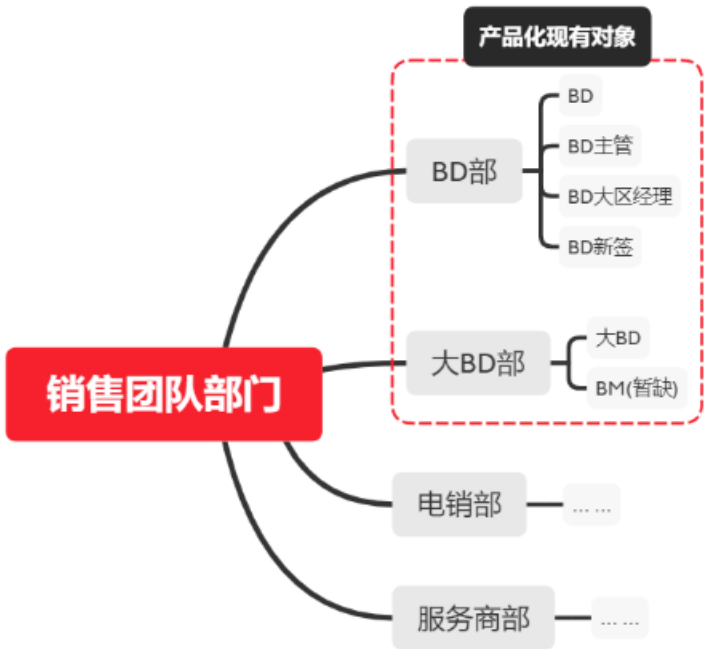
原始产品需求

PRD : 02_特殊提成项目

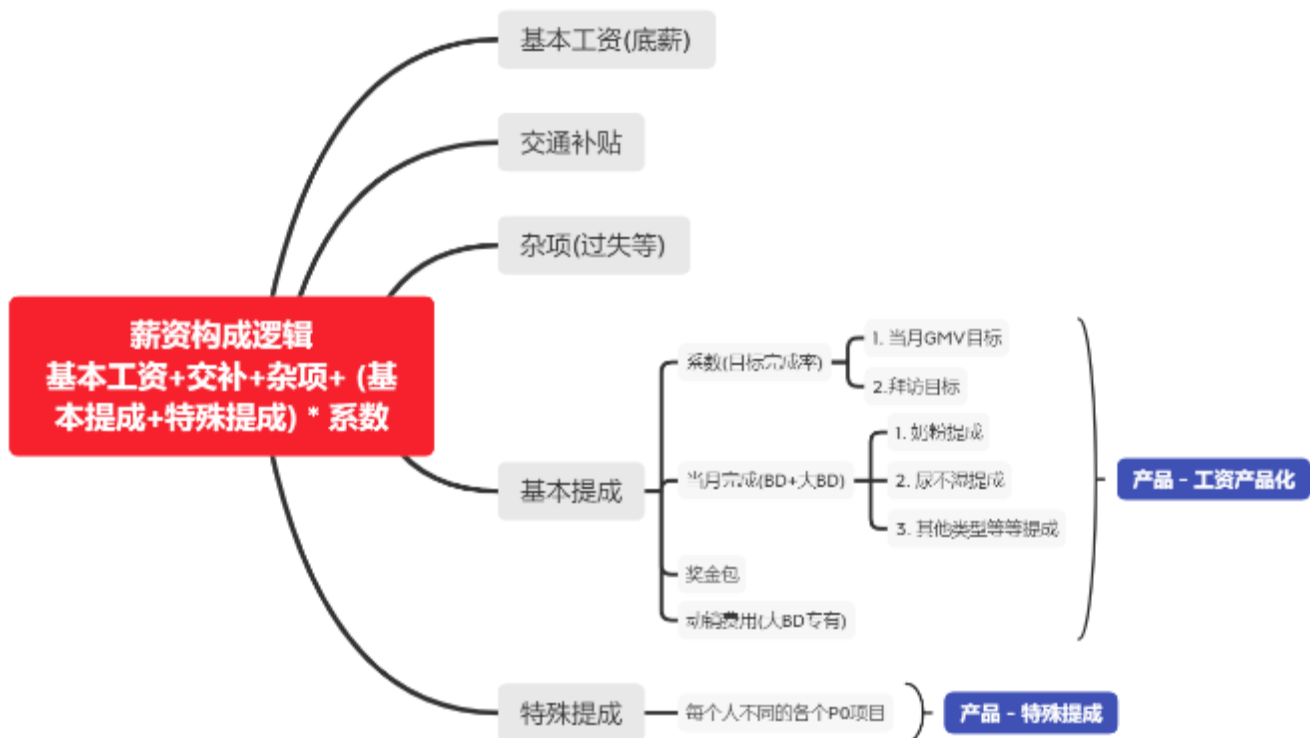
DEMO : [https://demo.yangtuoqia.com/%E4%B8%9A%E5%8A%A1%E6%9C%8D%E5%8A%A1/%E6%95%B0%E6%8D%AE%E4%B8%AD%E5%BF%83/02_%E9%A1%B9%E7%9B%AE/%E7%89%B9%E6%AE%8A%E6%8F%90%E6%88%90%E4%BA%A7%E5%93%81%E5%8C%96/%E6%8F%90%E6%88%90%E7%B3%BB%E7%BB%9Fdemo\(%E6%96%B0\)/index.html](https://demo.yangtuoqia.com/%E4%B8%9A%E5%8A%A1%E6%9C%8D%E5%8A%A1/%E6%95%B0%E6%8D%AE%E4%B8%AD%E5%BF%83/02_%E9%A1%B9%E7%9B%AE/%E7%89%B9%E6%AE%8A%E6%8F%90%E6%88%90%E4%BA%A7%E5%93%81%E5%8C%96/%E6%8F%90%E6%88%90%E7%B3%BB%E7%BB%9Fdemo(%E6%96%B0)/index.html)

业务现状

薪资数据产品化范围

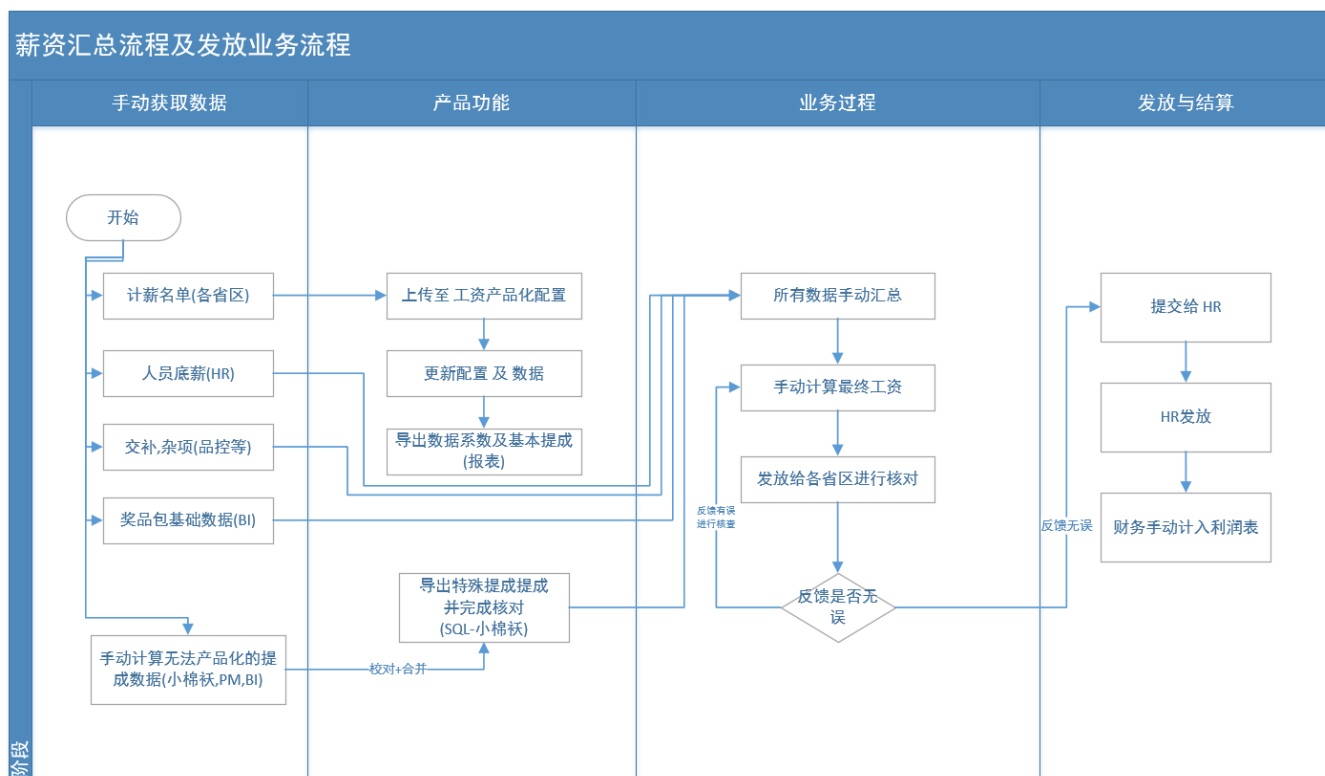


销售薪资构成



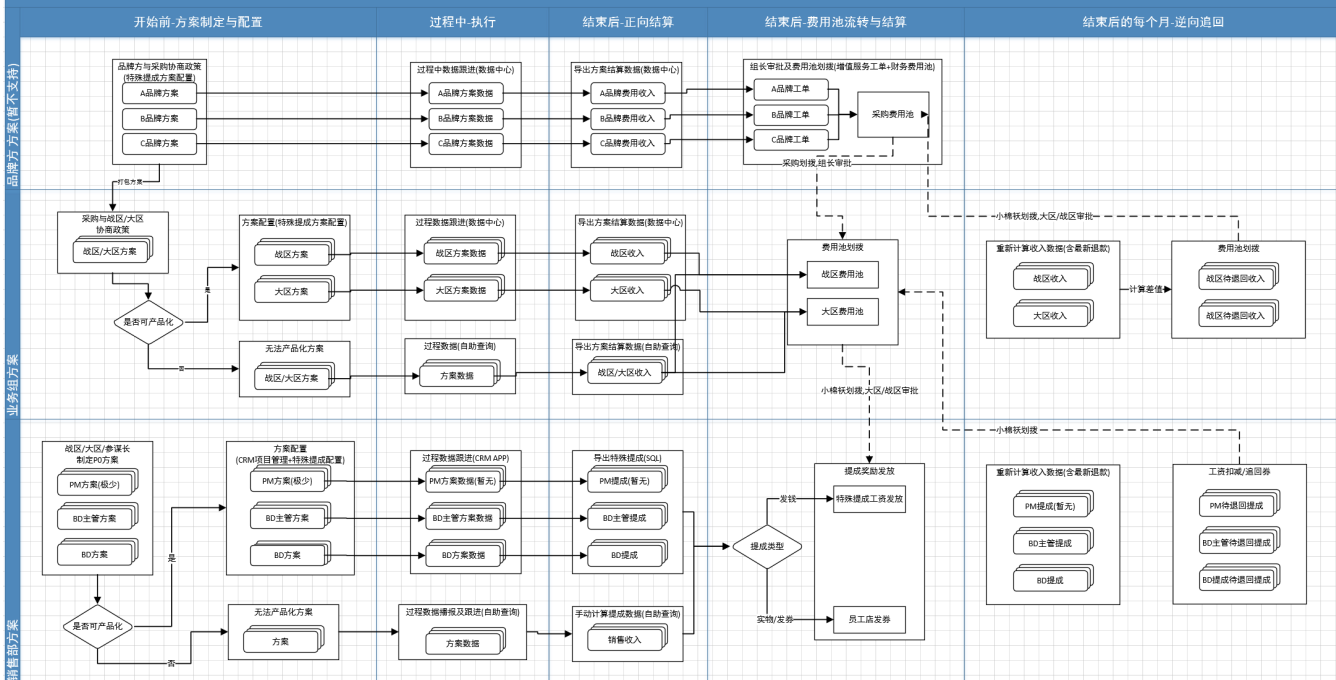
薪资汇总流程及发放业务流程

薪资汇总流程及发放业务流程



特殊提成业务流程

特殊提成业务流程



需求背景

业务痛点

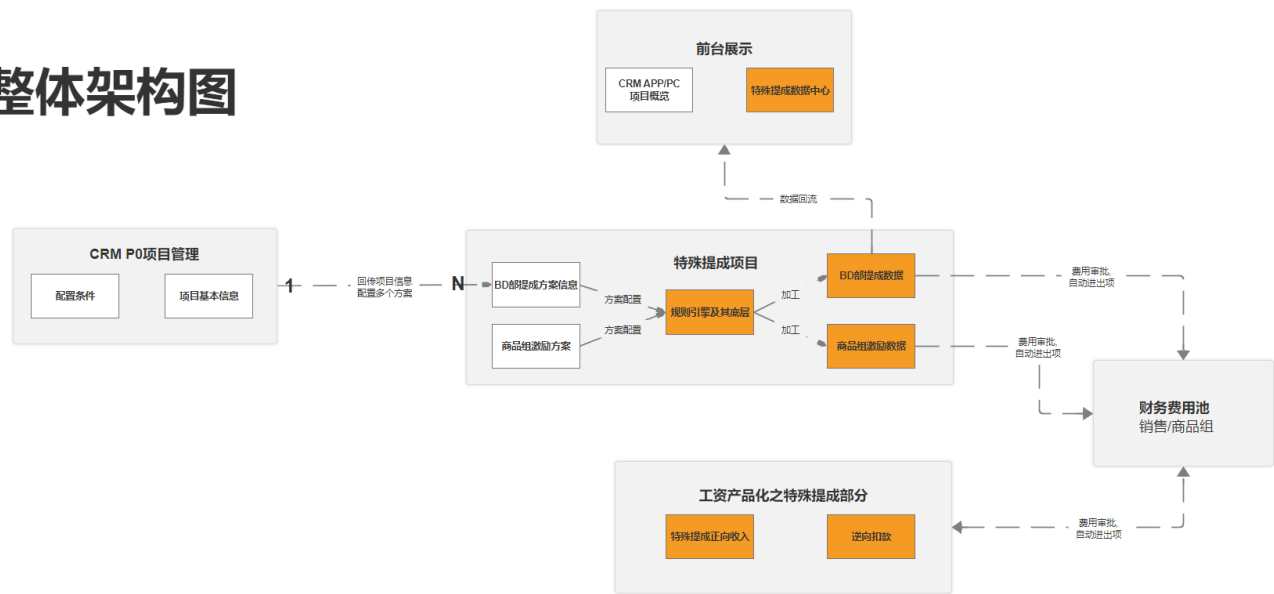
- 1) 方案数量繁多
- 2) 计算过程不严谨, 完全由业务组自行计算, 可能存在一定误差
- 3) 无法有效对清退订单进行奖励追回
- 4) 销售在过程中, 无法感知当前项目的进度
- 5) 业务规则变化后, 统计逻辑变更响应失效慢
- 6) 数据更新不及时, 需要人工手动计算, 手动播报数据

2) 产品框架

- 业务整体架构图
- 产品架构图
- 提成数据计算时间轴

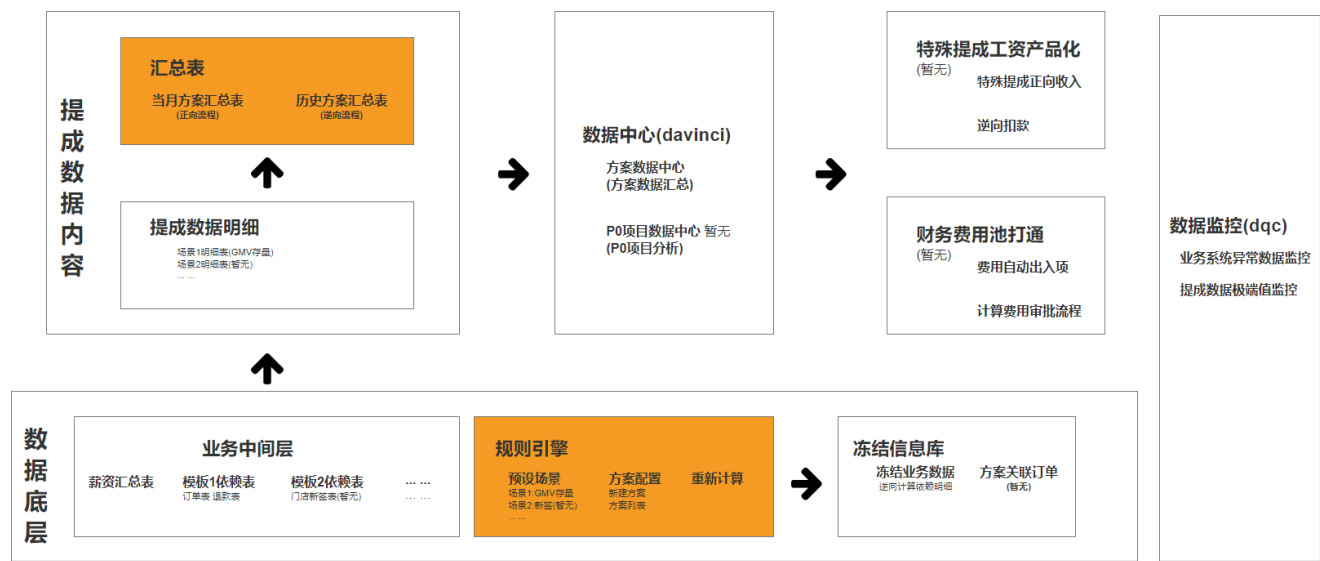
业务整体架构图

整体架构图



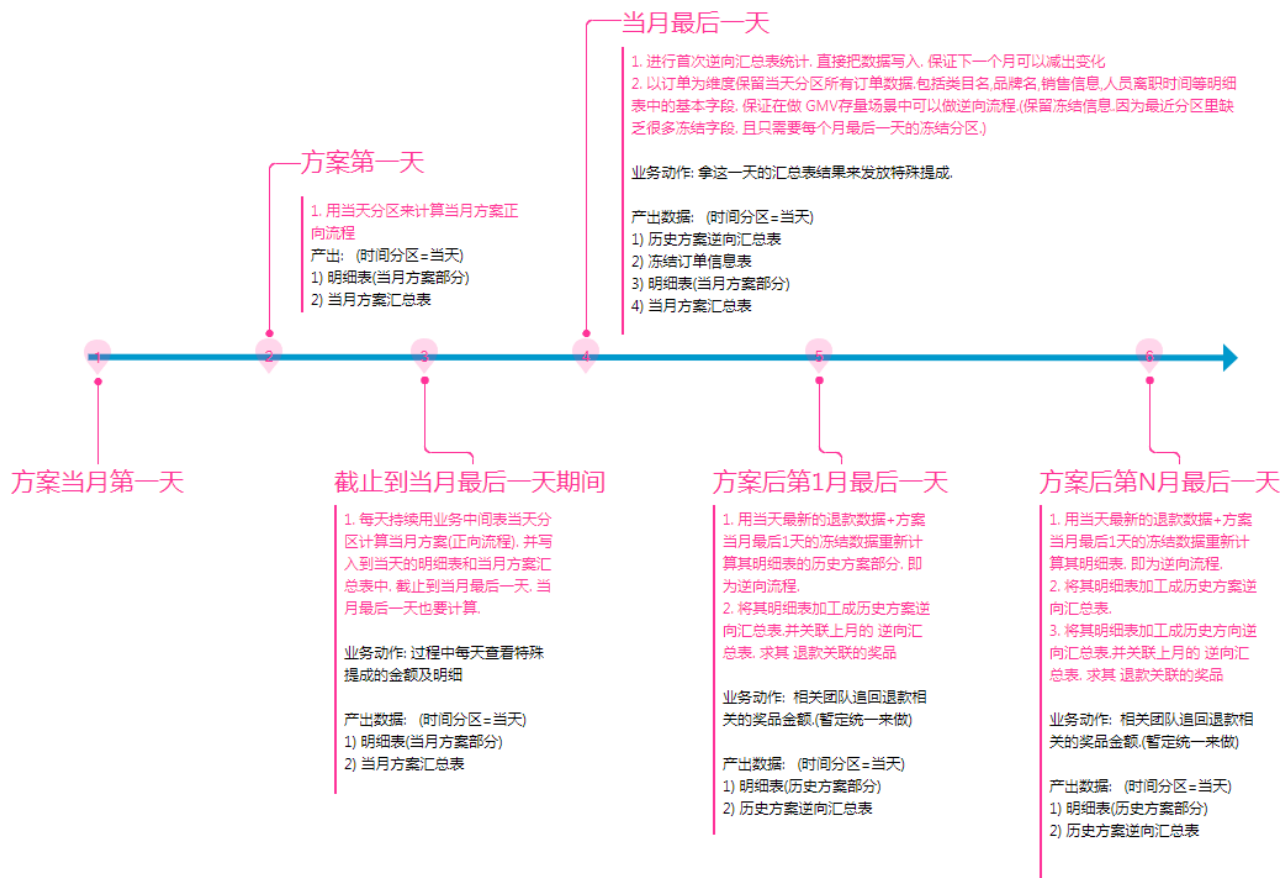
产品架构图

特殊提成产品架构



提成数据计算时间轴

计算事件时间轴



3) 数据口径

-
- 页面功能
- 数仓逻辑
 - 模板明细表
 - 汇总表

页面功能

提成方案配置：[一、方案配置](#)

数仓逻辑

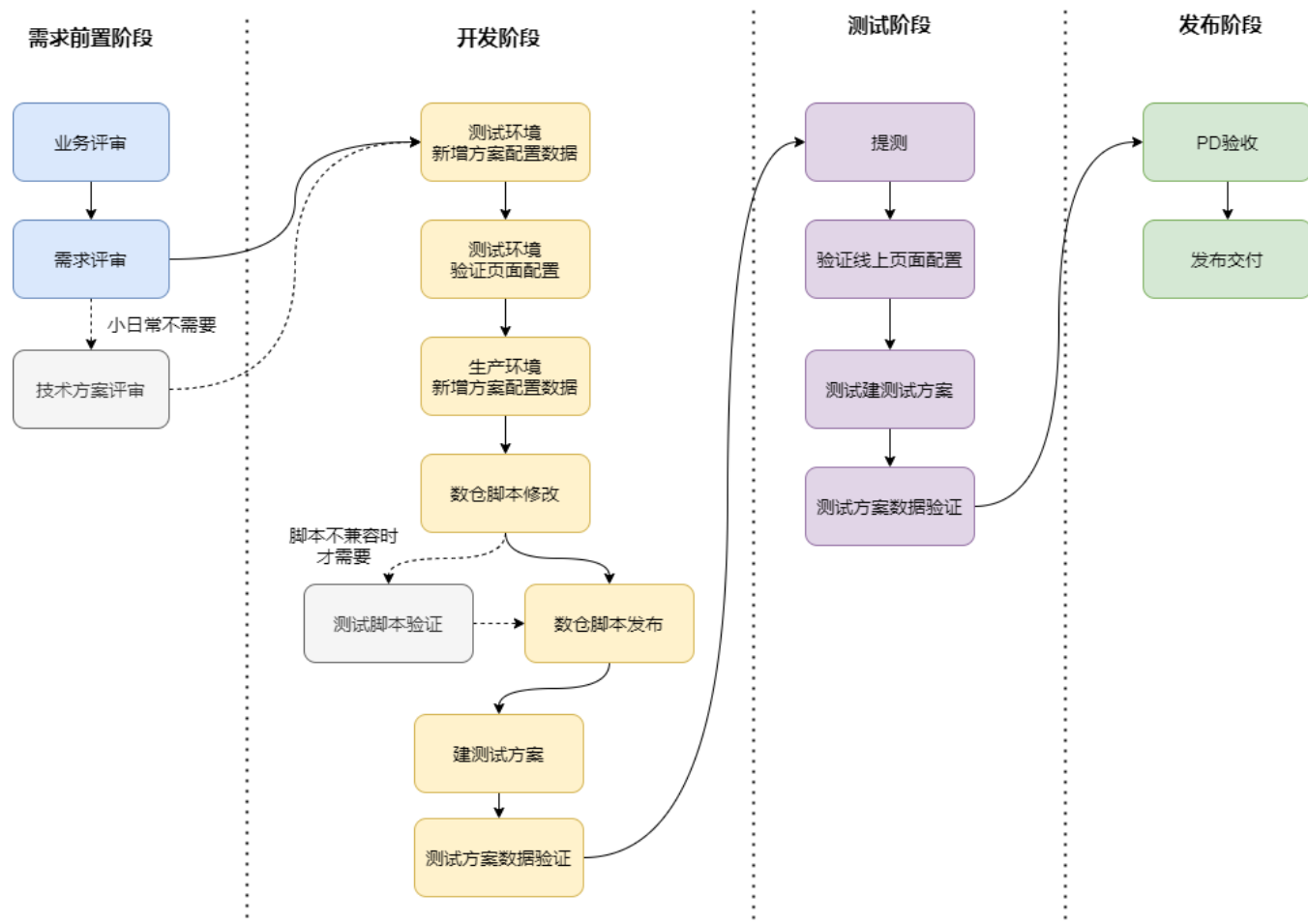
模板明细表

- gmw (dw_salary_gmw_rule_public_d)：[07_存量GMV模板规则通用明细表\(V2.0\)](#)
- 新签商品 (dw_salary_sign_item_rule_public_d)：[09_新签商品模板规则通用明细表\(V2.0\)](#)
- 新签品牌 (dw_salary_sign_brand_rule_public_d)：[08_新签品牌模板规则通用明细表\(V2.0\)](#)
- 品牌进店：[10_品牌进店 明细表](#)
- 基本提成指标 (dw_salary_base_salary_public_d)：[12. 基本提成指标](#)

汇总表

- 正向通用汇总表 (dw_salary_forward_plan_sum_d)：[01_当月方案汇总表](#)
- 当月逆向通用汇总表 (dw_salary_forward_plan_sum_mid_d)：[02_历史方案逆向汇总表](#)

4) 需求开发周期



2、系统架构设计

1) 提成方案业务模型

- [链接](#)
- [方案业务模型总览](#)
 - [事实数据 → 维度数据模型](#)
 - [维度数据模型](#)
- [模型细分](#)
 - 1) 方案基本信息 (bounty_plan)
 - 2) 发放对象 (bounty_payout_object)
 - 3) 模板规则 (bounty_rule_type)
 - 4) 统计指标 (bounty_indicator)
 - 5) 过滤条件 (bounty_filter)
 - 6) 提成奖励类型 (bounty_payout_rule_type)

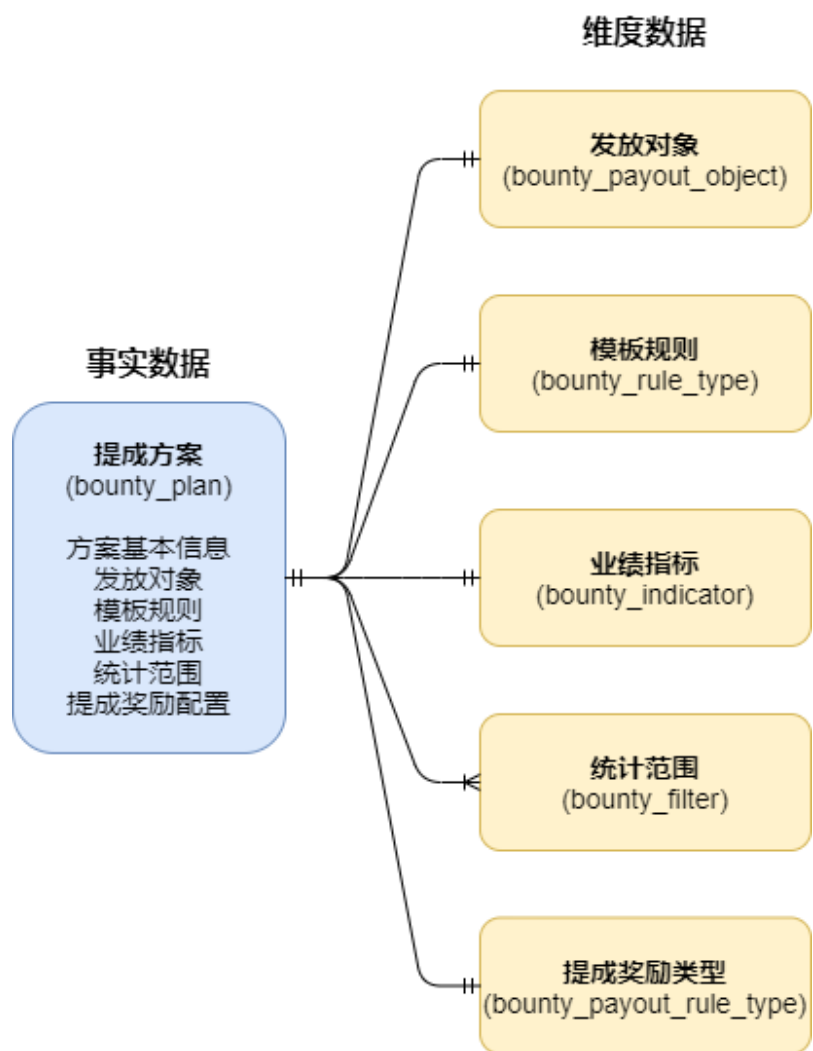
链接

提成方案页面：<https://iwork.hipac.cn/crm-addition/bountySetting>

新建方案页面：<https://iwork.hipac.cn/crm-addition/bountySettingDetail?type=add>

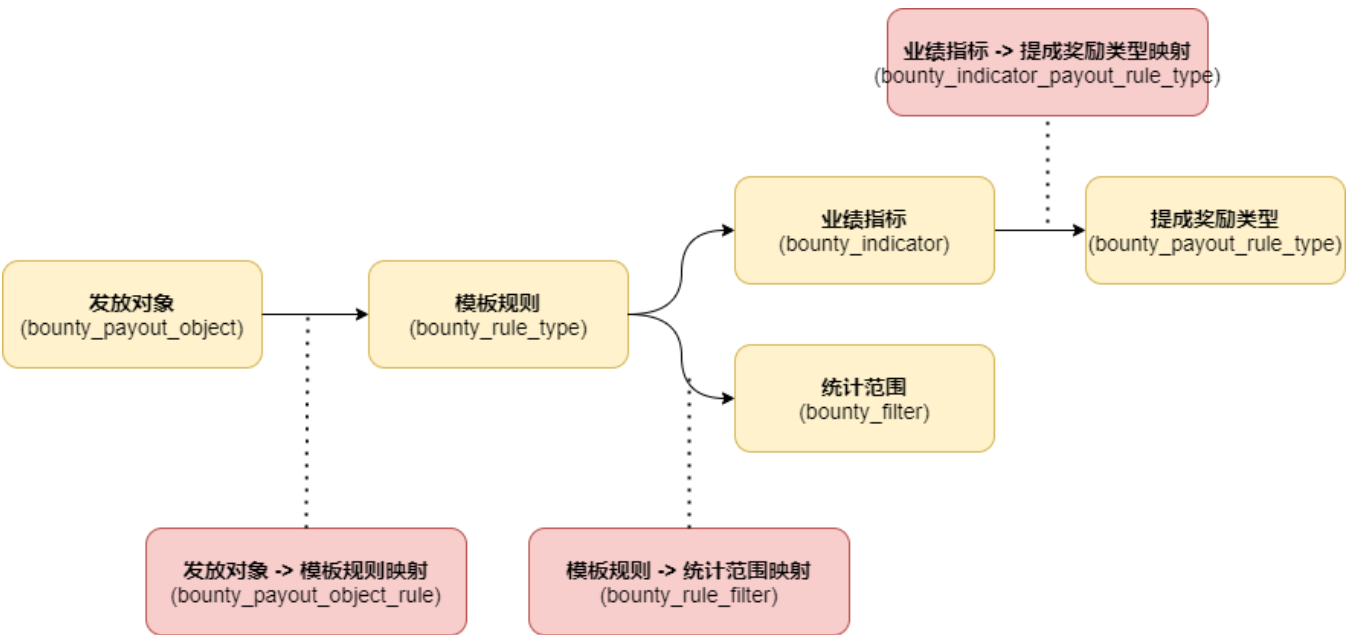
方案业务模型总览

事实数据 → 维度数据模型



bounty_rule_type 是枚举类型，可以通过crm代码BountyRuleTypeEnum 查看。

维度数据模型



模型细分

1) 方案基本信息 (bounty_plan)

方案本身的基础信息，包括方案名称，方案时间（支持月度、季度），方案归属业务方，备注

方案基本信息

* 方案名称:

新建的方案名称

* 方案时间类型:

月度方案

季度方案

* 方案月份:

2023-03

* 方案归属类型:

业务组方案

销售部方案

归属业务组:

长尾成人服孕妈妈组

备注:

方案备注备注

2) 发放对象 (bounty_payout_object)

方案统计奖励对应的人，目前支持BD、大BD、BD主管、指定受益人（业务项目PM）

一个发放对象对应bounty_payout_object表一行数据

* 方案对象:

请选择方案对象

* 模板规则:

【辖区】战区经理

【辖区】大区经理

* 业绩指标:

【辖区】BD主管

* 奖励计算时间:

【岗位】BD

【岗位】大BD

* AB类型:

【唯一】指定受益人

3) 模板规则 (bounty_rule_type)

方案对应的业务规则逻辑，基于不同发放对象，可以选择的模板不同

模板规则对应crm-master代码中 [com.yt.crm.biz.bountyplan.enums.BountyRuleTypeEnum](#) 枚举

* 模板规则:

请选择模板规则

* 业绩指标:

存量GMV

* 奖励计算时间:

新签商品

新签品牌

* AB类型:

品牌进店

基本提成指标

* 冻结销售团队标识:

4) 统计指标 (bounty_indicator)

方案配置 模板规则下具体统计的业绩指标口径，基于不同模板规则，可以选择的业绩指标不同

一个业绩指标对应bounty_indicator一行数据

* 模板规则:

存量GMV

* 业绩指标:

请选择业绩指标

* 奖励计算时间:

实货支付金额(去优惠券去退款)

实货GMV(去退款)

* AB类型:

人均实货支付金额(去优惠券去退款)

人均实货GMV(去退款)

* 冻结销售团队标识:

提货卡口径GMV(去退款)

过滤条件:

提货卡口径支付金额(去优惠券去退款)

提货卡充值GMV(去退款)

提货卡充值支付金额(去优惠券去退款)

5) 过滤条件 (bounty_filter)

方案配置 模板规则下，具体的过滤条件，用于判断需要统计的业务数据范围，例如gmV模板，统计范围作用就是过滤统计的订单范围

过滤条件分为必填和非必填，每个过滤条件对应bounty_filter表一行数据，动态渲染前端组件

* 奖励计算时间: 2023-03-01 ~ 2023-03-31

* AB类型: ☐ A类 ☐ B类

* 冻结销售团队标识: 等于 请选择, 可输入搜索

过滤条件: 根据方案需要填写对应项 (非必填)

门店分组:	请输入搜索	
一级类目:	请选择, 可输入搜索	
二级类目:	请输入搜索	批量添加ID
品牌名称:	等于 请输入搜索	批量添加ID
商品名称:	等于 请输入商品名称/ID	批量添加ID
战区:	等于 请选择, 可输入搜索	
大区:	等于 请选择, 可输入搜索	
主管区域:	等于 请选择, 可输入搜索	
参与对象:	等于 请输入搜索	批量添加ID
不追回品牌:	等于 请输入搜索	批量添加ID

6) 提成奖励类型 (bounty_payout_rule_type)

方案配置 业绩指标下, 对应人员的指标数据能够获得的奖励配置

提成类型对应crm-master代码中 `com.yt.crm.biz.bountyplan.enums.BountyPayoutRuleTypeEnum` 枚举

提成计算规则 不知道怎么填写? 请参看《提成规则配置教程》

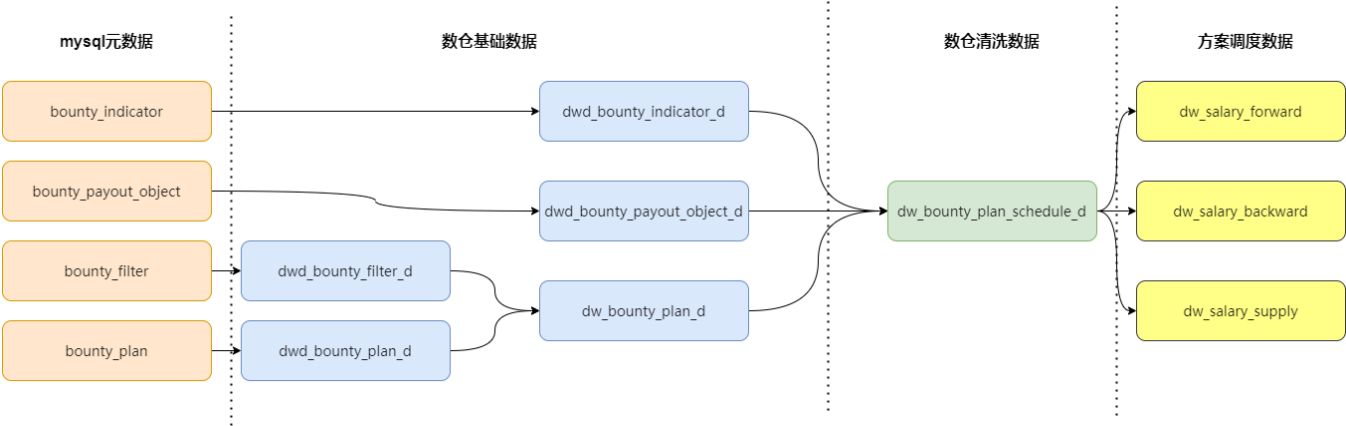
* 提成类型: 累计阶梯返现

* 提成规则: 新增行 共 2 行, 阶梯档位必须由大到小顺序填写

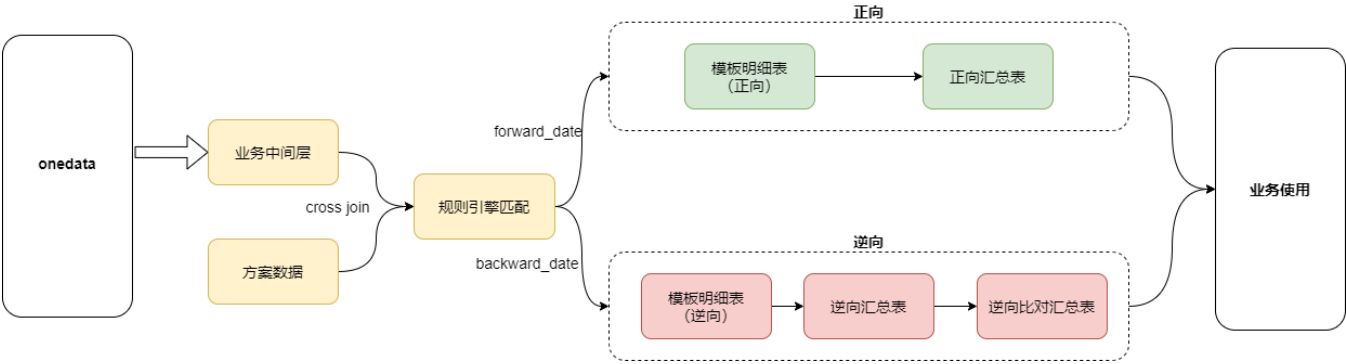
阶梯档位	返现金额(元)	操作
满 1000	返 10 元	删除 拖动排序
满 500	返 5 元	删除 拖动排序

2) 数仓调度架构模型

方案数据调度组装



调度模式



模板表

模板	正向&逆向	业务中间层	模板明细表	汇总表
*****	*****	*****	*****	*****
存量gmv	正向	dw_salary_gmv_rule_public_mid_v2_d	dw_salary_gmv_rule_public_d (分区 : pltype = 'cur')	dw_salary_forward_plan_sum_d (分区 : bounty_rule_type = 1)
	逆向		dw_salary_gmv_rule_public_d (分区 : pltype = 'pre')	dw_salary_backward_plan_sum_m (分区 : bounty_rule_type = 1)
新签商品	正向	dw_salary_sign_rule_public_mid_v2_d	dw_salary_sign_item_rule_public_d (分区 : pltype = 'cur')	dw_salary_forward_plan_sum_d (分区 : bounty_rule_type = 2)
	逆向		dw_salary_sign_item_rule_public_d (分区 : pltype = 'pre')	dw_salary_backward_plan_sum_m (分区 : bounty_rule_type = 2)
新签品牌	正向	dw_salary_sign_rule_public_mid_v2_d	dw_salary_sign_brand_rule_public_d (分区 : pltype = 'cur')	dw_salary_forward_plan_sum_d (分区 : bounty_rule_type = 3)
	逆向		dw_salary_sign_brand_rule_public_d (分区 : pltype = 'pre')	dw_salary_backward_plan_sum_m (分区 : bounty_rule_type = 3)
多品进店	正向	dw_salary_brand_shop_rule_public_mid_v2_d	dw_salary_brand_shop_sum_d (分区 : pltype = 'cur')	dw_salary_forward_plan_sum_d (分区 : bounty_rule_type = 4)
	逆向			

	逆向		dw_salary_brand_shop_sum_d (分区：pltype = 'pre')	dw_salary_backward_plan_sum_m (分区：bounty_rule_type = 4)
基本提成	正向	销售：ads_salary_result_sale_d 主管：ads_salary_result_manager_d	dw_salary_base_salary_public_d (分区：pltype = 'cur')	dw_salary_forward_plan_sum_d (分区：bounty_rule_type = 5)

调度脚本

模板 *****	正向&逆向 *****	业务中间层 *****	模板明细表 *****	汇总表 *****
存量gmv	正向	dw_salary_gmv_rule_public_mid_v2_d	dw_salary_gmv_rule_forward_public_d	dw_salary_gmv_rule_forward_j
	逆向		dw_salary_gmv_rule_backward_public_d	dw_salary_gmv_rule_backwarc
	补数		supply_gmv_rule_forward_d supply_gmv_rule_backward_d	
新签商品	正向	dw_salary_sign_rule_public_mid_v2_d	dw_salary_sign_item_forward_public_d	dw_salary_sign_item_forward_
	逆向		dw_salary_sign_item_backward_public_d	dw_salary_sign_item_backwarc
	补数		supply_sign_item_forward_d supply_sign_item_backward_d	
新签品牌	正向	dw_salary_sign_rule_public_mid_v2_d	dw_salary_sign_brand_forward_public_d	dw_salary_sign_brand_forwarc
	逆向		dw_salary_sign_brand_backward_public_d	dw_salary_sign_brand_backwa
	补数		supply_sign_brand_forward_d supply_sign_brand_backward_d	
多品进店	正向	dw_salary_brand_shop_rule_public_mid_v2_d	dw_salary_brand_shop_forward_sum_d	dw_salary_brand_shop_forwar
	逆向		dw_salary_brand_shop_backward_sum_d	dw_salary_brand_shop_backw
	补数		supply_brand_shop_forward_d supply_brand_shop_backward_d	
基本提成	正向	销售：ads_salary_result_sale_d	dw_salary_base_salary_forward_public_d	dw_salary_base_salary_forwarc
	补数	主管：ads_salary_result_manager_d	supply_base_salary_forward_d	

调度模板格式

核心问题：

- 1) 正逆向脚本 常规调度和补数调度，需要共用一套脚本（否则一套逻辑需要在多个脚本里维护，容易产生故障，造成资损）
- 2) 补数的数据，应仅产出本次补数数据，不能影响历史正逆向数据
- 3) 如果没有需要补数的数据，不应全量执行调度脚本，减少性能损耗

解决思路：

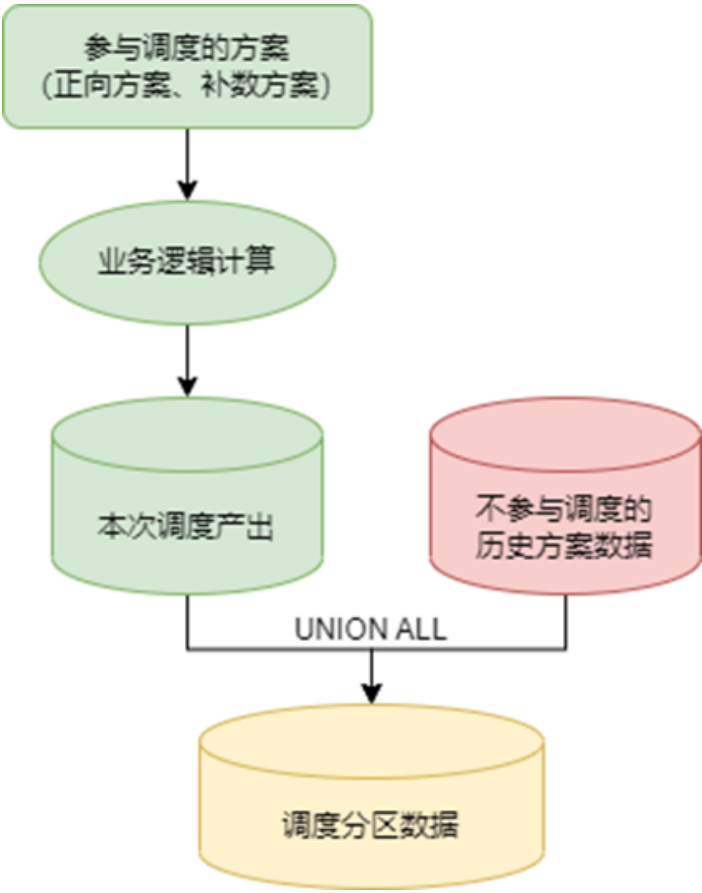
解决问题1) 通过dw_bounty_plan_schedule_d的调度日期字段（ forward_date、backward_date、supply_date ）和 shell脚本动态传参的能力解决

解决问题2) 通过脚本调度仅运算本次需要运算的方案数据，并UNION ALL历史产出不需要运算的方案数据，最后再一起insert overwrite解决

解决问题3) 通过shell脚本的临时运行变量赋值方式，先查询需要补数的方案以及补数日期，仅对这些需要补数的日期进行function调用，如果不需要补数直接exit

数据流程：

抽象流程：



补数脚本（仅有需要补数的日期时，才补数）：

```

function supply_forward() {
    supply_date=$1
    data_date=$2
    sh ./dw_salary_sign_item_rule_public_forward_d.sh $data_date 0 0 $supply_date
    sh ./dw_salary_forward_plan_sum_sign_item_rule_d.sh $data_date 0 0 $supply_date
}

IFS=";"

if [[ $forward_type = "forward" ]]
then
    forward_supply_date=$(apache-spark-sql -e "
        use ytdw;
        SELECT concat_ws('\;' , sort_array(collect_set(cast(need_supply_date as int))))
        FROM dw_bounty_plan_schedule_d
        lateral view explode(split(forward_date',')) temp as need_supply_date
        WHERE bounty_rule_type = 2
        AND supply_date = '$v_date'
        AND need_supply_date < '$v_date'
    ")
    echo forward_supply_date: $forward_supply_date
    for data_date in ${forward_supply_date[@]}
    do
        supply_forward $v_date $data_date
    done
fi

```

正逆向脚本：

```

-- 方案信息
with plan as (
    SELECT no,
           ...
    FROM dw_bounty_plan_schedule_d
    WHERE array_contains(split(forward_date, ','), '$v_date')
    AND ('$supply_mode' = 'not_supply' OR array_contains(split(supply_date, ','), '$supply_date'))
),

-- 当前脚本逻辑
cur as (
    ...
)

-- 写入逻辑
insert overwrite table ${target_table} partition (dayid='$v_date')
SELECT ...
FROM cur

UNION ALL

-- 历史数据兼容 (补数场景, 仅覆盖本次补数方案数据, 其余方案不变)
SELECT ...
FROM (
    SELECT *
    FROM ${target_table}
    WHERE dayid = '$v_date'
) history
LEFT JOIN (
    SELECT no FROM plan
) cur_plan ON history.planno = cur_plan.no
WHERE cur_plan.no is null

```

3、日常需求

2021.Q4

20211216_特殊提成七期-多品进店

- 一、背景
- 二、业务目标
- 三、产品描述
 - 数据刷新流程
- 四、名词/术语解释
- 五、用例
- 六、业务流程
- 七、领域模型
- 八、系统设计
 - 1、兼容月度、季度方案的设计：
 - 2、新增配置修改
 - 3、本期新增限制：
 - 4、系统整体架构
- 九、类图
- 十、数据库设计
- 十一、搜索设计
- 十二、接口设计
 - 1、HOP接口
- 十三、涉及数仓的设计
 - 1、edp任务
- 十四、中间件配置相关
- 十五、非功能性需求

一、背景

多品项目看板链接：[链接](#)

逻辑梳理文档: [202201 BD部 多品在店 政策](#)

[历史demo参考](#)

详细业务梳理过程请参见 逻辑梳理文档: [202201 BD部 多品在店 政策](#)

根据BD部 21年Q4下发的策略, 21年Q4至 22年Q4的 核心方向是 提升门店的B类合作尿不湿品牌数.

销售是激励导向的. 而且不可能无脑按存量发钱(已有基本提成).

因此下发的激励政策是 你多做1个品牌,我才给你钱. 之前该门店合作2个品牌.你这个Q还是2个. 不给钱.

本次项目俗称 "多品在店"

从完整的流程来讲,会拆分为

政策下发 > 指定目标门店 > 进店 > 结果 > 复盘 > 逆向追回

指定目标门店,进店和复盘 是 项目管理要做的.

结果(钱) 和 逆向追回 是 特殊提成要做的.

维度： 以门店为维度，对比该门店季度初值实货≥2000的品牌（季度内实货超过2000则算为有效品牌），以及季度末值实货≥2000的品牌，其始末变化给予积分奖励；				
前提： Q4有效下单门店数≥80%×Q3有效下单门店数（有效下单门店：有一个尿不湿品牌在Q内实货GMV≥2000）；				
积分规则：				
BD	门店	Q3内≥2000品牌	Q4内≥2000品牌	积分
张三	A	1	2	1
张三	B	2	1	-1
张三	C	2	2	0
张三	D	0	1	0
张三	E	0	2	2
张三	...			
总分		X	Y	Y-X
奖励规则： 门店季度末对比季度初每增加一个有效品牌记为1分； 特别备注： X=0,Y=1的情况下不计算积分；（举例：某门店Q3为0，Q4产生了一个有效下单品牌，此种情况下该门店不算积分，但如果Q4有2个有效下单品牌则可计入，计2分，同理产生3个则积3分；） • 10 < 积分总分 ≤ 20，每分奖励20元，奖金：积分总分 × 20元； • 20 < 积分总分 ≤ 30，每分奖励40元，奖金：积分总分 × 40元； • 30 < 积分总分 ≤ 50，每分奖励60元，奖金：积分总分 × 60元； • 50 < 积分总分 ≤ 100，每分奖励80元，奖金：积分总分 × 80元； • 积分总分 > 100，每分奖励100元，奖金：积分总分 × 100元； • 积分30分以上，且排名在前100名BD，将会受邀参加2021年年会！（如遇疫情等特殊情況，服从公司安排）。				
常见问题说明：				
➢ 必须在驻地BD的岗位工作满一个自然季度，才可获得奖励，离职亦如此； 注意！！只有在10月1日（含）前落驻地的BD，才能参加本次激励活动，且必须12月31日仍在“大门店业务部”，中间离职也会取消奖励。 比如：张三在10月2日落驻地，则其当季度无法享受本次激励政策，李四12月30日离职则亦无法享受本次激励政策；				
➢ 本激励以工资形式发放，并将会上线线上化，如后期发生退款将会从系统自动追回；				
➢ 激励周期：2021Q4——2022Q4				
➢ 品牌数在季度内去重，跨季度不去重（静默激活也算）；				
➢ 所有的数据都会上业务看板，方便大家直接查看，包括Q3的基础值，有效下单门店数，目前得分等等；				
➢ 区域变动或门店库内人员发生变更的情况：门店在张三名下时芭比船长下单2000，在李四名下时下单2000，以冻结BD先满足2000以上的算积分归属；				

二、业务目标

本次需求主要会为了满足 BD部 下发的 多品进店 政策.

主要产品功能是 1) 算正向的钱 2) 算逆向的钱 3) 透出过程数据到CRM项目管理

考虑到时间比较紧张,且 业务本期的核心诉求是为了强算逆向. 过程看数据 倒在其次. 毕竟大部分时间 积分都是负数.

所以理论上会分为两期

1) 多品在店一期：保证计算出来 正向数据 和 逆向数据. 不急着重回流CRM项目管理 . 且 CRM项目管理没有时间支持. 过程数据通过 搭建业务大宽表,让业务方用到PBI使用.

2) 多品在店二期： 过程数据在CRM项目管理 里面展示.

三、产品描述

数据刷新流程

- 1)当前方案在方案当前周期内 每天跑. 要跑得比较早. 便于 BI 抽到看板里
- 2) 在 当前周期 结果后的第一个月起,连跑12个月. 每次只跑每个月最后一天的分区.

例如, 2021年Q4的方案,则在 22年1月,2月,3月,4月到22年12月份都要重刷数据.

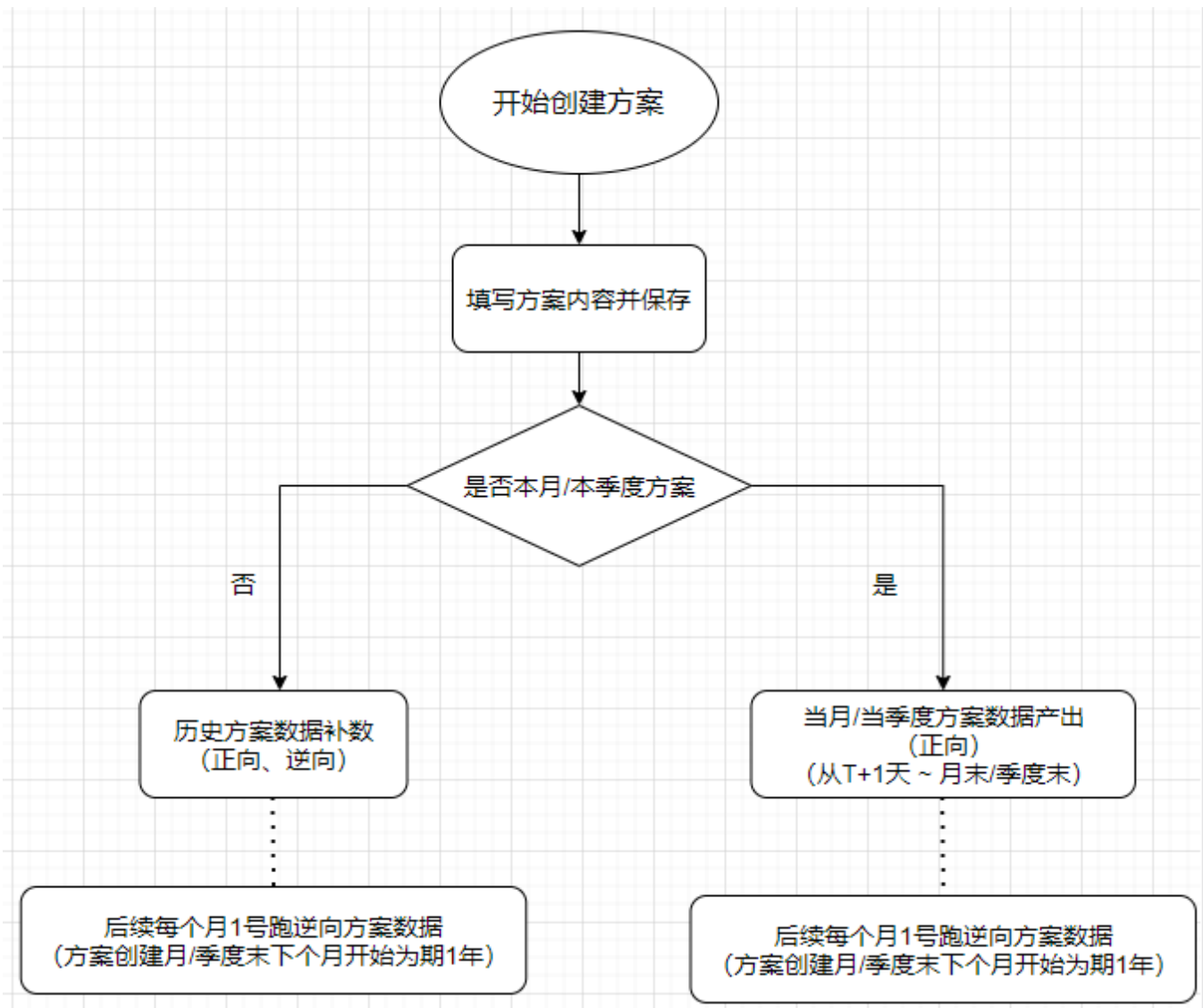
四、名词/术语解释

名词	定义
方案当前周期	指 当前方案 需要计算 有效品牌数 的时间. 一般是一个Q. 例如 2021年Q4
方案比对周期	指 当前方案 为了计算积分(有效品牌数差额) 的时候,需要比对的时间. 一般是一个Q. 如果 当前周期为 2021年Q4的方案,则其 比对周期多为 2021年Q3
{门店粒度 有效品牌门店数}	计算 比对周期和当前周期 有效品牌门店数的一个因子. 即 只要在对周期内 门店的品牌 满2000.则视作 +1 这时候再给 门店背后的 库内BD, 主管 计算其 有效品牌门店数

{冻结销售粒度 有效品牌门店数}	<p>只是计算 比对周期和当前周期 有效品牌门店数的一个因子.</p> <p>理论上只有计算BD才用到. 主管及其以上不需要用到</p> <p>即 在对应周期内,冻结BD/发放对象的 门店的品牌 累计初次满2000.则视作+1.</p> <p>可能有多个人同时做工1个品牌.按最先满足2000的来算.</p> <p>因为主管是没有冻结的概念.永远都是库内. 所以主管的 冻结销售粒度 有效品牌门店数 恒等于 门店粒度 有效品牌门店数</p> <p>也是本期方案能通过 发放对象自由切换,又适用有一定差异的 BD/主管 方案的核心原因.</p>
a) 比对周期的 库内 发放对象 的 {门店粒度 有效品牌门店数}	指当前统计对象在 当前周期的库内门店 在 比对周期 的 {门店粒度 有效品牌门店数}
b) 比对周期的 冻结BD 有归属品牌的 {门店粒度 有效品牌门店数}	指当前统计对象在 当前周期的非库内门店 但在当前周期归属过有品牌门店数给该BD的 门店 在 比对周期的 {门店粒度 有效品牌门店数} .
比对周期 有效品牌门店数	<p>以下两个相加</p> <p>1) a) 比对周期的 库内 发放对象 的 {门店粒度 有效品牌门店数}</p> <p>2) b) 比对周期的 冻结BD 有归属品牌的 {门店粒度 有效品牌门店数}</p> <p>如果是 主管及其以上,则期初里面没有 2) 这部分</p>
当前周期 的有效品牌门店数	当前周期内的 {冻结销售粒度 有效品牌门店数}
积分	是以一定的逻辑将 发放对象的门店粒度的 有效品牌门店数 转化为 积分.

五、用例

六、业务流程



七、领域模型

八、系统设计

1、兼容月度、季度方案的设计：

新增UDF plan_schedule_time (String month, String createTime, Integer timeType, Integer resultType)

month: 方案月份

createTime: 方案创建时间

timeType: 方案时间类型 (月度方案还是季度方案)

resultType: 返回结果类型, 1、需要跑正向的日期 2、需要跑逆向的日期 3、需要补数的日期

并在数仓提前根据dw_bounty_plan_d表数据, 加工出一张方案调度表, 用于判断哪些方案需要在对应脚本里面进行数据产出, 统一维护, 并减少在数据产出脚本中冗余、重复的判断逻辑

2、新增配置修改

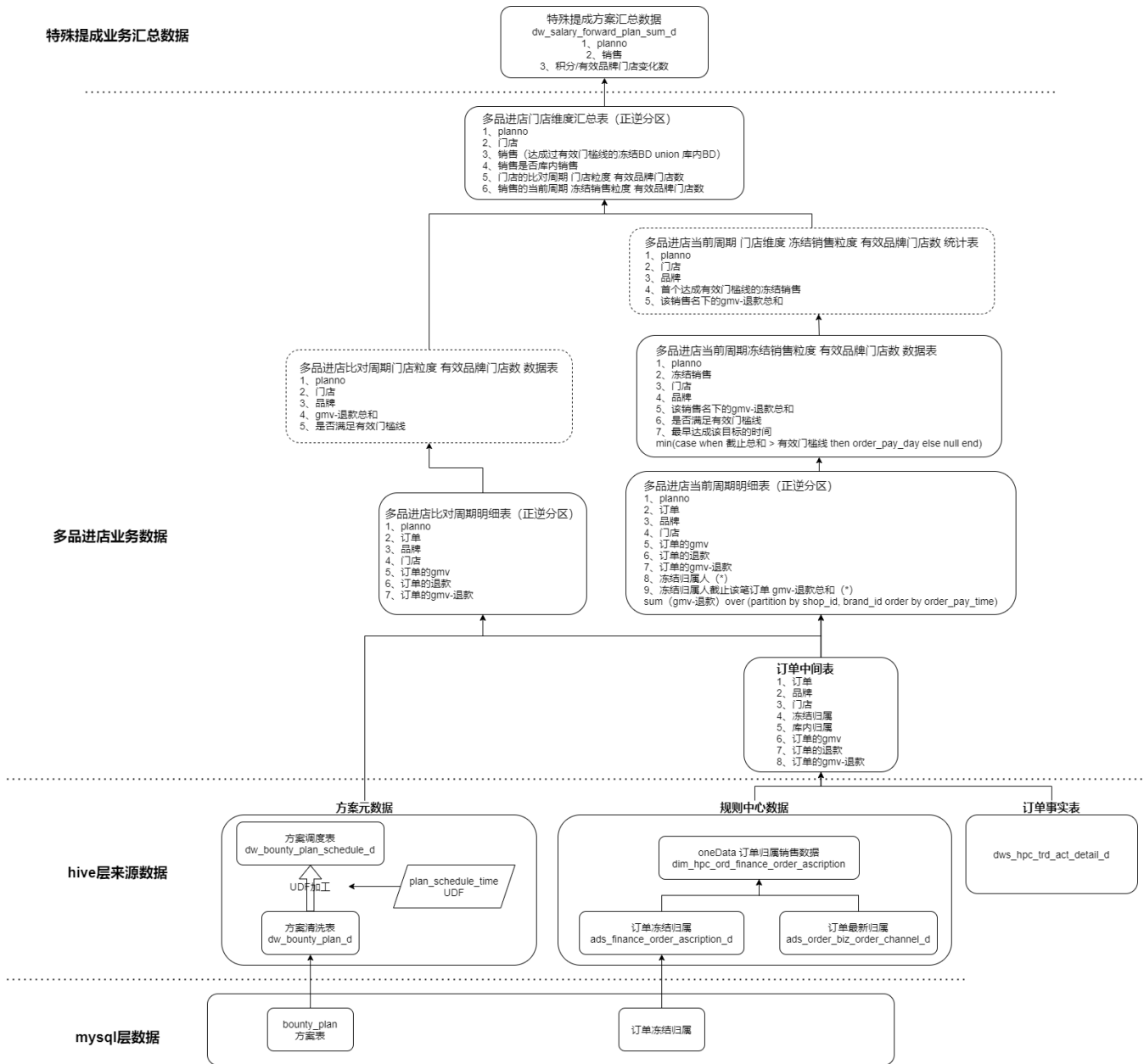
1. 方案主表 : bounty_plan

- a. 新增方案时间月份字段：time_type，1表示月度方案，2表示季度方案，历史数据默认为1；当方案时间类型为季度方案时，bounty_plan表time_type = 2，且month为季度的第一个月
- 2. 规则模板枚举：BountyRuleTypeEnum
 - a. 新增品牌进店模板
- 3. 发放对象—规则模板映射表：bounty_payout_object_rule
 - a. 新增映射，（战区经理，大区经理，BD主管，BD）→ 多品进店规则模板
- 4. 业绩指标表：bounty_indicator
 - a. 新增业绩指标，有效品牌门店数、多品在店积分、有效品牌门店数变化
- 5. 过滤条件表：bounty_filter
 - a. 新增过滤条件，有效门槛线，方案计算时间（跨月版），比对时间
- 6. 模板过滤条件映射表：bounty_rule_filter
 - a. 新增模板→过滤条件映射，品牌进店模板 →（AB类型，销售团队标识，门店分组，一级类目，二级类目，品牌名，战区，大区，主管区域，有效门槛线，方案计算时间（跨月版），比对时间）
- 7. 业绩指标提成类型映射表
 - a. 新增业绩指标→提成类型映射，（有效品牌门店数、多品在店积分、有效品牌门店数变化）→（累计阶梯单价返点，累计阶梯返现，累计阶梯反实物，排名返现）

3、本期新增限制：

- 1. 之前遗漏的，校验门店分组只能选方案月份及其创建的
- 2. p0项目页面不支持创建多品进店类型方案
- 3. 存量gmv、新签商品、新签品牌模板不支持创建季度方案

4、系统整体架构



- 九、类图
- 十、数据库设计
- 十一、搜索设计
- 十二、接口设计
- 1、HOP接口

接口功能描述	接口名称	变更类型（新增/更新）	原云图接口
*****	*****	*****	*****
查询发放对象	hipac.crm.bountyPlan.listPayoutObject	迁移	hipac.cloudatlas.bountyPlan.listPayoutObject
废弃方案	hipac.crm.bountyPlan.discard	迁移	hipac.cloudatlas.bountyPlan.discard

保存方案	hipac.crm.bountyPlan.save	迁移	hipac.cloudatlas.bountyPlan.save
查看方案详情	hipac.crm.bountyPlan.viewDetail	迁移	hipac.cloudatlas.bountyPlan.viewDetail
分页查询方案列表	hipac.crm.bountyPlan.listByPage	迁移	hipac.cloudatlas.bountyPlan.listByPage
查询方案变更记录	hipac.crm.bountyPlan.listOperationRecord	迁移	hipac.cloudatlas.bountyPlan.listOperationRecord
查询方案销售区域	hipac.crm.bountyPlan.listSalesArea	迁移	hipac.cloudatlas.bountyPlan.listSalesArea
查询提成奖励类型	hipac.crm.bountyPlan.listPayoutRuleType	迁移	hipac.cloudatlas.bountyPlan.listPayoutRuleType
查询过滤条件	hipac.crm.bountyPlan.listFilter	迁移	hipac.cloudatlas.bountyPlan.listFilter
查询业绩指标	hipac.crm.bountyPlan.listIndicator	迁移	hipac.cloudatlas.bountyPlan.listIndicator
查询规则模板	hipac.crm.bountyPlan.listRuleTypes	迁移	hipac.cloudatlas.bountyPlan.listRuleTypes

十三、涉及数仓的设计

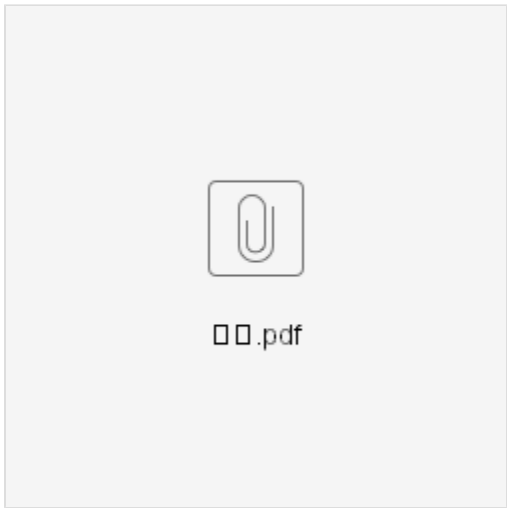
1、edp任务

项目名称	项目地址	任务名称	调度时间	变更类型 (新增/更新)	变更描述

十四、中间件配置相关

十五、非功能性需求

项目排期：



2022.Q2

特殊提成8期

需求wiki：[202203_03_日常_特殊提成8期小日常](#)

排期：

[illegible]

前端交互接口变更

需求二变更：

二、新增提成类型为“排名返现”时的指标门槛

需求背景

例如,只有下单超过1万时,才会参与排名。

目前会经常出现一群下单一两千的销售在PK排名。

需求内容

1. 新建和编辑方案时，下面位置新增输入框：指标门槛。样式：必填项+数字输入框，保证可输入小数、负数，并且基于上面选中的指标单位，展示其指标单位。
2. 修改：基于排名计算奖金的逻辑：参见 01_当月方案汇总表 的V4改动
3. 需求评审后补充-- 补充 新签品牌 时，按 门店数相关指标排序时，加上 GMV 排序的逻辑

1. 系统计算规则：不知道怎么办吗？请参考《系统规则及配套附件》

* 请选地址：

请选门诊号：

* 请选时间： 注：请选就诊时间的大于入院申请时间

药品名称	远视剂量 (元)	操作
硝	或	或
或	或	或
或	或	或

1. 系统设置：请选门诊号
 若患者上一步未选地址或选地址未选时间：请选地址
 若未选地址，但选了时间且未选地址或选了地址但未选时间：请选时间；若 1. 请选地址和2. 1. 请选时间均选了时间

病种ID	病种名称(主病种)	DRG	等级	医保类别
001	15	275	1	1000
001	19	1402	2	1000
002	15	1835	2	1000
004	2	402	4	0
005	1	800	5	0
006	6	300	4	0

需求：

变更：当 `hipac.crm.bountyPlan.listPayoutRuleType` 接口，`payoutRuleTypeCode = 5` 且 `isUpperLimit = true` 时，新增显示指标门槛，单位为指标的单位，保存时复用原先的 `payoutUpperLimit` 字段

需求三变更：

纯前端变更

具体里面的数据时间都放大

例如, 2个月的方案, 如果是从 1月-2月, 这种就不能用季度方案, 因为 2月底就要结算, 走月度方案, 这个问题在于 1月份的额度不支持, 可能要配置2次。
如果是2月-3月的, 可以选择 季度方案。

2. 存量GMV 新签品牌、新签商品层是数据支持到3个月的明细, 商品在店底层数据支持6个月明细。

[illegible]

需求：

四、修改特殊提成配置列表页

需求内容

1. 新增 方案时间类型筛选器, 单选, 筛选值: 月度方案/季度方案
2. 修改 表单中的 方案月份, 修改成 方案时间, 如果是季度方案, 则展示季度, 如果是月度方案, 则展示月度.
3. 怎么生效 方案月份/方案季度的筛选逻辑??? 可能方案: 当筛选月份后, 展示的方案为 1) 方案时间类型为 月份方案 且 方案时间为当月 2) 方案时间类型为 季度方案 且 方案季度包含当月

[illegible]

变更：页面查询接口，新增入参timeType，表示方案时间类型；返回结果现有字段timeType和month，前端显示时转换一下，如果是月度方案保持不变，如果是季度方案显示季度

方案配置

方案月份: 请选择月份

归属业务组: 请选择

方案状态: 正常

方案归属类型: 请选择

方案名称: 请输入

创建时间: 开始日期 - 结束日期

方案编号: 请输入方案编号查找

创建人: 请输入

最近修改时间: 开始日期 - 结束日期

新建方案

Filter
Only show requests with SameSite issues

20 ms
40 ms
60 ms
80 ms
100 ms
120 ms
140 ms
160 ms

Name

[-] httpac.crm.bountyPlan.listByPage

X Headers Preview Response Initiator Timing Cookies

```

{url: "httpac.crm.bountyPlan.listByPage", "v": "1.0.0", "code: 200, messa
v": "1.0.0"
code: 200
message: ""
}
[detail: ...]
*0: {editorName: null, ownerType: 2, mo: "11480", isBountyRuleType:
editorName: null
ownerType: 2
mo: "11480"
isBountyRuleType: false
subjectRuleType: none; 4
subjectName: "环境"
isBountyRuleType: 5
bizRuleName: "11321"
subjectId: 6024
indicatorId: 3
}
name: "环境"
createTime: "2022-02-11 18:15:56"
name: "方案名称-master环境方案"
id: 11480
lastEditTime: null
subjectName: "测试环境项目"
status: 1
*1: {editorName: "环境", ownerType: 2, mo: "11480", isBountyRuleType
*2: {editorName: null, ownerType: 1, mo: "11484", isBountyRuleType
*3: {editorName: null, ownerType: 2, mo: "11483", isBountyRuleType
*4: {editorName: null, ownerType: 2, mo: "11483", isBountyRuleType
*5: {editorName: null, ownerType: 1, mo: "11481", isBountyRuleType
*6: {editorName: null, ownerType: 1, mo: "11480", isBountyRuleType
*7: {editorName: "环境", ownerType: 1, mo: "11479", isBountyRuleType
*8: {editorName: null, ownerType: 1, mo: "11478", isBountyRuleType
*9: {editorName: null, ownerType: 2, mo: "11474", isBountyRuleType
pageInfo: 0
totalCount: 284
totalPage: 28
onPageClick: ""
hasList: false
monitorInfo: {"pageId": "hap-master-deploy-669541648539932613"
success: true

```

方案编号	方案归属类型	方案名称	关联项目	方案月份	归属业务组	方案状态	创建人	创建时间	最近操作人及时间	操作
11488	销售部署方案	master环境方案	测试项目	2022-01	21321	正常	孙国	02-11 18:15:56	郑城	查看 修改
11487	销售部署方案	测试项目	测试项目	2022-02	20220209	正常	郑城	02-09 15:44:25	郑城	查看 修改 复制
11484	业务部署方案	测试项目	测试项目	2021-10	21321	正常	孙国	01-10 19:25:22	郑城	查看 修改
11483	销售部署方案	master环境方案	测试项目	2022-01	21321	正常	郑城	01-07 10:23:44	郑城	查看 修改
114	销售部署方案	测试项目	测试项目	2022-01	21321	正常	郑城	02-07 10:23:44	郑城	查看 修改

特殊提成9期

数仓逻辑修改梳理

1、新增指定受益人、参与人员

老模板

dw_salary_gmv_rule_public_forward_d 存量GMV

dw_salary_gmv_rule_public_backward_d 存量GMV

dw_salary_sign_brand_rule_public_backward_d

dw_salary_sign_brand_rule_public_forward_d

dw_salary_sign_item_rule_public_backward_d

dw_salary_sign_item_rule_public_forward_d

plan.bounty_payout_object_code,	208	214	case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_id	✓
case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_id	209	215	when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_id	
when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_id	210	216	when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_id	
when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_id	211	217	when plan.bounty_payout_object_code in('BD','BIG_BD') then service_user_id_frozen	
when plan.bounty_payout_object_code in('BD','BIG_BD') then service_user_id_frozen	212	218	when plan.bounty_payout_object_code = 'GRANT_USER' then plan.grant_user	
end as grant_object_user_id,	213	219	end as grant_object_user_id,	--发放
case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_name	214	220	case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_name	
when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_name	215	221	when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_name	
when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_name	216	222	when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_name	
when plan.bounty_payout_object_code in('BD','BIG_BD') then service_user_name_frozen	217	223	when plan.bounty_payout_object_code in('BD','BIG_BD') then service_user_name_frozen	
end as grant_object_user_name,	218	224	when plan.bounty_payout_object_code = 'GRANT_USER' then null	
case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_dep_id	219	225	end as grant_object_user_name,	--2
when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_dep_id	220	226	case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_dep_id	
when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_dep_id	221	227	when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_dep_id	
when plan.bounty_payout_object_code in('BD','BIG_BD') then service_department_id_frozen	222	228	when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_dep_id	
end as grant_object_user_dep_id,	223	229	when plan.bounty_payout_object_code in('BD','BIG_BD') then service_department_id_frozen	
case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_dep_name	224	230	when plan.bounty_payout_object_code = 'GRANT_USER' then null	
when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_dep_name	225	231	end as grant_object_user_dep_id,	
when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_dep_name	226	232	case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_dep_name	
when plan.bounty_payout_object_code in('BD','BIG_BD') then service_department_name_frozen	227	233	when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_dep_name	
end as grant_object_user_dep_name	228	234	when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_dep_name	
FROM sign	229	235	when plan.bounty_payout_object_code in('BD','BIG_BD') then service_department_name_frozen	
INNER JOIN plan ON sign.planno = plan.no	230	236	when plan.bounty_payout_object_code = 'GRANT_USER' then null	
)	231	237	end as grant_object_user_dep_name,	
insert overwrite table dw_salary_sign_brand_rule_public_d partition (dayid='\${v_date}',pltype='cur')	232	238		
SELECT update_time,	233	239	plan.filter_user_value,	
update_month,	234	240	plan.filter_user_operator	
plan_type,	235	241	FROM sign	
plan_month,	236	242	INNER JOIN plan ON sign.planno = plan.no	参与人员过滤
plan_pay_time,	237	243	HAVING ytdw.simple_expr(grant_object_user_id, 'in', filter_user_value) = (case when filter_user_operator = '=' then	
	238	244		

多品模板

dw_salary_forward_brand_shop_current_public_d

dw_salary_backward_brand_shop_current_public_d

ord.category_1st_id,	63	64	ord.category_1st_id,	
ord.category_1st_name,	64	65	ord.category_1st_name,	
ord.category_2nd_id,	65	66	ord.category_2nd_id,	
ord.category_2nd_name,	66	67	ord.category_2nd_name,	
ord.item_id,	67	68	ord.item_id,	
ord.item_name,	68	69	ord.item_name,	
ord.item_style,	69	70	ord.item_style,	
ord.item_style_name,	70	71	ord.item_style_name,	
case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_id	71	72	case when plan.bounty_payout_object_code = 'WAR_ZONE_MANAGER' then war_zone_id	
when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_id	72	73	when plan.bounty_payout_object_code = 'AREA_MANAGER' then area_manager_id	
when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_id	73	74	when plan.bounty_payout_object_code = 'BD_MANAGER' then bd_manager_id	
when plan.bounty_payout_object_code = 'BD' then if(plan.payout_object_type = '冻结', ord.frozen_sale_u	74	75	when plan.bounty_payout_object_code = 'BD' then if(plan.payout_object_type = '冻结', ord.frozen_sale_u	
end as grant_object_user_id	75	76	when plan.bounty_payout_object_code = 'GRANT_USER' then plan.grant_user	
FROM plan	76	77	end as grant_object_user_id	指定受益人逻辑
CROSS JOIN (select * from dw_salary_brand_shop_rule_public_mid_v2_d where dayid = '\${v_date}') ord ON 1=1	77	78	FROM plan	
WHERE ord.pay_date between plan.calculate_date_value_start and plan.calculate_date_value_end	78	79	CROSS JOIN (select * from dw_salary_brand_shop_rule_public_mid_v2_d where dayid = '\${v_date}') ord ON 1=1	
AND ytdw.simple_expr(ord.item_style_name, 'in', plan.item_style_value) = if(plan.item_style_operator = '=', 1, 0)	79	80	WHERE ord.pay_date between plan.calculate_date_value_start and plan.calculate_date_value_end	
AND ytdw.simple_expr(ord.category_1st_id, 'in', plan.category_first_value) = if(plan.category_first_operator = '=', 1, 0)	80	81	AND ytdw.simple_expr(ord.item_style_name, 'in', plan.item_style_value) = if(plan.item_style_operator = '=', 1, 0)	
AND ytdw.simple_expr(ord.category_2nd_id, 'in', plan.category_second_value) = if(plan.category_second_operator = '=', 1, 0)	81	82	AND ytdw.simple_expr(ord.category_1st_id, 'in', plan.category_first_value) = if(plan.category_first_operator = '=', 1, 0)	
AND ytdw.simple_expr(ord.brand_id, 'in', plan.brand_value) = if(plan.brand_operator = '=', 1, 0)	82	83	AND ytdw.simple_expr(ord.category_2nd_id, 'in', plan.category_second_value) = if(plan.category_second_operator = '=', 1, 0)	
AND ytdw.simple_expr(ord.war_zone_dep_id, 'in', plan.war_area_value) = if(plan.war_area_operator = '=', 1, 0)	83	84	AND ytdw.simple_expr(ord.brand_id, 'in', plan.brand_value) = if(plan.brand_operator = '=', 1, 0)	
AND ytdw.simple_expr(ord.area_manager_dep_id, 'in', plan.bd_area_value) = if(plan.bd_area_operator = '=', 1, 0)	84	85	AND ytdw.simple_expr(ord.war_zone_dep_id, 'in', plan.war_area_value) = if(plan.war_area_operator = '=', 1, 0)	
AND ytdw.simple_expr(ord.bd_manager_dep_id, 'in', plan.manage_area_value) = if(plan.manage_area_operator = '=', 1, 0)	85	86	AND ytdw.simple_expr(ord.area_manager_dep_id, 'in', plan.bd_area_value) = if(plan.bd_area_operator = '=', 1, 0)	
AND ytdw.simple_expr(if(plan.payout_object_type = '冻结', ord.sale_team_frozen_id, ord.sale_team_id), 'in', plan	86	87	AND ytdw.simple_expr(ord.bd_manager_dep_id, 'in', plan.manage_area_value) = if(plan.manage_area_operator = '=', 1, 0)	
AND if(ord.shop_group = '' OR plan.shop_group_value = '', 0, ytdw.simple_expr(substr(plan.shop_group_value, 2, 1e	87	88	AND ytdw.simple_expr(if(plan.payout_object_type = '冻结', ord.sale_team_frozen_id, ord.sale_team_id), 'in', plan	
GROUP BY plan.no,	88	89	AND if(ord.shop_group = '' OR plan.shop_group_value = '', 0, ytdw.simple_expr(substr(plan.shop_group_value, 2, 1e	
plan.month,	89	90	CROSS BY plan.no,	

dw_salary_brand_shop_sum_d

```
运行 ah (C:\pack\peach1\使用的工具\特制启动3.4. 模拟设备版本3.4. 多品出店系列数6. dw_salary_brand_shop_sum_d)
Side-by-side viewer | Do not ignore | Highlight words | 6 differences

--发政对象为库内的人员, 应当同时发政对象即可
kn_plan_user as (
  SELECT distinct planno, shop_id, grant_object_user_id, is_kn_sale_user
  FROM (
    SELECT distinct current_data.planno,
    current_data.shop_id,
    current_data.grant_object_user_id,
    '库内' as is_kn_sale_user
  FROM current_data
  INNER JOIN plan ON current_data.planno = plan.no AND plan.payout_object_type = '库内'
  UNION ALL
  SELECT distinct plan.no as planno,
  shop.shop_id,
  case when plan.bounty_payout_object_code= 'MAR_ZONE_MANAGER' then shop_service.shop_theater_ma
  when plan.bounty_payout_object_code= 'AREA_MANAGER' then shop_service.shop_region_manage
  when plan.bounty_payout_object_code= 'BD_MANAGER' then shop_service.shop_supervisor_id
  when plan.bounty_payout_object_code= 'BD' then shop_service.shop_service_id
  end as grant_object_user_id,
  '库内' as is_kn_sale_user
  FROM shop
  INNER JOIN plan ON shop.planno = plan.no AND plan.payout_object_type = '库内'
  LEFT JOIN shop_service ON shop.shop_id = shop_service.shop_id AND plan.plan_date = shop_service.dayid
  ) t
), t
--人员离职状态
user_admin as (
  SELECT user_id,
  dismiss_status,
  leave_time,
  start_time,
  end_time
  FROM user_admin
  WHERE dismiss_status = 0
  ) t
), t
--发政对象为库内的人员, 应当同时发政对象即可
kn_plan_user as (
  SELECT distinct planno, shop_id, grant_object_user_id, is_kn_sale_user
  FROM (
    SELECT distinct current_data.planno,
    current_data.shop_id,
    current_data.grant_object_user_id,
    '库内' as is_kn_sale_user
  FROM current_data
  INNER JOIN plan ON current_data.planno = plan.no AND plan.payout_object_type = '库内'
  UNION ALL
  SELECT distinct plan.no as planno,
  shop.shop_id,
  case when plan.bounty_payout_object_code= 'MAR_ZONE_MANAGER' then shop_service.shop_theater_manage
  when plan.bounty_payout_object_code= 'AREA_MANAGER' then shop_service.shop_region_manager_id
  when plan.bounty_payout_object_code= 'BD_MANAGER' then shop_service.shop_supervisor_id
  when plan.bounty_payout_object_code= 'BD' then shop_service.shop_service_id
  end as grant_object_user_id,
  '库内' as is_kn_sale_user
  FROM shop
  INNER JOIN plan ON shop.planno = plan.no AND plan.payout_object_type = '库内'
  LEFT JOIN shop_service ON shop.shop_id = shop_service.shop_id AND plan.plan_date = shop_service.dayid
  ) t
), t
--人员离职状态
user_admin as (
  SELECT user_id,
  dismiss_status,
  leave_time,
  start_time,
  end_time
  FROM user_admin
  WHERE dismiss_status = 0
  ) t
), t
```

```
shop.shop_name,
plan_user.grant_object_user_id,
plan_user.is_kn_sale_user,
if(user_admin.dismiss_status = 0, '否', '是') as is_leave,
user_admin.leave_time,
if(user_admin.dismiss_status = 0, ifnull(count(distinct compare_data.brand_id), 0), 0) as compare_brand_shop,
if(user_admin.dismiss_status = 0, ifnull(count(distinct current_data.brand_id), 0), 0) as current_brand_shop,
sum(current_data.total_gmv_less_refund) as total_gmv_less_refund
FROM plan
INNER JOIN shop ON plan.no = shop.planno
INNER JOIN (
  SELECT planno, shop_id, grant_object_user_id, is_kn_sale_user FROM frozen_plan_user
  UNION ALL
  SELECT planno, shop_id, grant_object_user_id, is_kn_sale_user FROM kn_plan_user
) plan_user ON shop.planno = plan_user.planno AND shop.shop_id = plan_user.shop_id
LEFT JOIN user_admin ON user_admin.user_id = plan_user.grant_object_user_id AND user_admin.start_time <= concat(p
LEFT JOIN compare_data ON plan_user.planno = compare_data.planno AND compare_data.shop_id = plan_user.shop_id
LEFT JOIN current_data ON plan_user.planno = current_data.planno AND current_data.shop_id = plan_user.shop_id
GROUP BY plan.no,
plan.month,
shop.shop_id,
shop.shop_name,
plan_user.grant_object_user_id,
plan_user.is_kn_sale_user,
user_admin.dismiss_status,
user_admin.leave_time
)
insert overwrite table dw_salary_brand_shop_sum_d partition (dayid='${v_date}', ptype='${p_type}')
SELECT planno,
plan_month,
plan_shop_name,
plan_user.grant_object_user_id,
plan_user.is_kn_sale_user,
if(user_admin.dismiss_status = 0, '否', '是') as is_leave,
user_admin.leave_time,
if(user_admin.dismiss_status = 0, ifnull(count(distinct compare_data.brand_id), 0), 0) as compare_brand_shop,
if(user_admin.dismiss_status = 0, ifnull(count(distinct current_data.brand_id), 0), 0) as current_brand_shop,
sum(current_data.total_gmv_less_refund) as total_gmv_less_refund
FROM plan
INNER JOIN shop ON plan.no = shop.planno
INNER JOIN (
  SELECT planno, shop_id, grant_object_user_id, is_kn_sale_user FROM frozen_plan_user
  UNION ALL
  SELECT planno, shop_id, grant_object_user_id, is_kn_sale_user FROM kn_plan_user
) plan_user ON shop.planno = plan_user.planno AND shop.shop_id = plan_user.shop_id
LEFT JOIN user_admin ON user_admin.user_id = plan_user.grant_object_user_id AND user_admin.start_time <= concat(plan
LEFT JOIN compare_data ON plan_user.planno = compare_data.planno AND compare_data.shop_id = plan_user.shop_id
LEFT JOIN current_data ON plan_user.planno = current_data.planno AND current_data.shop_id = plan_user.shop_id AND cu
GROUP BY plan.no,
plan.month,
shop.shop_id,
shop.shop_name,
plan_user.grant_object_user_id,
plan_user.is_kn_sale_user,
user_admin.dismiss_status,
user_admin.leave_time,
plan.filter_user_value,
plan.filter_user_operator
)
HAVING ytdw.simple_expr(grant_object_user_id, 'in', filter_user_value) = (case when filter_user_operator = '=' then
)
```

2、gmV模板新增指标

dw_salary_gmv_rule_public_mid_v2_d

```
insert overwrite table dw_salary_gmv_rule_public_mid_v2_d partition(dayid='${v_date}')
select business_unit, --业务线
category_id_first, --商品一级类目,
category_id_second, --商品二级类目,
category_id_first_name,
category_id_second_name,
brand_id,
brand_name, --商品品牌,
item_id,
item_name, --商品名称,
item_style,
case when order.item_style=0 then 'A' when order.item_style=1 then 'B' else '其他' end as item_style_name, --ab
case when order.sp_id is not null then '是' else '否' end as is_sp_shop, --是否服务商订单
case when ytdw.get_service_info('service_feature_id:4', service_info_freezed, service_user_id) is not null then
case when spec_order.trade_no is not null then '是' else '否' end as is_spec_order, --是否特殊订单,
order.shop_id,
shop_name, --门店名称,
item_performance_category_1st_id, --业绩一级类目,
item_performance_category_2nd_id, --业绩二级类目,
item_performance_category_1st_name,
item_performance_category_2nd_name,
item.brand_id,
item.brand_name, --业绩品牌,
order.item_id,
item_name, --商品名称,
item_style,
case when order.item_style=0 then 'A' when order.item_style=1 then 'B' else '其他' end as item_style_name, --
case when order.sp_id is not null then '是' else '否' end as is_sp_shop, --是否服务商订单
case when ytdw.get_service_info('service_feature_id:4', service_info_freezed, service_user_id) is not null then
case when spec_order.trade_no is not null then '是' else '否' end as is_spec_order, --是否特殊订单,
order.shop_id,
shop_name, --门店名称,
```

order.order_id,	114	123	order.order_id,
ord_seller.sale_team_name,	115	124	ord_seller.sale_team_name,
ord_seller.sale_team_frozen_name,	116	125	ord_seller.sale_team_frozen_name,
ord_seller.sale_team_id,	117	126	ord_seller.sale_team_id,
ord_seller.sale_team_frozen_id,	118	127	ord_seller.sale_team_frozen_id,
shop_group.mapping_group_id as shop_group	119	128	shop_group.mapping_group_id as shop_group,
--订单表	120	129	
from (121	130	--提货卡口径指标
select *	122	131	if(order.is_pickup_recharge_order = 1 OR business_unit not in ('卡券票','其他'), order.total_pay_amount - nvl(ord
from dw_order_d	123	132	if(order.is_pickup_recharge_order = 1 OR business_unit not in ('卡券票','其他'), order.total_pay_amount - nvl(ord
where dayid = '\$v_date'	124	133	if(order.is_pickup_recharge_order = 1 OR business_unit not in ('卡券票','其他'), order.pay_amount - nvl(order,
--保留近100天数据	125	134	if(order.is_pickup_recharge_order = 1 OR business_unit not in ('卡券票','其他'), order.pay_amount - nvl(order,pic
and substr(pay_time,1,8) >= replace(date_add('\$v_op_time', -100), '-', '')	126	135	
and bu_id=0	127	136	--提货卡充值指标
) order	128	137	if(is_pickup_recharge_order = 1, order.total_pay_amount - nvl(refund.refund_actual_amount, 0), 0) as pickup_rech
--退款表	129	138	if(is_pickup_recharge_order = 1, order.total_pay_amount, 0) as pickup_recharge_gmv, --提货卡充值gmv
left join (130	139	if(is_pickup_recharge_order = 1, order.pay_amount, 0) as pickup_recharge_pay_amount, --提货卡充值支付金额
select order_id,	131	140	if(is_pickup_recharge_order = 1, order.pay_amount - nvl(refund.refund_actual_amount, 0), 0) as pickup_recharge_p
sum(refund_actual_amount) as refund_actual_amount,	132	141	
sum(case when multiple_refund=10 then refund_actual_amount else 0 end) as refund_retreat_amount	133	142	--订单表
from dw_efs_order_refund_new_d	134	143	from (
where dayid = '\$v_date'	135	144	select *
and refund_status=9	136	145	from dw_order_d
group by order_id	137	146	where dayid = '\$v_date'
) refund on order.order_id=refund.order_id	138	147	--保留近100天数据
	139	148	and substr(pay_time,1,8) >= replace(date_add('\$v_op_time', -100), '-', '')
	140	149	and bu_id=0

dw_salary_gmv_rule_public_forward_d

dw_salary_gmv_rule_public_backward_d

when plan.bounty_payout_object_code = 'GRANT_USER' then null	251	262	--统计指标----
end as grant_object_user_dep_name,	252	263	plan.bounty_indicator_name as sts_target_name,
	253	264	case when plan.bounty_indicator_code in ('STOCK_GMV_1_GOODS_GMV_MINUS_REFUND', 'STOCK_GMV_AVG_GOODS_GMV_MINI
---统计指标----	254	265	when plan.bounty_indicator_code in ('STOCK_GMV_1_GOODS_PAY_AMT_MINUS_COUPONS_MINUS_REF', 'STOCK_GMV_AV
plan.bounty_indicator_name as sts_target_name,	255	266	when plan.bounty_indicator_code in ('PICKUP_PAY_GMV_LESS_REFUND', 'AVG_PICKUP_PAY_GMV_LESS_REFUND') the
case when plan.bounty_indicator_code in ('STOCK_GMV_1_GOODS_GMV_MINUS_REFUND', 'STOCK_GMV_AVG_GOODS_GMV_MI	256	267	when plan.bounty_indicator_code in ('PICKUP_PAY_AMT_LESS_COUPON_REFUND', 'AVG_PICKUP_PAY_AMT_LESS_COUPO
when plan.bounty_indicator_code = 'STOCK_GMV_1_GOODS_GMV' then ord.gmv --实收GMV	257	268	when plan.bounty_indicator_code in ('PICKUP_RECHARGE_GMV_LESS_REFUND', 'AVG_PICKUP_RECHARGE_GMV_LESS_RE
when plan.bounty_indicator_code in ('STOCK_GMV_1_GOODS_PAY_AMT_MINUS_COUPONS_MINUS_REF', 'STOCK_GMV_	258	269	when plan.bounty_indicator_code in ('PICKUP_RECHARGE_AMT_LESS_COUPON_REFUND', 'AVG_PICKUP_RECHARGE_AMT
end as sts_target,	259	270	end as sts_target,
	260	271	
ord.pay_day,	261	272	ord.pay_day,
plan.no as planno,	262	273	plan.no as planno,
	263	274	
plan.filter_user_value,	264	275	plan.filter_user_value,
plan.filter_user_operator	265	276	plan.filter_user_operator,
	266	277	
FROM plan	267	278	
CROSS JOIN ord ON 1 = 1	268	279	--冗余字段
where ord.pay_day between calculate_date_value_start and calculate_date_value_end	269	280	to_json(named_struct(
and ytdw.simple_expr(ord.sale_team_frozen_id, 'in', freeze_sales_team_value) = (case when freeze_sales_team_oper	270	281	'gmv_less_refund', ord.gmv_less_refund,
and ytdw.simple_expr(item_style_name, 'in', item_style_value) = (case when item_style_operator = '=' then 1 else 0	271	282	'gmv', ord.gmv,
and ytdw.simple_expr(category_id_first, 'in', category_first_value) = (case when category_first_operator = '=' th	272	283	'pay_amount', ord.pay_amount,
and ytdw.simple_expr(category_id_second, 'in', category_second_value) = (case when category_second_operator = '='	273	284	'pay_amount_less_refund', ord.pay_amount_less_refund,
and ytdw.simple_expr(brand_id, 'in', brand_value) = (case when brand_operator = '=' then 1 else 0 end)	274	285	'refund_actual_amount', ord.refund_actual_amount,
and ytdw.simple_expr(item_id, 'in', item_value) = (case when item_operator = '=' then 1 else 0 end)	275	286	'refund_retreat_amount', ord.refund_retreat_amount,
and ytdw.simple_expr(war_zone_dep_id, 'in', war_area_value) = (case when war_area_operator = '=' then 1 else 0 en	276	287	'pickup_pay_gmv', ord.pickup_pay_gmv,
and ytdw.simple_expr(area_manager_dep_id, 'in', bd_area_value) = (case when bd_area_operator = '=' then 1 else 0	277	288	'pickup_pay_pay_amount', ord.pickup_pay_pay_amount,
and ytdw.simple_expr(bd_manager_dep_id, 'in', manage_area_value) = (case when manage_area_operator = '=' then 1 e	278	289	'pickup_recharge_gmv', ord.pickup_recharge_gmv,
and if(ord.shop_group = '' OR plan.shop_group_value = '', 0, ytdw.simple_expr(substr(plan.shop_group_value, 2, 1	279	290	'pickup_recharge_pay_amount', ord.pickup_recharge_pay_amount,
HAVING ytdw.simple_expr(grant_object_user_id, 'in', filter_user_value) = (case when filter_user_operator = '=' th	280	291	'pickup_pay_gmv_less_refund', ord.pickup_pay_gmv_less_refund,
)	281	292	'pickup_pay_pay_amount_less_refund', ord.pickup_pay_pay_amount_less_refund,
Insert overwrite table dw_salary_gmv_rule_public_d partition (dayid='\$v_date',pltype='cur')	282	293	'pickup_recharge_gmv_less_refund', ord.pickup_recharge_gmv_less_refund,
SELECT cur.update_time,	283	294	'pickup_recharge_pay_amount_less_refund', ord.pickup_recharge_pay_amount_less_refund
cur.update_month,	284	295) as extra
			FROM plan

3、新增基本提成指标模板

dw_salary_virtual_node_salary_ddl 新增建表语句虚拟节点

dw_salary_base_salary_forward_public_d 新增基本提成指标public脚本

dw_salary_base_salary_forward_plan_sum_d 新增基本提成指标汇总脚本

dw_salary_supply_data_base_salary 新增基本提成指标补数脚本

4、棉之悦定制逻辑

dw_salary_brand_shop_rule_public_mid_v2_d

--订单表	110	110	--订单表
from (111	111	from (
SELECT order_id,	112	112	SELECT order_id,
trade_id,	113	113	trade_id,
category_1st_id,	114	114	category_1st_id,
category_1st_name,	115	115	category_1st_name,
category_2nd_id,	116	116	category_2nd_id,
category_2nd_name,	117	117	category_2nd_name,
brand_id,	118	118	if(brand_id IN (15472, 19284), 19284, brand_id) as brand_id,
brand_name,	119	119	if(brand_id IN (15472, 19284), '棉之悦', brand_name) as brand_name,
item_id,	120	120	item_id,
item_name,	121	121	item_name,
item_style,	122	122	item_style,
item_style_name,	123	123	item_style_name,
shop_id,	124	124	shop_id,
shop_name,	125	125	shop_name,
shop_store_type,	126	126	shop_store_type,

2023.Q1

2023_02-特殊提成日常

- 一、背景
- 二、业务目标
- 三、产品描述
- 四、名词/术语解释
- 五、用例
- 六、业务流程
- 七、领域模型
- 八、系统设计
 - 需求1：新签品牌模板和新签商品模板 新增指标 新签GMV占比全月完成率
 - 需求2：新增控区HI卡充值相关指标
 - 需求3：新增不追回品牌筛选
 - 需求4：新签模板新增门店维度
 - 实现流程
- 九、类图
- 十、数据库设计
- 十一、搜索设计
- 十二、接口设计
- 十三、涉及数仓的设计
- 十四、中间件配置相关
- 十五、非功能性需求

一、背景

主要是针对以往业务方提出的特殊提成的需求中，比较简单且应用比较广泛地逻辑进行优化

二、业务目标

三、产品描述

202302_01_日常_特殊提成日常

四、名词/术语解释

五、用例

六、业务流程

七、领域模型

八、系统设计

需求1：新签品牌模板和新签商品模板 新增指标 新签GMV占比全月完成率

指标、指标奖励配置
<pre>insert into `bounty_indicator` (`id`, `title`, `bounty_rule_type`, `code`, `sort`, `unit`) values (33, 'GMV', 2, 'SIGN_ITEM_GMV_RATE', 8, '%'), (34, 'GMV', 3, 'SIGN_BRAND_GMV_RATE', 8, '%'); insert into `bounty_indicator_payout_rule_type` (`bounty_indicator_id`, `payout_rule_type_code`, `bounty_unit`, `sort`, `is_upper_limit`) values (33, 1, '', 1, 0), (33, 3, '', 3, 0), (33, 5, '', 5, 1), (34, 1, '', 1, 0), (34, 3, '', 3, 0), (34, 5, '', 5, 1);</pre>

目标取数逻辑

```
---
case when bounty_payout_object_id IN (4) AND bounty_indicator_code = 'B_SHIHUO_RATE_NO_C' then
'class_b_capacity_pure'
  when bounty_payout_object_id IN (1,2,3) AND bounty_indicator_code = 'B_SHIHUO_RATE_NO_C' then
'class_b_area_pure'
---
select t.user_id,
       t.indicator,
       t.target,
       substr(a.start_time, 0, 8) as start_time,
       substr(a.end_time, 0, 8) as end_time
from (SELECT * FROM dwd_kpi_indicator_target_d WHERE dayid = '$v_date' and is_deleted = 0) t
INNER JOIN (SELECT * FROM dwd_kpi_assessment_d WHERE dayid = '$v_date' AND is_deleted = 0 AND status IN (2,3))
a ON t.assessment_id = a.id
```

需求2：新增控区HI卡充值相关指标

指标、指标奖励配置

```
insert into `bounty_indicator` (`id`, `title`, `bounty_rule_type`, `code`, `sort`, `unit`)
values (35, 'GMV()', 1, 'HI_RECHARGE_GMV_LESS_REFUND', 14, ''),
      (36, 'GMV()', 1, 'AVG_HI_RECHARGE_GMV_LESS_REFUND', 15, '/');

insert into `bounty_indicator_payout_rule_type` (`bounty_indicator_id`, `payout_rule_type_code`, `bounty_unit`,
`sort`, `is_upper_limit`)
values (35, 1, '', 1, 0),
      (35, 2, '%', 2, 1),
      (35, 3, '', 3, 0),
      (35, 5, '', 4, 1),
      (36, 1, '', 1, 0),
      (36, 3, '', 2, 0),
      (36, 4, '/(//)', 3, 1),
      (36, 5, '', 4, 1);
```

目标取数逻辑

```
case when item_style = 1 and category_id_first_name = '' and is_pickup_recharge_order = 0 then SUM(gmv) - sum
(refund_actual_amount)
```

需求3：新增不追回品牌筛选

过滤条件、过滤条件模板映射配置

```
INSERT INTO `bounty_filter` (`id`, `title`, `key`, `field_sql`, `required`, `global`, `global_type`,
`plan_date_sign`, `component_type`, `component_value_config`)
VALUES (25, '', 'unback_brand', 'unback_brand', 0, 1, 1, 0, 'api_multi_select', '{"type\\":\\"url\\", \\"url\\":\\"
hipac.crm.bountyPlan.listFilterResult\\",\\"code\\":\\"BRAND\\"}');

insert into `bounty_rule_filter` (`bounty_rule_type`, `bounty_rule_type_id`, `bounty_filter_id`)
values (0, 4, 25),
      (0, 5, 25),
      (0, 6, 25),
      (0, 7, 25);
```

过滤逻辑

```
case when ytdw.simple_expr(brand_id, 'in', brand_value) = (case when brand_operator = '=' then 1 else 0 end)
then sum(ord.gmv - ord.refund_actual_amount)
else sum(ord.gmv - refund.refund_actual_amount)
end
```

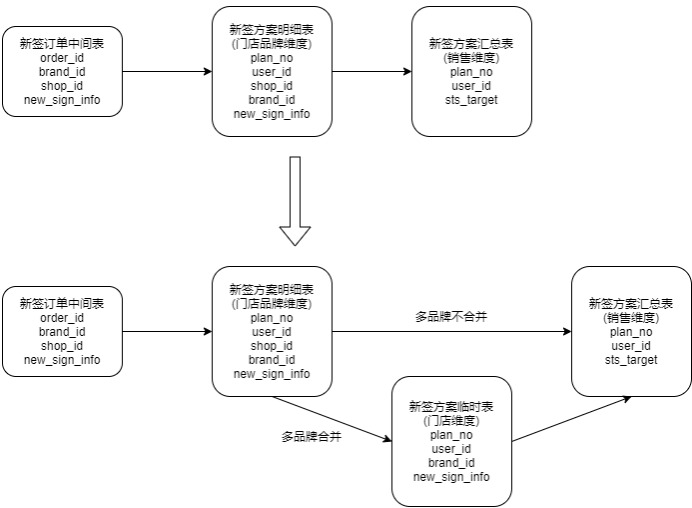
需求4：新签模板新增门店维度

过滤条件配置

```
INSERT INTO `bounty_filter` (`id`, `title`, `key`, `field_sql`, `required`, `global`, `global_type`,
`plan_date_sign`, `component_type`, `component_value_config`)
VALUES (26, '', 'merge_brand', 'merge_brand', 1, 1, 1, 0, 'enum_radio', '{"type":"fixed","data":[{"label":"","value":""},{ "label":"","value":""}]}');

insert into `bounty_rule_filter` (`bounty_rule_type`,`bounty_rule_type_id`,`bounty_filter_id`)
values (0, 6, 26);
```

实现流程



统计维护逻辑

```
with detail as (
    .....
),
with detail_total as (
    .....
)
case when plan.merge_plan = 1 then detail_total.xxx else detail.xxx end
```

- 九、类图
- 十、数据库设计
- 十一、搜索设计

十二、接口设计

十三、涉及数仓的设计

十四、中间件配置相关

十五、非功能性需求

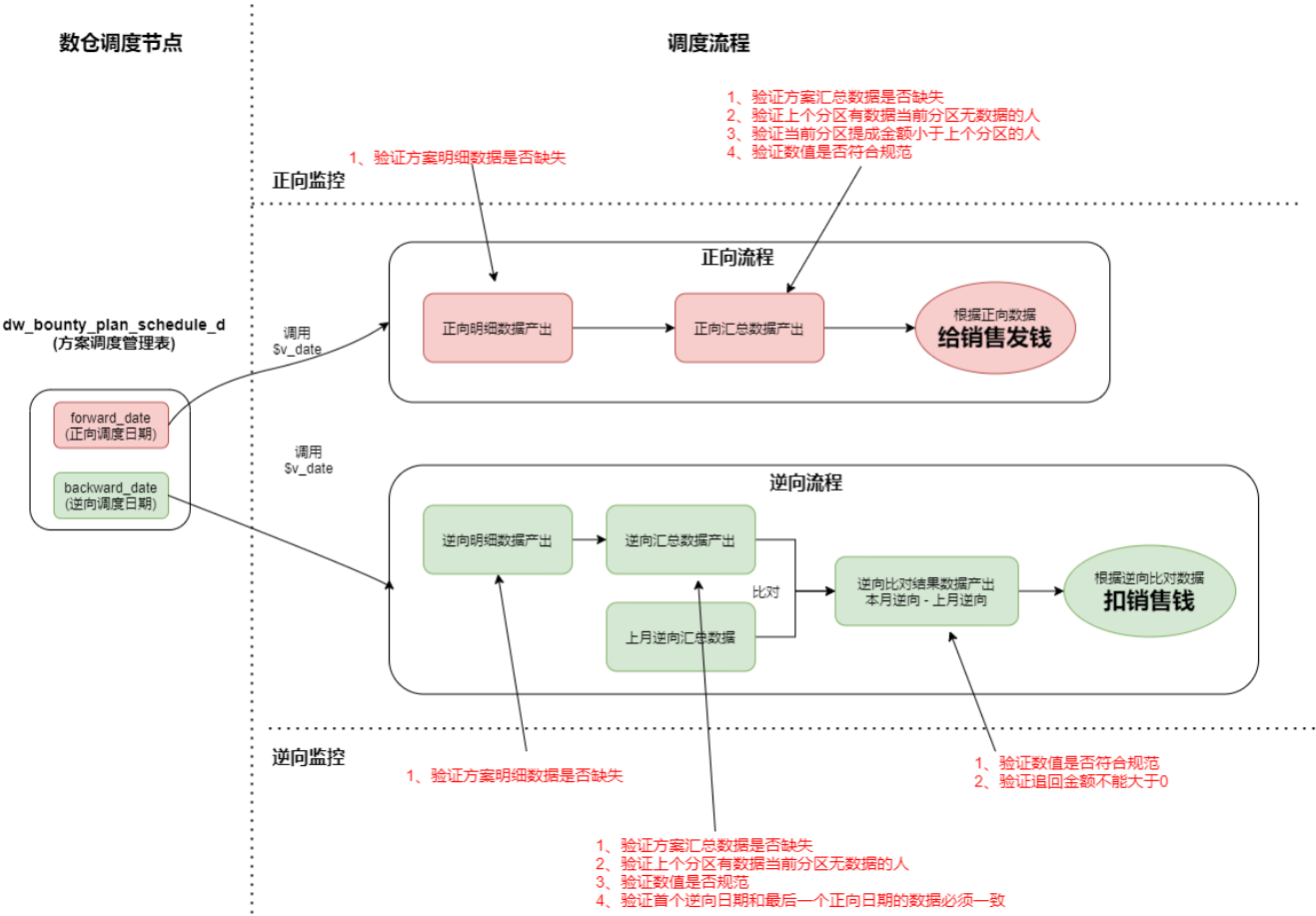
大BD接入特殊提成

prd : [202303_01_日常_特殊提成日常](#)

4、资损防控

资损防控dqc配置：<https://dqc-cn-hangzhou.data.aliyun.com/?defaultProjectId=53926#/rule/list>

<input type="checkbox"/>	规则名称	规则ID	表名	规则字段	分区表达式	关联调度	规则模板	强/弱	启停状态	配置人	操作
<input type="checkbox"/>	特殊提成逆向数据监控（上个分区有数据当前分区无数据的人）	28501140	dw_bounty_plan_schedule_d	自定义SQL	NOTAPARTITIONTABLE	已关联		弱	启动	tao.zheng8833/Ro	订阅管理 修改 删除 停止 设为强规则 日志 生成模板
<input type="checkbox"/>	特殊提成正向数据监控（上个分区有数据当前分区无数据的人）	28501139	dw_bounty_plan_schedule_d	自定义SQL	NOTAPARTITIONTABLE	已关联		弱	启动	tao.zheng8833/Ro	订阅管理 修改 删除 停止 设为强规则 日志 生成模板
<input type="checkbox"/>	特殊提成业务监控（方案最后一天正向数据不为空）	28501138	dw_bounty_plan_schedule_d	自定义SQL	NOTAPARTITIONTABLE	已关联		弱	启动	tao.zheng8833/Ro	订阅管理 修改 删除 停止 设为强规则 日志 生成模板
<input type="checkbox"/>	特殊提成逆向数据监控（金额字段不可为空）	28501137	dw_bounty_plan_schedule_d	自定义SQL	NOTAPARTITIONTABLE	已关联		弱	启动	tao.zheng8833/Ro	订阅管理 修改 删除 停止 设为强规则 日志 生成模板
<input type="checkbox"/>	特殊提成正向数据监控（金额差值不能大于0）	28501136	dw_bounty_plan_schedule_d	自定义SQL	NOTAPARTITIONTABLE	已关联		弱	启动	tao.zheng8833/Ro	订阅管理 修改 删除 停止 设为强规则 日志 生成模板
<input type="checkbox"/>	特殊提成正向数据监控（当前分区提成金额不能小于上个分区）	28501130	dw_bounty_plan_schedule_d	自定义SQL	NOTAPARTITIONTABLE	已关联		弱	启动	tao.zheng8833/Ro	订阅管理 修改 删除 停止 设为强规则 日志 生成模板



调度流程