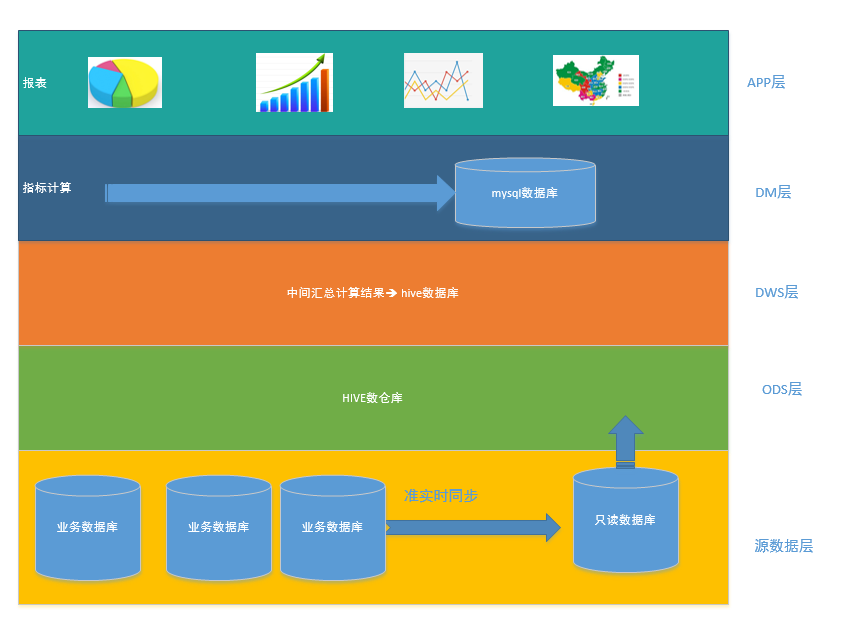
高效运营支撑平台

# 项目描述

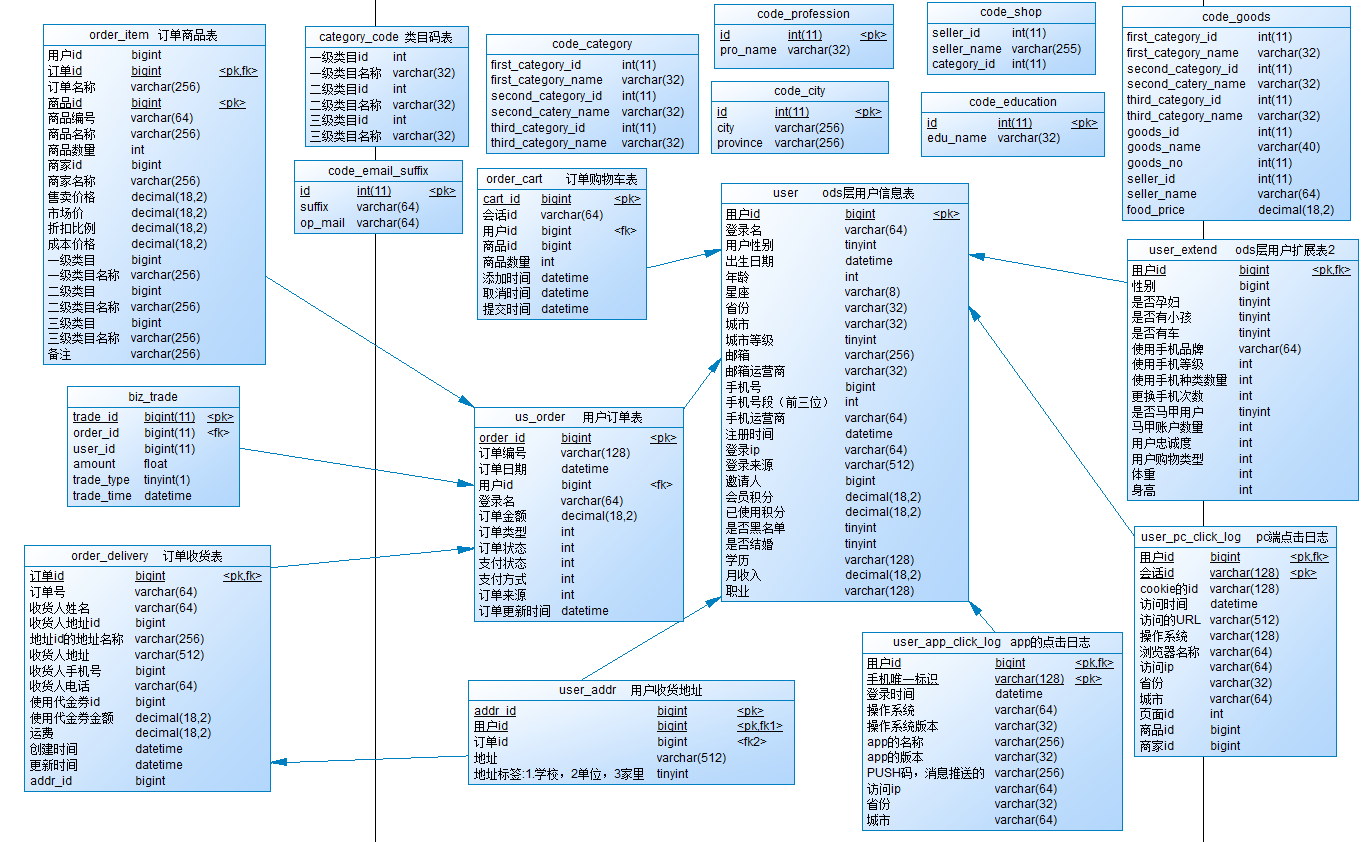
本平台是外卖行业类的自营业务的运营分析平台的一部分,通过对业务数据的分析,获得用户主题和订单的相关分析情况,生成用户的访问模型和用户的订单行为模型,对访问时段和下单区域分布进行分析和报表展示.

# 业务系统说明

用户使用浏览器(PC端)和app对外卖平台进行访问,记录用户的注册信息,通过调研平台和数据挖掘记录用户的扩展信息,业务系统对用户的每次点击和下单信息记录到数据库中,并提供一台从库将主库的业务数据实时同步到从库中,同时分配对从库的只读访问权限,运营平台通过sqoop将数据抽取到运营分析平台,运营分析平台进行指标的计算,将结果存入到mysql中,将结果展示到报表.具体架构如下



业务表数据模型



# 需求分析

## 用户访问模型

1. 生成用户基本宽表user\_basic

将用户的基本信息表和用户的扩展表进行合并形成用户宽表.

包括信息如下:

用户id,登录名 ,用户性别,出生日期,年龄,星座,省份,城市 ,城市等级,邮箱 ,邮箱运营商 ,手机号 ,手机号段（前三位）,手机运营商,注册时间,登录ip,登录来源,邀请人,会员积分,已使用积分,会员等级名称,是否黑名单,是否结婚,学历,月收入 ,职业,是否孕妇,是否有小孩,是否有车,使用手机品牌,使用手机等级,使用手机种类数量,更换手机数量 ,是否马甲用户,马甲账户数量,用户忠诚度,用户购物类型,体重,身高。

1. 根据用户的访问日志信息,统计以下指标,结果存入用户访问模型表user\_visit

指标如下:

PC端指标

* + - * 最近一次访问时间
      * 最近一次访问使用的session
      * 最近一次使用的coookie
      * 最近一次的pc端的pv量
      * 最近一次访问使用的浏览器
      * 最近一次访问使用的操作系统
      * 第一次pc端访问的日期
      * 第一次pc端访问的session
      * 第一次pc端访问的cookie
      * 第一次访问的pv
      * 第一次访问使用的浏览器
      * 第一次访问的os
      * PC连续7天访问次数(跑任务的日期的前7天,以下相同不再赘述)
      * 连续15天访问次数
      * 连续30天访问次数
      * 连续60天访问的次数
      * 连续90天访问的次数
      * 近30天pc端访问的次数
      * 近30天pc端的pv
      * 近30天pc端每天的平均pv
      * 近30天的0到5点的pv数量
      * 近30天的6到7点的pv数量
      * 近30天的8到9的pv数量
      * 近30天的10到11的pv数量
      * 近30天的12到13的pv数量
      * 近30天的14到16点的pv数量
      * 近30天的17到19点的pv数量
      * 近30天的18到19点的pv数量
      * 近30天的20到21点的pv数量
      * 近30天的22到23点的pv数量
      * 近30天访问使用的不同ip数量
      * 近30天最常用的ip
      * 近30天使用的cookie的数量
      * 近30使用最常用的cookie\_id
      * 近30pc最常用浏览器
      * 近30天使用最常用系统

APP端指标

* + - * 最近一次app访问的日期
      * 最近一次访问app的名称
      * 最近一次app访问的操作系统
      * 第一次app访问日期
      * 第一app访问名称
      * 第一次app访问os
      * app第一次访问ip
      * app 近7天访问pv数
      * app 近15天访问pv数
      * app 近30天的访问pv数
      * app近60天的访问pv数
      * app近90天的访问pv数
      * app近30天0到5点的访问pv数
      * app近30天的6到7点的访问pv数
      * app近30天8到9的访问pv数
      * app近30天10到11访问pv数
      * app近30天12到13点的访问pv数
      * app近30天14到15点的访问pv数
      * app近30天16到17点的访问pv数
      * app近30天18到19点的访问pv数
      * app近30天20到21点的访问pv数
      * app近30天22到23点的访问pv数

综合指标

* 最近一次访问的ip
* 最近一次访问的城市
* 最近一次访问的省份
* 第一次访问的ip
* 第一次访问的城市
* 第一次访问的省份

## 用户订单模型

订单指标

* 第一次下单时间,
* 最近一次下单时间,
* 首单距今时间,
* 尾单距今时间,
* 近30天订单数量（不含退拒）,
* 近30天订单金额（不含退拒）,
* 近60天订单数量（不含退拒）,
* 近60天订单金额（不含退拒）,
* 近90天订单数量（不含退拒）,
* 近90天订单金额（不含退拒）,
* 近30天订单数量（含退拒）,
* 近30天订单金额（含退拒）,
* 近60天购买次数（含退拒）,
* 近60天购买金额（含退拒）,
* 近90天购买次数（含退拒）,
* 近90天购买金额（含退拒）,
* 近90天的客单价（含退拒）,
* 最大消费金额 ,
* 最小消费金额,
* 累计消费次数（不含退拒）,
* 累计消费金额（不含退拒）,
* 客单价（含退拒）,
* 最常用收货地址,
* 最常用支付方式
* 退货商品数量,
* 退货商品金额,
* 拒收商品数量,
* 拒收商品金额,
* 最近一次退货时间

下单分布：

* 学校下单总数,
* 公司下单总数,
* 家里下单总数,
* 凌晨下单总数,
* 上午下单总数,
* 中午下单总数，
* 下午下单总数,
* 晚上下单总数,
* 指标口径:凌晨:0-05 上午:06-12 中午:13-15 下午16-20 晚上：21-24

## 访问情况时段统计图

将指定的日期的每个时间区间的访问统计情况进行图形展示,横坐标是时间点,纵坐标是访问次数

## 当天访问用户top20

以柱状图的形式显示当天访问用户的top20,横坐标为用户登录名,纵坐标为pv量

## 订单情况时段统计图

将指定的日期的每个时间区间的订单统计情况进行图形展示,横坐标是时间点,纵坐标是下单数量

## 当天区域下单量的分布图

以地图的方式显示当天下单金额的各省份的分布图.

# 技术选型

数据导出:sqoop

数据计算:Spark SQL

数据仓库:Hive

任务调度:Azkaban

# 运营平台模型设计

## ODS层 源数据层

映射关系

|  |  |  |
| --- | --- | --- |
| 源表(mysql) | 目标表(hive) | 映射关系 |
| user | qfbap\_ods.ods\_user |  |
| user\_extend | qfbap\_ods.ods\_user\_extend |  |
| user\_pc\_click\_log | qfbap\_ods.ods\_user\_pc\_click |  |
| user\_app\_click\_log | qfbap\_ods.ods\_user\_app\_click\_log |  |
| user\_addr | qfbap\_ods.ods\_user\_addr |  |
| user\_order | qfbap\_ods.ods\_user\_order |  |
| order\_item | qfbap\_ods.ods\_order\_item |  |
| order\_delivery | qfbap\_ods.ods\_order\_delivery |  |
| order\_cart | qfbap\_ods.ods\_order\_cart |  |
| biz\_trade | qfbap\_ods.ods\_biz\_trade |  |
| order\_cart | qfbap\_ods.ods\_order\_cart |  |

## DWD层

|  |  |  |
| --- | --- | --- |
| 源表(hive) | 目标表(hive) | 映射关系 |
| qfbap\_ods.ods\_user | qfbap\_dwd.dwd\_qfbap\_user |  |
| qfbap\_ods.ods\_user\_extend | qfbap\_dwd.dwd\_user\_extend |  |
| qfbap\_ods.ods\_user\_pc\_click\_log | qfbap\_dwd.dwd\_user\_pc\_pv | SELECT  t.user\_id,  t.session\_id,  t.cookie\_id,  MIN(visit\_time) in\_time,  MAX(visit\_time) out\_time,  (  case  WHEN MIN(visit\_time) = MAX(visit\_time)  then 3  else unix\_timestamp(MAX(visit\_time)) - unix\_timestamp(MIN(visit\_time))  end  ) stay\_time,  COUNT(1) pv,  t.visit\_os,  t.browser\_name,  t.visit\_ip,  t.province,  t.city  FROM  qfbap\_ods.ods\_user\_pc\_click\_log t  GROUP BY  t.session\_id,  t.cookie\_id,  t.user\_id,  t.visit\_os,  t.browser\_name,  t.visit\_ip,  t.province,  t.city; |
| qfbap\_ods.ods\_user\_app\_pv | qfbap\_dwd.dwd\_user\_app\_pv | 增加log\_hour,对hour(log\_time)🡪 log\_hour |
| qfbap\_ods.ods\_user\_addr | qfbap\_dwd.dwd\_user\_addr |  |
| qfbap\_ods.ods\_user\_order | qfbap\_dwd.dwd\_user\_order |  |
| qfbap\_ods.ods\_order\_item | qfbap\_dwd.dwd\_order\_item |  |
| qfbap\_ods.ods\_order\_delivery | qfbap\_dwd.dwd\_order\_delivery |  |
| qfbap\_ods.ods\_order\_cart | qfbap\_dwd.dwd\_order\_cart |  |
| qfbap\_ods.ods\_biz\_trade | qfbap\_dwd.dwd\_biz\_trade |  |
| qfbap\_ods.ods\_order\_cart | qfbap\_dwd.dwd\_order\_cart |  |

## DWS层

|  |  |  |
| --- | --- | --- |
| 源表(hive) | 目标表(hive) | 映射关系 |
| qfbap\_dwd.dwd\_user\_pc\_pv | qfbap\_dws.dws\_user\_visit\_month1 | select  a.user\_id,  a.type,  a.cnt,  a.content,  a.rn  from (  select  t.user\_id,  t.type,  t.cnt,  t.content,  row\_number() over(distribute by user\_id,type sort by cnt desc) rn  from (  select  user\_id,  'visit\_ip' as type,-- 近30天访问ip  sum(pv) as cnt,  visit\_ip as content  from qfbap\_dwd.dwd\_user\_pc\_pv  where dt >= date\_add('2018-12-25',-29)  group by  user\_id,  visit\_ip  union all  select  user\_id,  'cookie\_id' as type, -- 近30天常用cookie  sum(pv) as cnt,  cookie\_id as content  from qfbap\_dwd.dwd\_user\_pc\_pv  where dt >= date\_add('2018-12-25',-29)  group by  user\_id,  cookie\_id  union all  select  user\_id,  'browser\_name' as type,-- 近30天常用浏览器  sum(pv) as cnt,  browser\_name as content  from qfbap\_dwd.dwd\_user\_pc\_pv  where dt >= date\_add('2018-12-25',-29)  group by  user\_id,  browser\_name  union all  select  user\_id,  'visit\_os' as type, -- 近30天常用操作系统  sum(pv) as cnt,  visit\_os as content  from qfbap\_dwd.dwd\_user\_pc\_pv  where dt >= date\_add('2018-12-25',-29)  group by  user\_id,  visit\_os  ) t  ) a |
|  |  |  |

## DM层

|  |  |  |
| --- | --- | --- |
| 源表(hive) | 目标表(mysql) | 映射关系 |
| qfbap\_dwd.dwd\_user  qfbap\_dwd.dwd\_user\_extend | dm\_user\_basic | select  a.user\_id ,  a.user\_name ,  a.user\_gender ,  a.user\_birthday ,  a.user\_age ,  a.constellation ,  a.province ,  a.city ,  a.city\_level ,  a.e\_mail ,  a.op\_mail ,  a.mobile ,  a.num\_seg\_mobile ,  a.op\_mobile ,  a.register\_time ,  a.login\_ip ,  a.login\_source ,  a.request\_user ,  a.total\_score ,  a.used\_score ,  a.is\_blacklist ,  a.is\_married ,  a.education ,  a.monthly\_income ,  a.profession ,  b.is\_pregnant\_woman,  b.is\_have\_children ,  b.is\_have\_car ,  b.phone\_brand ,  b.phone\_brand\_level,  b.phone\_cnt ,  b.change\_phone\_cnt ,  b. is\_maja ,  b.majia\_account\_cnt,  b.loyal\_model ,  b.shopping\_type\_model,  b.weight ,  b.height ,  b.dw\_date  from qfbap\_dwd.dwd\_user a  left join qfbap\_dwd.dwd\_user\_extend b on a.user\_id = b.user\_id |
| dws\_user\_basic  qfbap\_dwd.dwd\_user\_app\_pv  qfbap\_dwd.dwd\_user\_pc\_pv  qfbap\_dws.dws\_user\_visit\_month1 | qfbap\_dm.dm\_user\_visit | 经过计算,详见下方脚本:计算用户行为 |

|  |
| --- |
| 计算用户行为  select  us.user\_id  ,r\_pc.latest\_pc\_visit\_date-- 最近一次访问时间  ,r\_pc.latest\_pc\_visit\_session -- 最近一次访问的session  ,r\_pc.latest\_pc\_cookies-- 最近一次的coookie  ,r\_pc.latest\_pc\_pv -- 最近一次的pc端的pv量  ,r\_pc.latest\_pc\_browser\_name -- 最近一次访问使用的浏览器  ,r\_pc.latest\_pc\_visit\_os-- 最近一次访问使用的操作系统  ,r\_pc.first\_pc\_visit\_date -- 最早pc端访问的日期  ,r\_pc.first\_pc\_visit\_session--最早pc端访问的session  ,r\_pc.first\_pc\_cookies-- 最早pc端访问的cookie  ,r\_pc.first\_pc\_pv-- 最早一次访问的pv  ,r\_pc.first\_pc\_browser\_name -- 最早一次访问使用的浏览器  ,r\_pc.first\_pc\_visit\_os-- 最早一次访问的os  ,r\_pc.day7\_pc\_cnt --PC连续7天访问次数  ,r\_pc.day15\_pc\_cnt-- 连续15天访问次数  ,r\_pc.month1\_pc\_cnt -- 连续30天访问次数  ,r\_pc.month2\_pc\_cnt-- 连续60天访问的次数  ,r\_pc.month3\_pc\_cnt -- 连续90天访问的次数  ,r\_pc.month1\_pc\_days --近30天pc端访问的次数  ,r\_pc.month1\_pc\_pv --近30天pc端的pv  ,r\_pc.month1\_pc\_avg\_pv --近30天pc端每天的平均pv  ,r\_pc.month1\_pc\_hour025\_cnt --0到5点的数量  ,r\_pc.month1\_pc\_hour627\_cnt --6到7点的数量  ,r\_pc.month1\_pc\_hour829\_cnt -- 8到9的数量  ,r\_pc.month1\_pc\_hour10211\_cnt -- 10到11的数量  ,r\_pc.month1\_pc\_hour12213\_cnt --12到13的数量  ,r\_pc.month1\_pc\_hour14216\_cnt -- 14到16点的数量  ,r\_pc.month1\_pc\_hour17219\_cnt -- 17到19点的数量  ,r\_pc.month1\_pc\_hour18219\_cnt -- 18到19点的数量  ,r\_pc.month1\_pc\_hour20221\_cnt -- 20到21点的数量  ,r\_pc.month1\_pc\_hour22223\_cnt -- 22到23点的数量  ,pc\_month1.month1\_pc\_diff\_ip\_cnt --近30天访问使用的不同ip数量  ,pc\_month1.month1\_pc\_common\_ip --近30天最常用的ip  ,pc\_month1.month1\_pc\_diff\_cookie\_cnt --近30天使用的cookie的数量  ,pc\_month1.month1\_pc\_common\_cookie --近30使用最常用的cookie\_id  ,pc\_month1.month1\_pc\_common\_browser\_name -- pc最常用浏览器  ,pc\_month1.month1\_pc\_common\_os -- 近30天使用最常用系统  ,r\_app.latest\_app\_visit\_date, --最近一次app访问的日期  r\_app.latest\_app\_name, -- 最近一次访问app的名称  r\_app.latest\_app\_visit\_os, -- 最近一次app访问的操作系统  r\_app.first\_app\_visit\_date, -- 第一次app访问日期  r\_app.first\_app\_name,-- 第一app访问名称  r\_app.first\_app\_visit\_os,-- 第一次app访问os  r\_app.first\_visit\_ip,--app第一次访问ip  r\_app.day7\_app\_cnt,-- app 近7天访问次数  r\_app.day15\_app\_cnt,-- app 近15天访问次数  r\_app.month1\_app\_cnt, -- app 近30天的访问次数  r\_app.month2\_app\_cnt, -- app近60天的访问次数  r\_app.month3\_app\_cnt, -- app近90天的访问次数  r\_app.month1\_app\_hour025\_cnt, -- app近30天0到5点的访问次数  r\_app.month1\_app\_hour627\_cnt,-- app近30天的6到7点的访问次数  r\_app.month1\_app\_hour829\_cnt, -- app近30天8到9的访问次数  r\_app.month1\_app\_hour10211\_cnt,-- app近30天10到11访问次数  r\_app.month1\_app\_hour12213\_cnt, -- app近30天12到13点的访问次数  r\_app.month1\_app\_hour14215\_cnt ,-- app近30天14到15点的访问次数  r\_app.month1\_app\_hour16217\_cnt ,-- app近30天16到17点的访问次数  r\_app.month1\_app\_hour18219\_cnt ,-- app近30天18到19点的访问次数  r\_app.month1\_app\_hour20221\_cnt ,-- app近30天20到21点的访问次数  r\_app.month1\_app\_hour22223\_cnt ,-- app近30天22到23点的访问次数  (case  when r\_pc.latest\_pc\_visit\_date>= r\_app.latest\_app\_visit\_date  then r\_pc.latest\_visit\_ip  else r\_app.latest\_visit\_ip  end ) latest\_visit\_ip , --最近一次访问的ip  (case  when r\_pc.latest\_pc\_visit\_date >= r\_app.latest\_app\_visit\_date  then r\_pc.latest\_city  else r\_app.latest\_city  end ) latest\_city , --最近一次访问的城市  (case  when r\_pc.latest\_pc\_visit\_date >= r\_app.latest\_app\_visit\_date  then r\_pc.latest\_city  else r\_app.latest\_city  end  ) latest\_province ,-- 最近一次访问的省份  (case  when r\_pc.first\_pc\_visit\_date <= r\_app.first\_app\_visit\_date  then r\_pc.first\_visit\_ip  else r\_app.first\_visit\_ip  end  ) first\_visit\_ip , -- 第一次访问的ip  (case  when r\_pc.first\_pc\_visit\_date <= r\_app.first\_app\_visit\_date  then r\_pc.first\_city  else r\_app.first\_city  end  ) first\_city ,-- 第一次访问的城市  (case  when r\_pc.first\_pc\_visit\_date <= r\_app.first\_app\_visit\_date  then r\_pc.first\_province  else r\_app.first\_province  end  ) first\_province -- 第一次访问的省份  from qfbap\_dwd.dws\_user\_basic us  left join (  -- PC端的统计指标  select  user\_id  ,max(case when pc.rn\_desc = 1 then in\_time end) latest\_pc\_visit\_date -- 最近一次访问时间  ,max(case when pc.rn\_desc=1 then visit\_ip end) latest\_visit\_ip --最近一次的访问ip  ,max(case when pc.rn\_desc=1 then province end) latest\_province  ,max(case when pc.rn\_desc=1 then city end) latest\_city  ,max(case when pc.rn\_desc = 1 then session\_id end) latest\_pc\_visit\_session -- 最近一次访问的session  ,max(case when pc.rn\_desc = 1 then cookie\_id end ) latest\_pc\_cookies-- 最近一次的coookie  ,max(case when pc.rn\_desc = 1 then pv end) latest\_pc\_pv -- 最近一次的pc端的pv量  ,max(case when pc.rn\_desc = 1 then browser\_name end ) latest\_pc\_browser\_name -- 最近一次访问使用的浏览器  ,max(case when pc.rn\_desc = 1 then visit\_os end ) latest\_pc\_visit\_os-- 最近一次访问使用的操作系统  ,max(case when pc.rn\_asc = 1 then in\_time end) first\_pc\_visit\_date -- 最早pc端访问的日期  ,max(case when pc.rn\_asc=1 then visit\_ip end) first\_visit\_ip --最近一次的访问ip  ,max(case when pc.rn\_asc=1 then province end) first\_province  ,max(case when pc.rn\_asc=1 then city end) first\_city  ,max(case when pc.rn\_asc = 1 then session\_id end ) first\_pc\_visit\_session--最早pc端访问的session  ,max(case when pc.rn\_asc = 1 then cookie\_id end ) first\_pc\_cookies-- 最早pc端访问的cookie  ,max(case when pc.rn\_asc = 1 then pv end ) first\_pc\_pv-- 最早一次访问的pv  ,max(case when pc.rn\_asc = 1 then browser\_name end) first\_pc\_browser\_name -- 最早一次访问使用的浏览器  ,max(case when pc.rn\_asc = 1 then visit\_os end) first\_pc\_visit\_os-- 最早一次访问的os  ,sum(dt7) day7\_pc\_cnt --连续7天访问次数  ,sum(dt15) day15\_pc\_cnt-- 连续15天访问次数  ,sum(dt30) month1\_pc\_cnt -- 连续30天访问次数  ,sum(dt60) month2\_pc\_cnt-- 连续60天访问的次数  ,sum(dt90) month3\_pc\_cnt -- 连续90天访问的次数  ,count(distinct (case when dt30=1 then substr(in\_time,0,8) end)) month1\_pc\_days --近30天pc端访问的次数  ,sum(case when dt30=1 then pv end ) month1\_pc\_pv --近30天pc端的pv  ,sum(case when dt30=1 then pv end )  /count(distinct(case when dt30=1 then substr(in\_time,0,8) end)) month1\_pc\_avg\_pv --近30天pc端每天的平均pv  ,count(case when dt30=1 and hr025=1 then 1 end ) month1\_pc\_hour025\_cnt --0到5点的数量  ,count(case when dt30=1 and hr627=1 then 1 end ) month1\_pc\_hour627\_cnt --6到7点的数量  ,count(case when dt30=1 and hr829=1 then 1 end ) month1\_pc\_hour829\_cnt -- 8到9的数量  ,count(case when dt30=1 and hr10211=1 then 1 end ) month1\_pc\_hour10211\_cnt -- 10到11的数量  ,count(case when dt30=1 and hr12213=1 then 1 end ) month1\_pc\_hour12213\_cnt --12到13的数量  ,count(case when dt30=1 and hr14215=1 then 1 end ) month1\_pc\_hour14216\_cnt -- 14到16点的数量  ,count(case when dt30=1 and hr16217=1 then 1 end ) month1\_pc\_hour17219\_cnt -- 17到19点的数量  ,count(case when dt30=1 and hr18219=1 then 1 end ) month1\_pc\_hour18219\_cnt -- 18到19点的数量  ,count(case when dt30=1 and hr20221=1 then 1 end ) month1\_pc\_hour20221\_cnt -- 20到21点的数量  ,count(case when dt30=1 and hr22223=1 then 1 end ) month1\_pc\_hour22223\_cnt -- 22到23点的数量  from (  select  row\_number() over(distribute by user\_id sort by in\_time asc) rn\_asc,  row\_number() over(distribute by user\_id sort by in\_time desc) rn\_desc,  user\_id,  session\_id,  cookie\_id,  visit\_os,  browser\_name,  visit\_ip,  province,  city,  (case when in\_time >=date\_add('2018-12-25',-6) then 1 end ) dt7,  (case when in\_time>=date\_sub('2018-12-25',14) then 1 end ) dt15,  (case when in\_time >= date\_sub('2018-12-25',29) then 1 end) dt30,  (case when in\_time >= date\_sub('2018-12-25',59) then 1 end) dt60,  (case when in\_time >= date\_sub('2018-12-25',179) then 1 end) dt90,  (case when hour(in\_time) between 0 and 5 then 1 end) hr025,  (case when hour(in\_time) between 6 and 7 then 1 end ) hr627,  (case when hour(in\_time) between 8 and 9 then 1 end ) hr829,  (case when hour(in\_time) between 10 and 11 then 1 end ) hr10211,  (case when hour(in\_time) between 12 and 13 then 1 end ) hr12213,  (case when hour(in\_time) between 14 and 15 then 1 end ) hr14215,  (case when hour(in\_time) between 16 and 17 then 1 end ) hr16217,  (case when hour(in\_time) between 18 and 19 then 1 end ) hr18219,  (case when hour(in\_time) between 20 and 21 then 1 end ) hr20221,  (case when hour(in\_time) between 22 and 23 then 1 end ) hr22223,  in\_time,  out\_time,  stay\_time,  pv  from qfbap\_dwd.dwd\_user\_pc\_pv t  where dt ='2018-12-25'  ) pc  group by user\_id  ) r\_pc on us.user\_id = r\_pc.user\_id  left join (  -- pc端近30天的访问数据  select  user\_id,  count(distinct  case  when type='visit\_ip'  then content  end) month1\_pc\_diff\_ip\_cnt, --近30天访问使用的不同ip数量  max(case  when rn=1  and type='visit\_ip'  then content  end) month1\_pc\_common\_ip, --近30天最常用的ip  count(  distinct  case  when rn=1  and type = 'cookie\_id'  then content  end  ) month1\_pc\_diff\_cookie\_cnt, --近30天使用的cookie的数量  max(case  when rn=1  and type='cookie\_id'  then content  end) month1\_pc\_common\_cookie, --近30使用最常用的cookie\_id  max(case  when rn=1  and type='browser\_name'  then content  end) month1\_pc\_common\_browser\_name,  max(case  when rn=1  and type='visit\_os'  then content  end) month1\_pc\_common\_os -- 近30天使用最常用系统    from qfbap\_dws.dws\_user\_visit\_month1  group by user\_id  ) pc\_month1 on us.user\_id = pc\_month1.user\_id  left join (  -- app端的统计指标  select  app.user\_id,  max(case when rn\_desc = 1 then log\_time end) latest\_app\_visit\_date ,  max(case when rn\_desc=1 then app\_name end) latest\_app\_name,  max(case when rn\_desc=1 then visit\_os end) latest\_app\_visit\_os,  max(case when rn\_desc=1 then visit\_ip end) latest\_visit\_ip,  max(case when rn\_desc=1 then city end) latest\_city,  max(case when rn\_desc=1 then province end) latest\_province,  max(case when rn\_asc=1 then log\_time end ) first\_app\_visit\_date,  max(case when rn\_asc=1 then app\_name end) first\_app\_name,  max(case when rn\_asc=1 then visit\_os end) first\_app\_visit\_os,  max(case when rn\_asc=1 then visit\_ip end) first\_visit\_ip,  max(case when rn\_asc=1 then city end) first\_city,  max(case when rn\_asc=1 then province end) first\_province,  sum(app\_dt7) day7\_app\_cnt,  sum(app\_dt15) day15\_app\_cnt,  sum(app\_dt30) month1\_app\_cnt,  sum(app\_dt60) month2\_app\_cnt,  sum(app\_dt90) month3\_app\_cnt,  sum(case when app\_dt30 =1 then app\_hr\_025 end) month1\_app\_hour025\_cnt,  sum(case when app\_dt30 =1 then app\_hr\_627 end) month1\_app\_hour627\_cnt,  sum(case when app\_dt30 =1 then app\_hr\_829 end) month1\_app\_hour829\_cnt,  sum(case when app\_dt30 =1 then app\_hr\_10211 end) month1\_app\_hour10211\_cnt,  sum(case when app\_dt30 =1 then app\_hr\_12213 end) month1\_app\_hour12213\_cnt,  sum(case when app\_dt30 =1 then app\_hr\_14215 end) month1\_app\_hour14215\_cnt ,  sum(case when app\_dt30 =1 then app\_hr\_16217 end) month1\_app\_hour16217\_cnt ,  sum(case when app\_dt30 =1 then app\_hr\_18219 end) month1\_app\_hour18219\_cnt ,  sum(case when app\_dt30 =1 then app\_hr\_20221 end) month1\_app\_hour20221\_cnt ,  sum(case when app\_dt30 =1 then app\_hr\_22223 end) month1\_app\_hour22223\_cnt  from (  select  user\_id ,  log\_time ,  log\_hour ,  phone\_id ,  visit\_os ,  os\_version ,  app\_name ,  app\_version,  device\_token,  visit\_ip,  province,  city,  row\_number() over(distribute by user\_id sort by log\_time asc) rn\_asc,  row\_number() over(distribute by user\_id sort by log\_time desc) rn\_desc,  (case when log\_time>=date\_sub('2018-12-25',6) then 1 end) app\_dt7,  (case when log\_time>=date\_sub('2018-12-25',14) then 1 end) app\_dt15,  (case when log\_time>=date\_sub('2018-12-25',29) then 1 end) app\_dt30,  (case when log\_time>=date\_sub('2018-12-25',59) then 1 end) app\_dt60,  (case when log\_time>=date\_sub('2018-12-25',89) then 1 end) app\_dt90,  (case when hour(log\_time) between 0 and 5 then 1 end) app\_hr\_025,  (case when hour(log\_time) between 6 and 7 then 1 end) app\_hr\_627,  (case when hour(log\_time) between 8 and 9 then 1 end) app\_hr\_829,  (case when hour(log\_time) between 10 and 11 then 1 end) app\_hr\_10211,  (case when hour(log\_time) between 12 and 13 then 1 end) app\_hr\_12213,  (case when hour(log\_time) between 14 and 15 then 1 end) app\_hr\_14215,  (case when hour(log\_time) between 16 and 17 then 1 end) app\_hr\_16217,  (case when hour(log\_time) between 18 and 19 then 1 end) app\_hr\_18219,  (case when hour(log\_time) between 20 and 21 then 1 end) app\_hr\_20221,  (case when hour(log\_time) between 22 and 23 then 1 end) app\_hr\_22223  from qfbap\_dwd.dwd\_user\_app\_pv  ) app  group by user\_id  ) r\_app on us.user\_id = r\_app.user\_id; |

# 环境搭建

开发环境

版本控制

流程和任务及bug追踪

业务数据的初始化

# SQL思路分析

给大家介绍两种常见的SQL思路,最常用的max,min这种很简单就不说了,说一下常见的分析思路.

1. 当看到有求第一次最后一次,最大最小或者topn的详细数据,增加排名序号

例如我们需求中的用户行为求最近一次访问时间,最近一次访问地址,第一次访问时间,第一次访问地址,我们可以对数据进行两次排名

1. 根据用户分组按照时间进行升序和降序排名

select

row\_number() over(distribute by user\_id sort by visit\_time asc) as rn\_asc,

row\_number() over(distribute by user\_id sort by visit\_time desc) as rn\_desc,

visit\_time,

visit\_ip

from user\_app\_click\_log

1. 从上面的结果中取rn\_asc=1的数据就是第一次的访问记录行
2. 从上面的结果中取rn\_desc=1的数据就是最后一次的访问记录行
3. 使用case语句进行判断,进行行转列

case when rn\_asc=1 then visit\_time end as first\_time

case when rn\_desc=1 then visit\_time end as latest\_time

1. 取数据

select

(case when rn\_asc=1 then visit\_time end) as first\_time,

(case when rn\_desc=1 then visit\_time end) as latest\_time,

(case when rn\_asc=1 then visit\_ip end) as first\_ip,

(case when rn\_desc=1 then visit\_ip end) as latest\_ip

from (

select

row\_number() over(distribute by user\_id sort by visit\_time) as rn\_asc,

row\_number() over(distribute by user\_id sort by visit\_time) as rn\_desc,

visit\_time,

visit\_ip

from user\_app\_click\_log ) t

where rn\_asc=1 and rn\_desc=1

f) topn,有了排名topn就很容易了.

1. 计算N天的访问次数

给每条记录打上一个标签,这也是数仓中常用的存储设计方式,例如有销售记录,按照区域进行统计,数仓中会转换为一张宽表

源数据

2018-12-29 1000000 北京

2018-12-28 2000000 上海

数仓中的存储格式很可能是这样的

日期 北京 上海 广州

2018-12-29 1000000 0 0

2018-12-28 0 2000000 0

那么我们在统计数据的时候,北京的金额就sum(北京),上海的就sum(上海),广州的就sum(广州)即可

计算N天的访问次数也可以采用类似的思想,如果访问日期在这个范围内,则给一个标识值,比如当前日期为2018-12-29

select

case when visit\_time>=date\_sub(‘2018-12-29’,7) then 1 end as dt\_7,

case when visit\_time>=date\_sub(‘2018-12-29’,14) then 1 end as dt\_15,

case when visit\_time>=date\_sub(‘2018-12-29’,29) then 1 end as dt\_30

from user\_app\_click\_log

有了flag值以后,我们就可以进行汇总了

select

sum(dt\_7) day7,

sum(dt\_15) day15,

sum(dt\_30) day30

from (

select

case when visit\_time>=date\_sub(‘2018-12-29’,7) then 1 end as dt\_7,

case when visit\_time>=date\_sub(‘2018-12-29’,14) then 1 end as dt\_15,

case when visit\_time>=date\_sub(‘2018-12-29’,29) then 1 end as dt\_30

from user\_app\_click\_log) t

group by user\_id

# 任务调度

调度的任务主要是抽取数据和进行指标计算,调度工具使用azkaban

## 数据初始化

数据初始化分为两部分mysql部分和hive部分

### mysql

1. 执行sql脚本,创建数据库和表,脚本为\qfbap\数据初始化\qfbap\_data\_init.sql
2. 执行脚本后,会自动开始生成数据
3. 数据生成一定量后,可以将生成数据的任务停止

使用sqlyog等数据库客户端软件,新建一个查询窗口,执行以下语句,修改事件

源事件为:

|  |
| --- |
| DELIMITER $$  ALTER EVENT `even\_init` ON SCHEDULE EVERY 1 SECOND STARTS '2018-12-17 00:22:00' ON COMPLETION PRESERVE ENABLE DO BEGIN  CALL usp\_create\_user\_withorder(10);  END$$ |

**修改为:**

|  |
| --- |
| DELIMITER $$  ALTER EVENT `even\_init` ON SCHEDULE EVERY 1 SECOND STARTS '2018-12-17 00:22:00' ON COMPLETION PRESERVE ENABLE DO BEGIN  -- CALL usp\_create\_user\_withorder(10);  END$$ |

**或者修改为:**

|  |
| --- |
| DELIMITER $$  ALTER EVENT `even\_init` ON SCHEDULE EVERY 1 SECOND STARTS '2018-12-17 00:22:00' ON COMPLETION PRESERVE disable DO BEGIN  CALL usp\_create\_user\_withorder(10);  END$$  DELIMITER ; |

### hive

1. 执行三个sql文件,来创建库和表:

hive\_ods\_create\_table.sql,

hive\_dwd\_create\_table.sql,

hive\_dws\_create\_table.sql

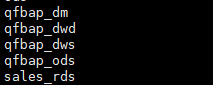
使用hive -f 来执行上面的三个文件即可.

hive -f hive\_ods\_create\_table.sql

hive -f hive\_dwd\_create\_table.sql

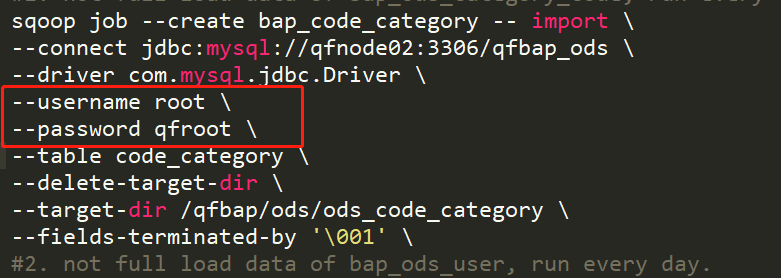
hive -f hive\_dws\_create\_table.sql

执行完后,show databases查看数据库是否创建成功



1. 执行两个shell文件,来创建sqoop任务

注意修改文件,更改成自己的密码



load\_data\_full\_sqoop.sh

load\_data\_incr\_sqoop.sh

这两个文件中的所有的用户名和密码都更改为自己的

并且更改sqoop的配置sqoop-site.xml文件

将这段配置去掉注释,执行一次sqoop的任务,记录客户端的密码

|  |
| --- |
| <property>      <name>sqoop.metastore.client.record.password</name>      <value>true</value>      <description>If true, allow saved passwords in the metastore.      </description>  </property> |

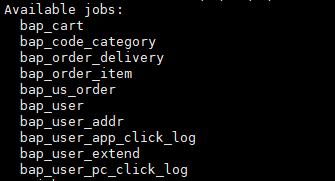
chmod +x \*.sh

./load\_data\_full\_sqoop.sh

./load\_data\_incr\_sqoop.sh

执行完后,查看命令

sqoop job –list 看一下sqoop任务是否创建成功了



注:说明\*\*\*\_full\_\*\*\*是全量,\*\*\*\_incr\_\*\*\*是增量

## ods层

从mysql数据库使用sqoop将数据抽取到hive的ods层

有两个脚本:

load\_data\_full.sh

load\_data\_incr.sh

## dwd层

从hive的ods层将数据抽取到hive的dwd层

有两个脚本

load\_data\_dwd.sh

load\_data\_dwd.sql

## dws层

从hive的dwd层抽取数据到hive的dwd层

有两个脚本

load\_data\_dws.sh

load\_data\_dws.sql

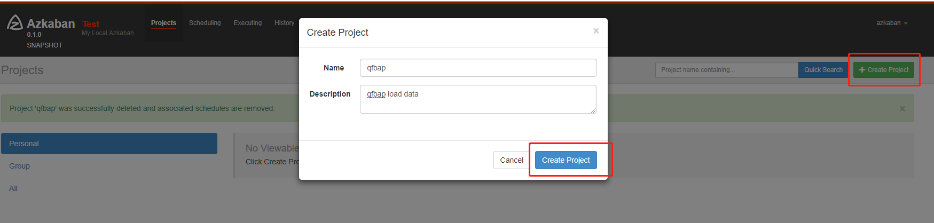
## dm层

从hive的dws层计算数据到hive的dm层和mysql的dm层

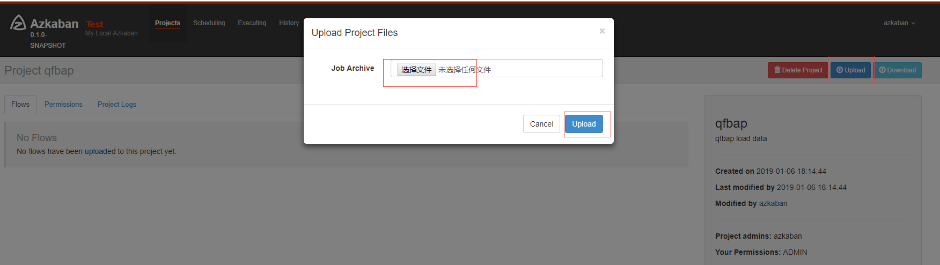
spark任务

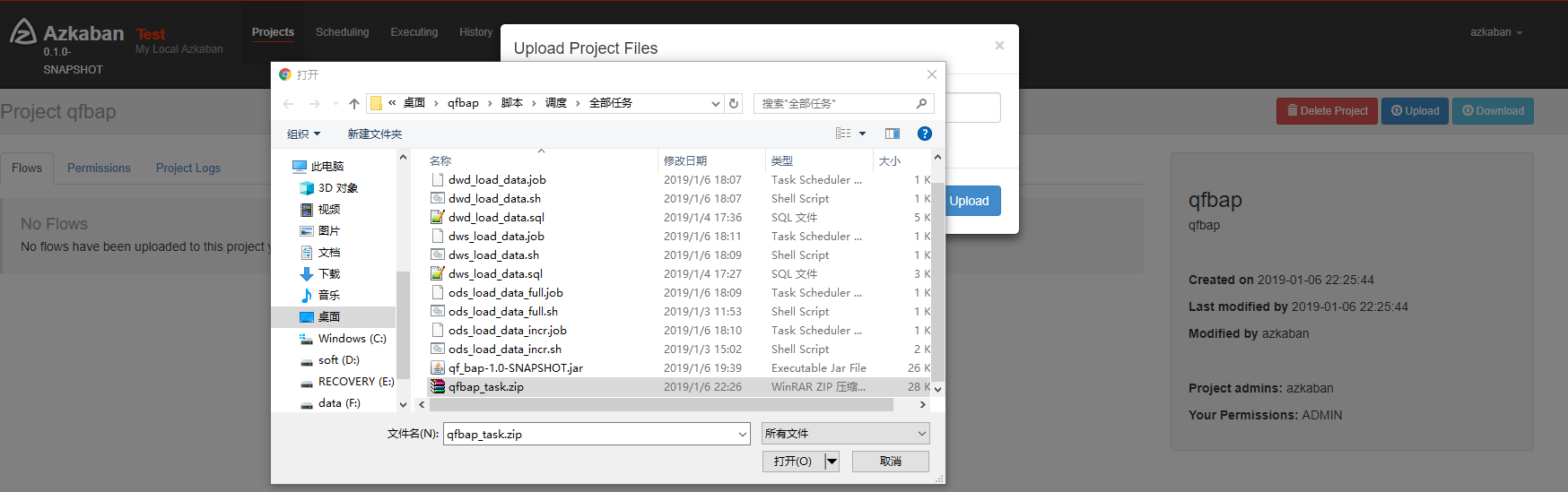
## 调度任务

1. 创建一个任务

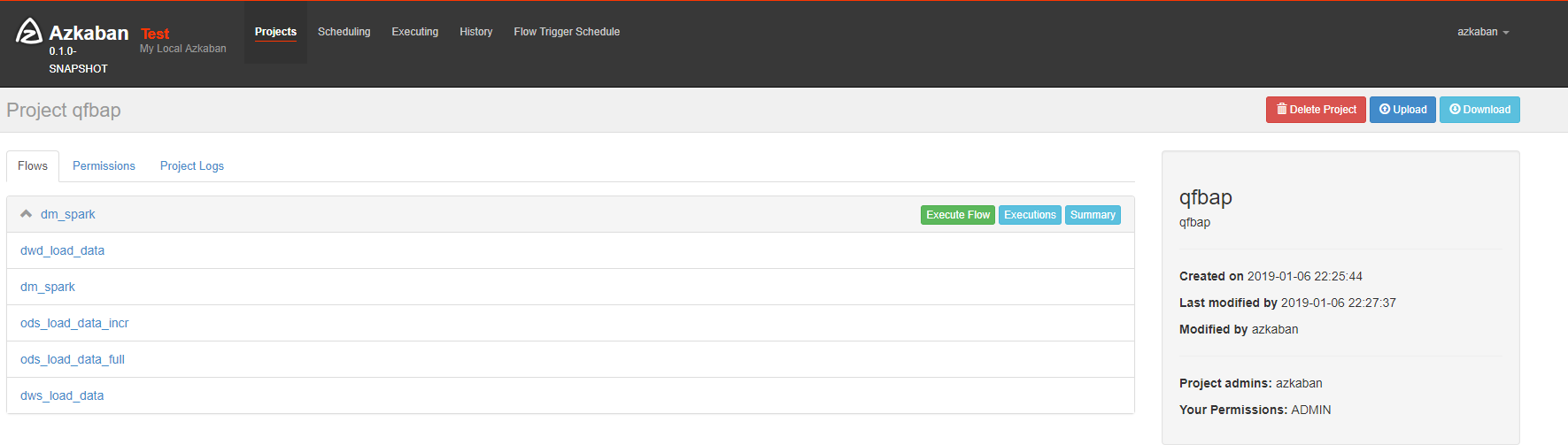


1. 上传任务

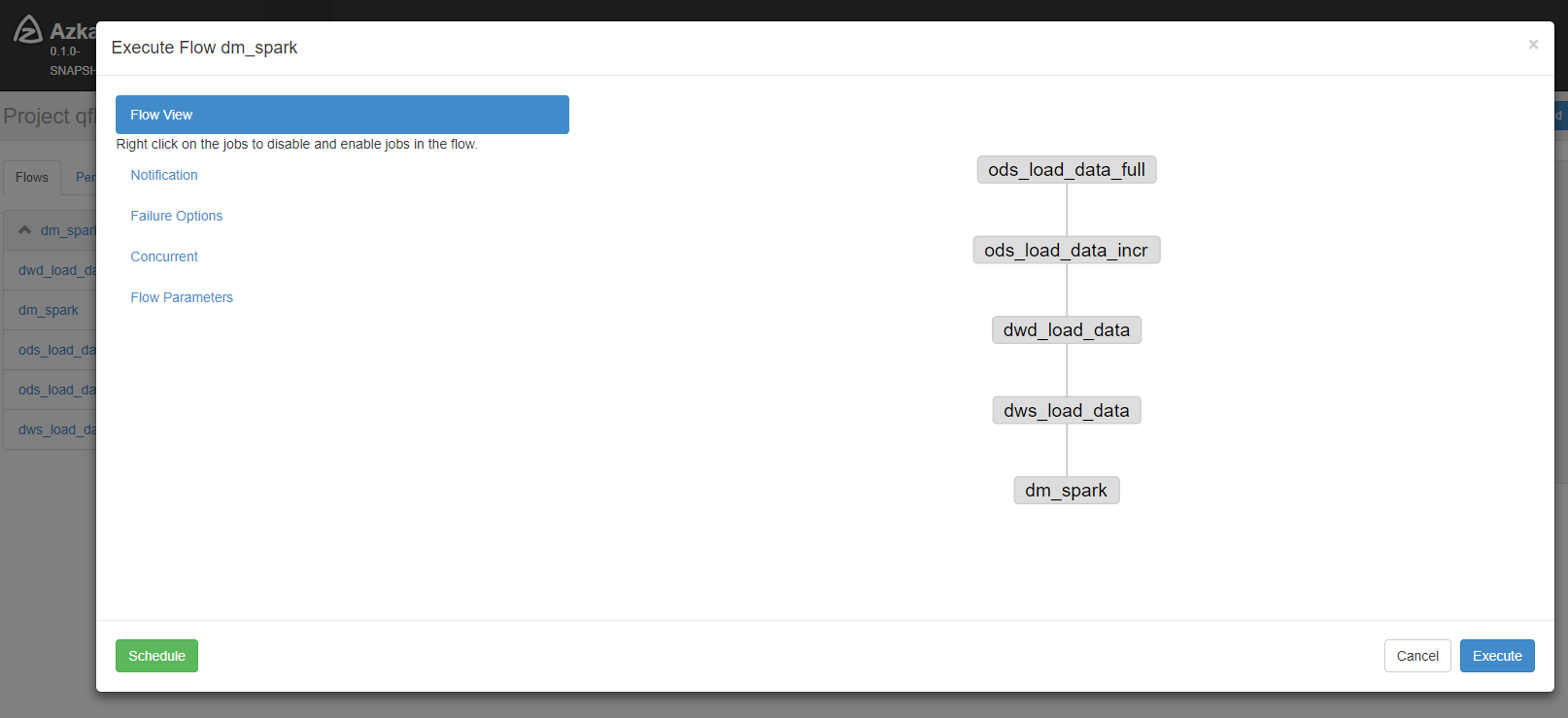




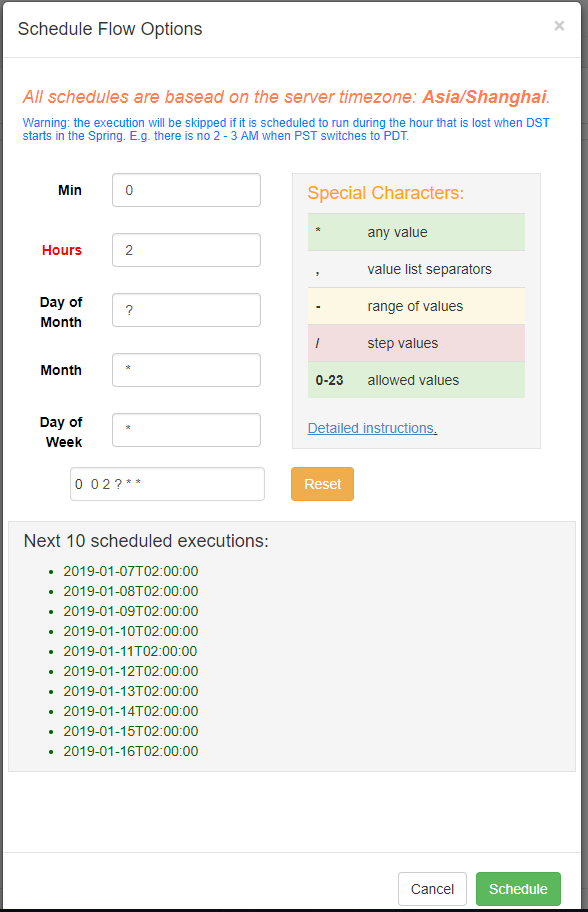
1. 上传后可以看到有ods层的,dwd层的和dws层的以及dm层的spark任务



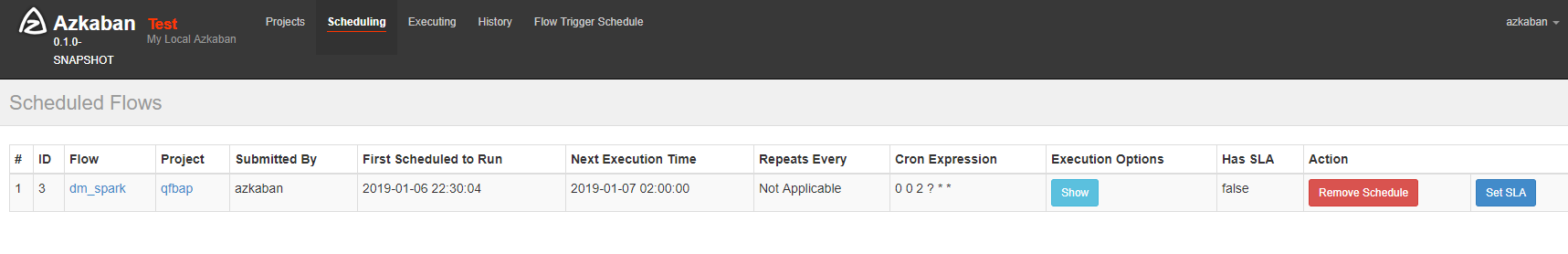
1. 点上面红框的按钮,执行,可以看到任务的流程



1. 点schedule按钮,设置调度计划,设置每天凌晨2点执行一次



调度任务就执行完成了



每天执行一次,凌晨2点执行.

# 数据展示