**京东分析师(JA)报表开发流程**

# 总体流程简介



# 权限申请

因为审批的大佬们都是很忙的，申请权限的时候，在提交在线申请、邮件申请后请使用咚咚主动联系相关责任人、审批人，推进审批进度。另外，随着时间的推移，本文档的内容可能会不准确，请以相关页面的说明为准。

京东大数据平台(BDP)wiki地址：<http://biwiki.bdp.jd.com/w/index.php?title=%E9%A6%96%E9%A1%B5>

## 大数据专家认证考试

通过大数据专家认证考试是使用京东大数据平台的前提条件，因此第一步就是参加考试啦。

考试报名地址及官方学习资料：<http://biwiki.bdp.jd.com/w/index.php?title=%E4%BA%AC%E4%B8%9C%E5%A4%A7%E6%95%B0%E6%8D%AE%E4%B8%93%E5%AE%B6%E8%AE%A4%E8%AF%81>

练习题及解答：

Svn：\operation\trunk\operation\_docs\01 类目运营产品开发组\01 项目\03 统计分析系统\JD大数据专家认证\大数据专家认证初级练习题自测\_刘宇.docx

## 数据集市权限申请

申请地址：<http://bdp.jd.com/audit/auth/start/ide0001.html>

1. 输入erp后其他内容自动填充。

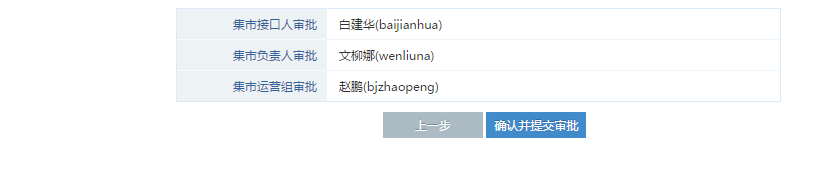


1. 选择微信手Q集市。



1. 确认信息后点击下方“确认并提交审批”。





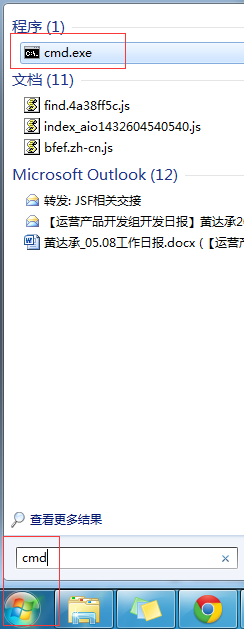
## 堡垒机权限申请

申请地址：<http://bdp.jd.com/audit/auth/start/bastion.html>

1.除了填写erp外还需要填写你机器的MAC地址，勾选“已签署”保密协议。



2.MAC地址获取方法：cmd：ipconfig –all

3.按照图示添写运营配置信息



4.确认信息后点击“确认并提交审批”



## 调度平台权限申请

1.申请方式为在线申请，说明地址：<http://biwiki.bdp.jd.com/w/index.php?title=%E8%B0%83%E5%BA%A6%E5%B9%B3%E5%8F%B0%E6%9D%83%E9%99%90%E7%94%B3%E8%AF%B7>

2.内容：

附节点编号：BJHC-JDW-Client-170129\_mart\_wxsq

需要哪些操作:“创建调度任务+查询修改自己的任务+历史补数+变量维护”，方便进行数据重跑。



## 2.5 京东分析师权限申请

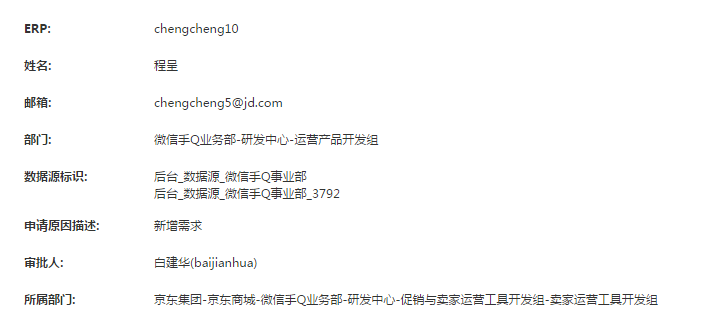
1.申请方式为在线申请：

在京东大数据平台的流程中心新建申请，流程中心的地址为：

http://bdp.jd.com/atom/workflow/apply/list.html。

1.1**开发权限申请：**

在申请流程列表中选择[京东分析师开发权限申请](http://bdp.jd.com/atom/workflow/apply/add.html?workFlowId=10038)，



重要的是数据源标识，需要填写：

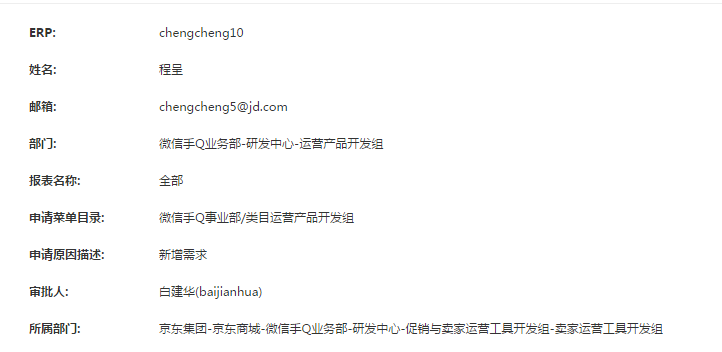
后台\_数据源\_微信手Q事业部

后台\_数据源\_微信手Q事业部\_3792

审批人选择白建华，审批通过之后找孙爱林帮忙执行，添加权限

**1.2查询权限申请：**

在申请流程列表选择[京东分析师查看权限申请](http://bdp.jd.com/atom/workflow/apply/add.html?workFlowId=10037)：



大致和开发权限的填写一样，报表名称填写全部

除了查询自己开发的报表，有可能还需要查询别人开发的报表，比如对现有报表新增需求的时候，这时候就需要申请查询权限了。

# 开发py脚本

## 3.1 py脚本功能：

从第一章JA报表开发流程可以看到，我们首先需要从gdm(通用数据层)表中抽取需要的数据，插入到一个(新建的)app表中，这一功能就是py脚本实现的。

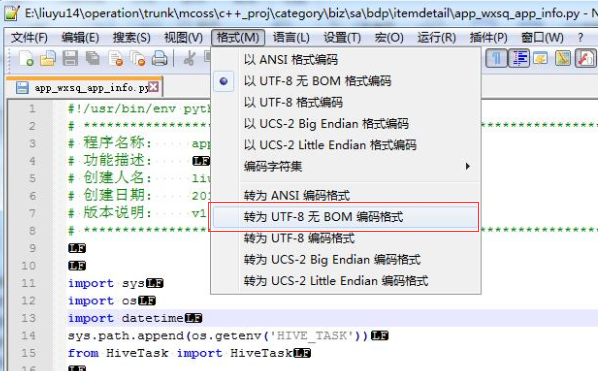
## svn脚本存放路径

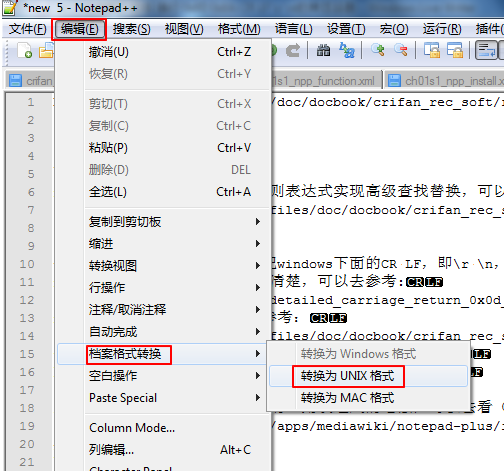
operation\trunk\mcoss\c++\_proj\category\biz\sa\bdp\itemdetail\

## 脚本编码

UTF-8，并且转换为Unix格式。

附notepad++方法：





## 3.4 脚本内容

主要包括建表与插值，可参考其他同学开发好的py脚本，主要修改sql语句，其他基本不用改。

其他内容可参考附录中的脚本规范。

# 堡垒机部署

## 4.1 堡垒机使用手册

<http://biwiki.bdp.jd.com/w/index.php?title=%E5%A0%A1%E5%9E%92%E6%9C%BA%E4%BD%BF%E7%94%A8%E6%89%8B%E5%86%8C>

## 4.2 堡垒机使用方法

1. 进入https://172.17.44.71页面，登录用户名为ERP，新用户的默认密码为123456，登录时须修改密码才可以正常使用。

2. 下载客户端（secureCRT）后，登录用户名为ERP，密码为自己页面手动修改的密码，ip为堡垒机ip 172.17.44.71

3. 客户端终端（Terminal）调成Xterm

## 4.3 脚本部署目录

/home/mart\_wxsq/bin/mcoss

## 4.4 增加脚本执行权限

Chmod +x 脚本名.py

# 调度平台调度

从Hive中抽取数据最终落地到Mysql中都需要在调度平台上创建任务来完成，也是JA报表开发过程中最关键的部分。

主要分为两个任务，一个任务是从Hive gdm表中抽取需要的数据并插入Hive app表中，这个任务的sql语句已经在py脚本中开发完成，只需在任务中执行一下脚本就好。

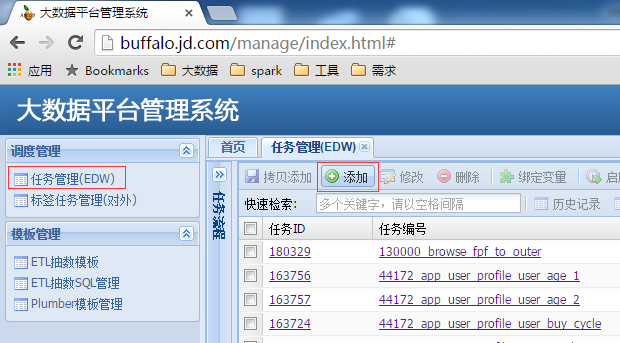
另一个任务是从我们新建的Hive app表中拉取数据落地到Mysql表中，这个任务的sql语句直接在任务模板中填写。

## 5.1 调度平台地址：

<http://buffalo.jd.com/manage/index.html>

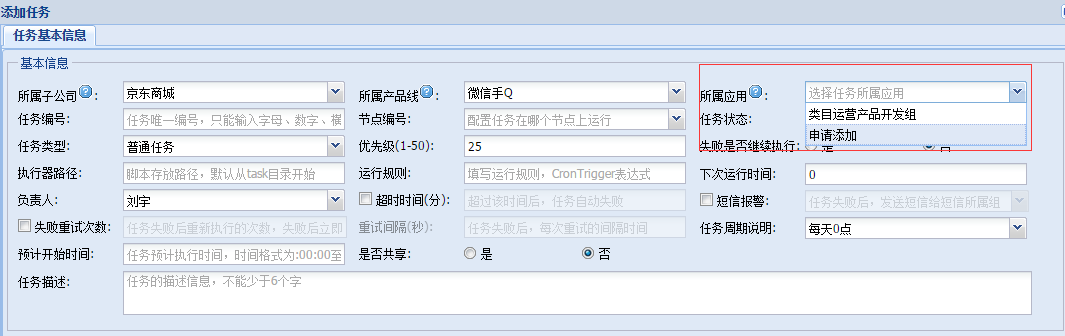
## Hive gdm表→Hive app表 任务

1.点击“任务管理”——“添加”



2.填写任务信息

这里注意，第一次使用调度平台时，“所属应用”这一栏的下拉选项中应该只有“申请添加”。点击“申请添加”后，会跳转到“添加应用”页面，在“类目运营产品开发组”后面点击“申请加入”即可。（图中没有显示“申请加入”按钮，因为我已经加入啦~）

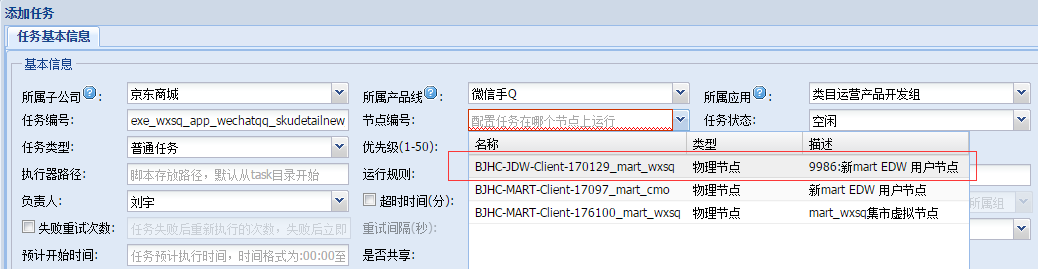




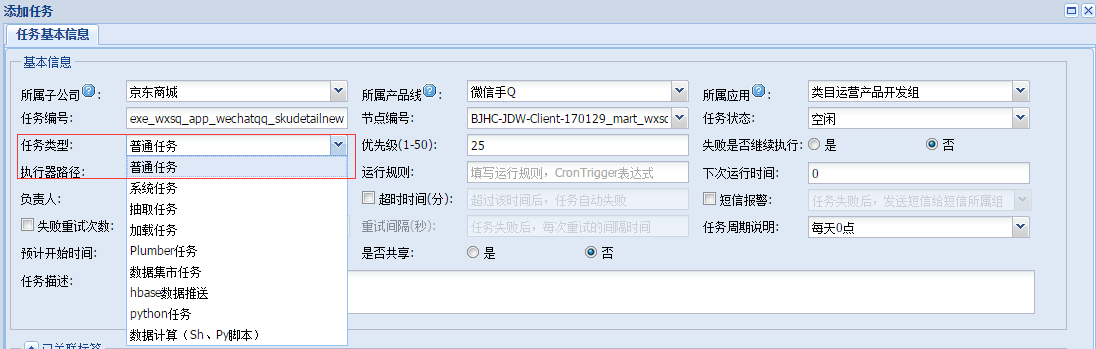
之后就可以选择“所属应用”了，

3.“任务编号”格式：exe\_wxsq\_表名

4.“节点编号”选择我们申请权限时的节点：BJHC-JDW-Client-170129\_mart\_wxsq



5.这里我们需要执行py脚本的任务，“任务类型”选择“普通任务”，“优先级”写默认的“25”就好，“失败是否继续执行”表示任务失败的话下次时间节点还是否运行任务。



6.“执行器路径”：脚本部署的路径，注意要填相对路径，

edw/etl/app/bin/mcoss/脚本名.py

7.“运行规则”：填写CronTrigger表达式。

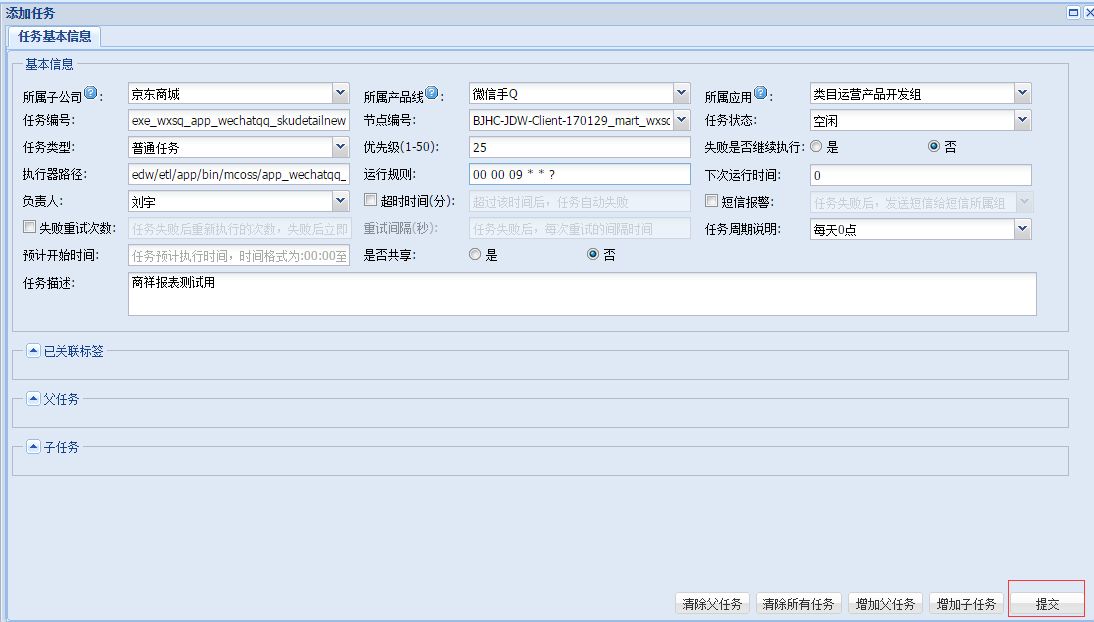
8.“下次运行时间”不用填写，会自动生成。

9.“负责人”选自己。

10.“是否共享”：如果选择“是”，则别人也能看到、修改你的任务。

11.“任务描述”：写上任务的描述信息。

12.任务信息填写完成后，点击右下方“提交”按钮即可。

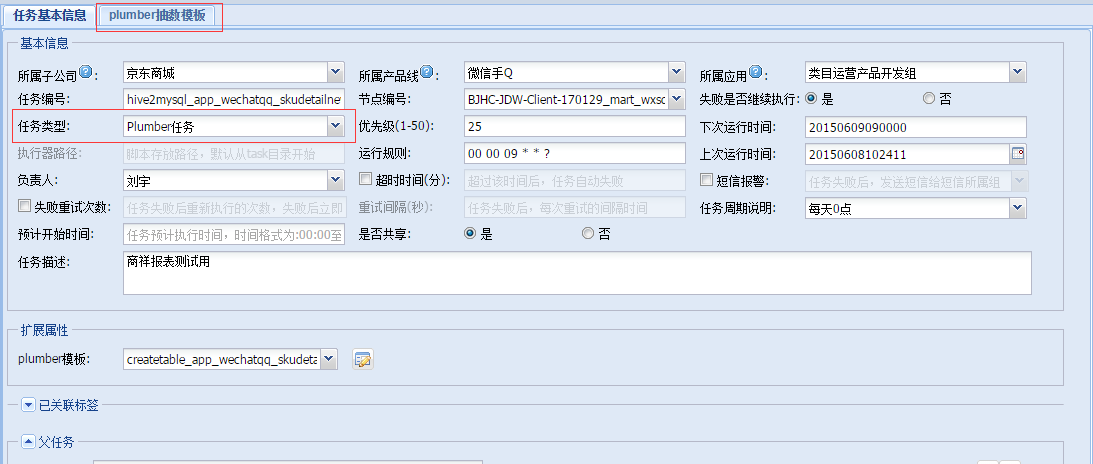


## Hive app表→Mysql 表 任务

仍然是按照5.2中创建调度任务的流程，这里只说下创建导入Mysql任务的不同点。

1.“任务编号”：hive2mysql\_表名

2.“任务类型”：Plumber任务，选择Plumber任务类型后，可以看到左上方多出一个标签页“plumber抽数模板”，点击该标签。

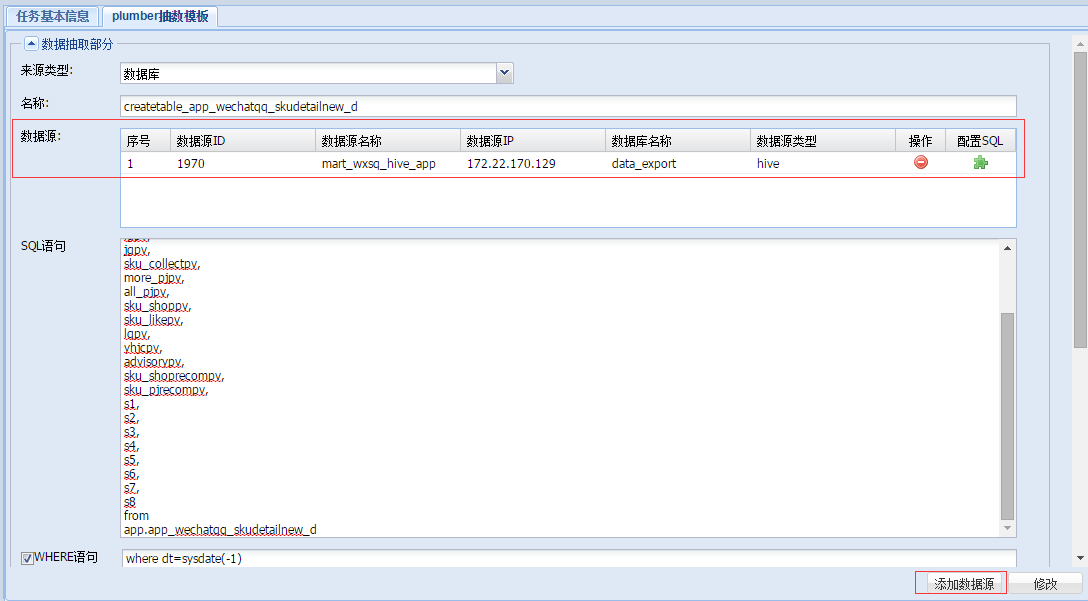


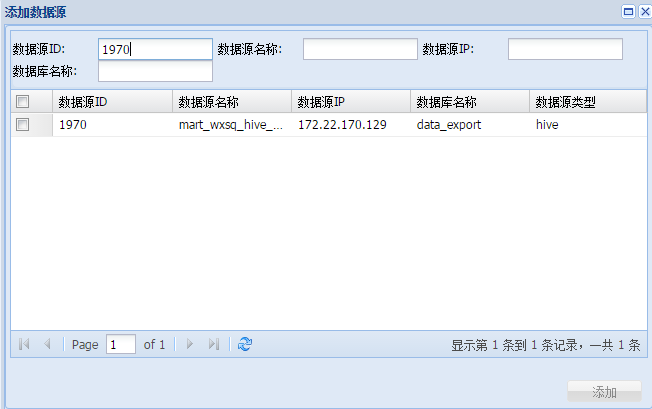
3.接下来填写plumber抽数模板信息：

(1)“来源类型”：数据库。

(2)“名称”：功能\_表名。

(3)“数据源”：点击右下角“添加数据源”按钮，添加ID为1970的数据源。





(4)SQL语句：填写从Hive中的app表拉取数据的SQL语句，例如：

select

idate,

wxsqid,

pv,

uv,

lgpv,

jgpv,

sku\_collectpv,

more\_pjpv,

all\_pjpv,

sku\_shoppv,

sku\_likepv,

lqpv,

yhjcpv,

advisorypv,

sku\_shoprecompv,

sku\_pjrecompv,

s1,

s2,

s3,

s4,

s5,

s6,

s7,

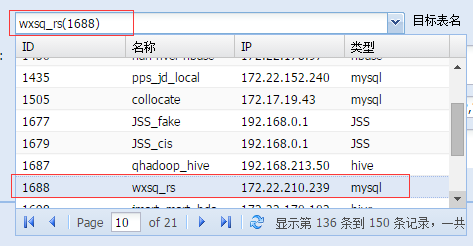
s8

from

app.app\_wechatqq\_skudetailnew\_d

(5)“WHERE语句”：例如：where dt=sysdate(-1)

(6)“目标数据源”：选择wxsq\_rs



(7)“目标表名”：直接填写表名

(8)“DELETE语句”：为保证重跑任务时能正确插入数据，每次插入数据前先执行delete语句，同时，建表语句也在这里填写。例如：

create table IF NOT EXISTS app\_wechatqq\_skudetailnew\_d (

idate varchar(20),

wxsqid int,

pv bigint,

uv bigint,

lgpv bigint,

jgpv bigint,

sku\_collectpv bigint,

more\_pjpv bigint,

all\_pjpv bigint,

sku\_shoppv bigint,

sku\_likepv bigint,

lqpv bigint,

yhjcpv bigint,

advisorypv bigint,

sku\_shoprecompv bigint,

sku\_pjrecompv bigint,

s1 varchar(20),

s2 varchar(20),

s3 varchar(20),

s4 varchar(20),

s5 varchar(20),

s6 varchar(20),

s7 varchar(20),

s8 varchar(20),

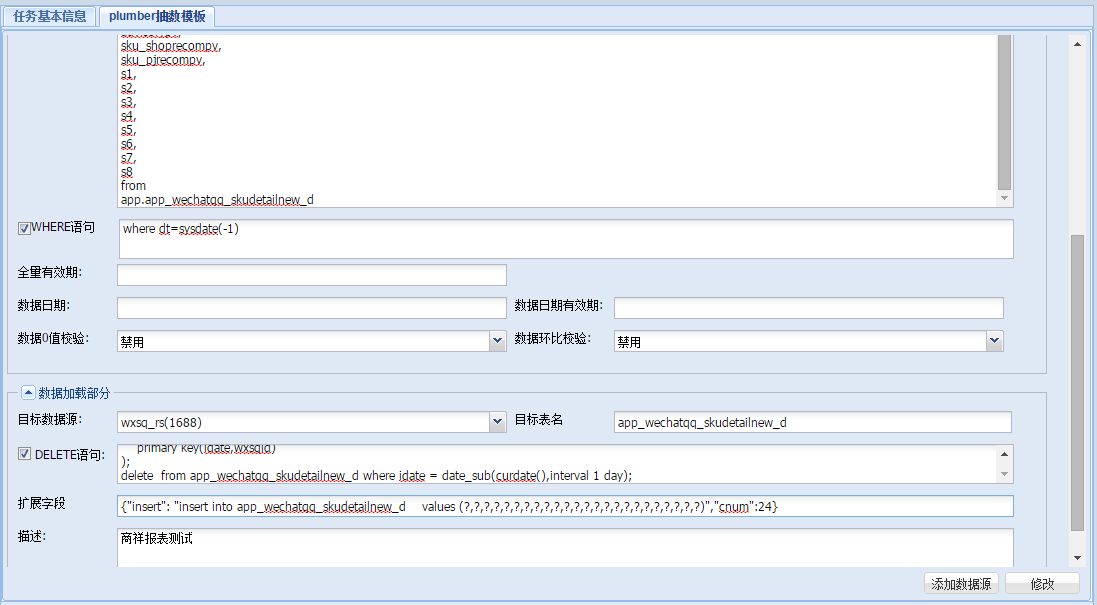
primary key(idate,wxsqid)

);

delete from app\_wechatqq\_skudetailnew\_d where idate = date\_sub(curdate(),interval 1 day);

(9)“扩展字段”：按固定格式填写insert语句，”cnum”表示有多少列数据，用”?”作为占位符，例如：

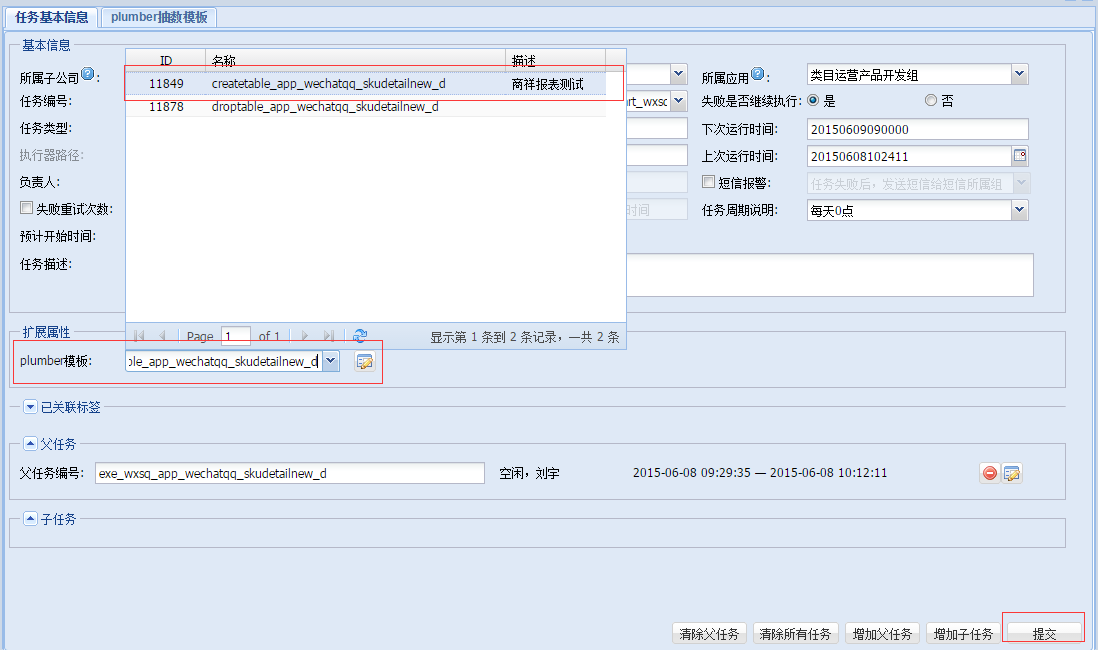
{"insert": "insert into app\_wechatqq\_skudetailnew\_d values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)","cnum":24}



(10)点击右下角“新增”按钮，即可新增抽数模板。

4.回到“任务基本信息”标签页，“plumber模板”：选择刚才新增的抽数模板。

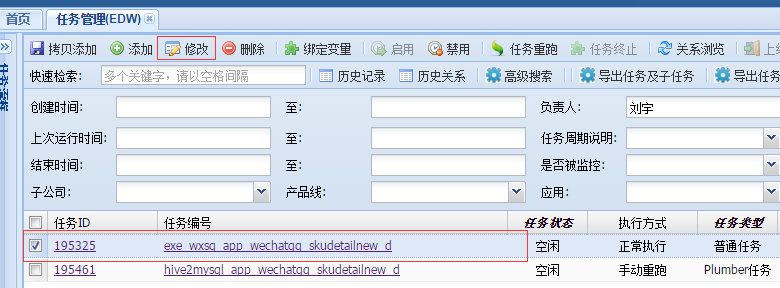
5.点击右下角“提交”，完成导入Mysql任务的创建。



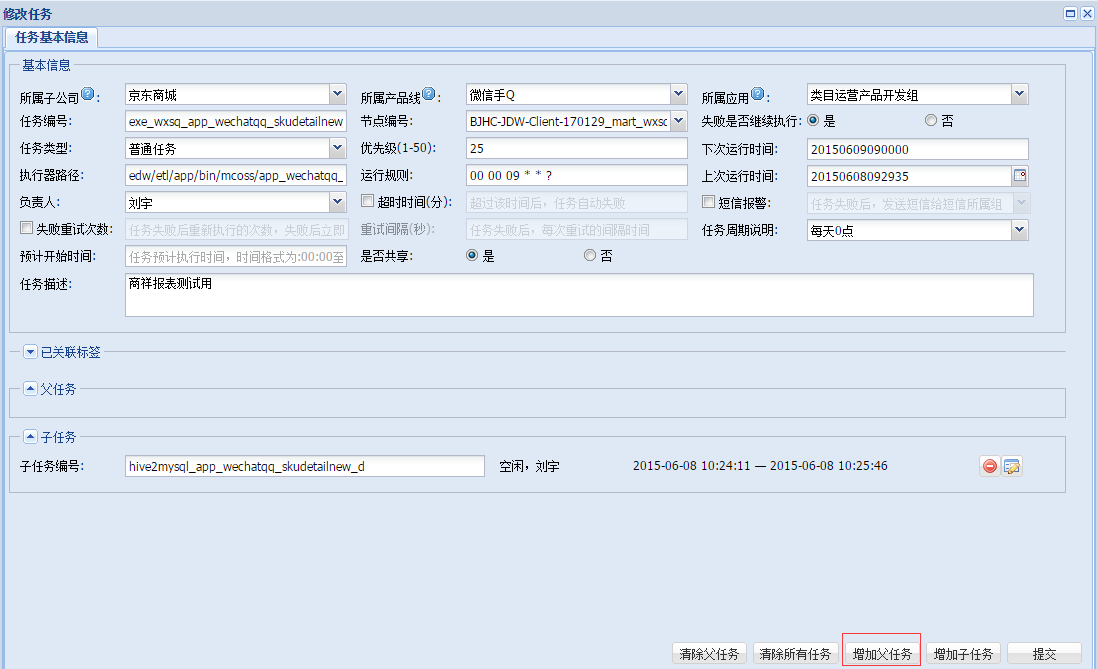
## 任务依赖关系添加

1.添加Hive原始数据的依赖。

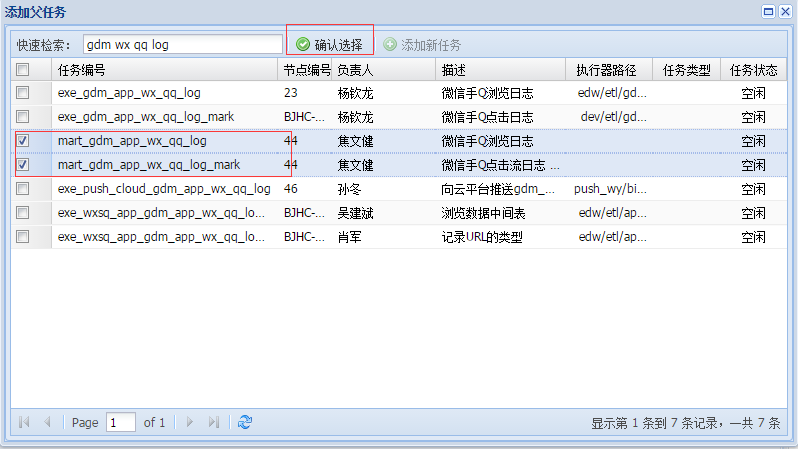
选中从Hive gdm表抽数的任务，点击“修改”。



点击右下方“增加父任务”。



选择抽取数据源的任务，点击“确认选择”。

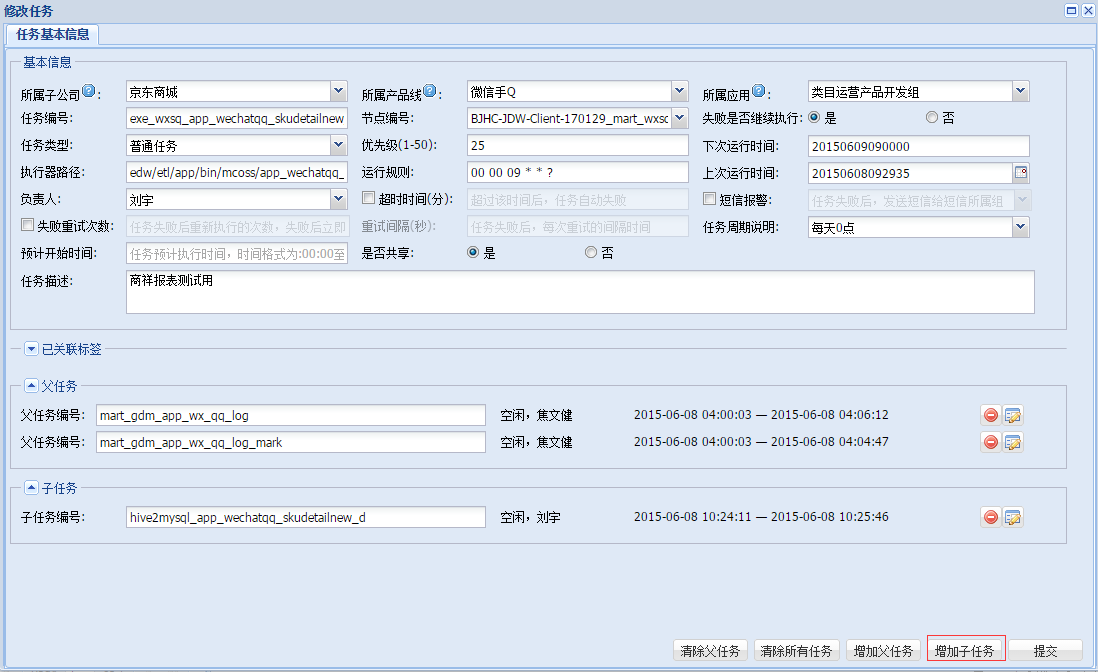


点击右下方“提交”，提交任务修改。

2.添加导入Mysql任务与Hive gdm到app抽数任务之间的依赖。

同样选择从Hive gdm表抽数的任务，点击“修改”。

点击右下角“增加子任务”，选择之前创建的导入Mysql的任务，“确认选择”并“提交”。

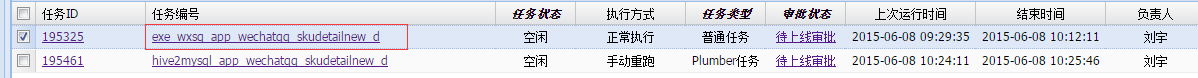


## 任务调试

1.选择需要调试的任务，点击“任务重跑”，所选任务会立即手动执行。



2.直接点击任务的“任务编号”，即可查看任务的日志





## 数据重录(需要时进行)

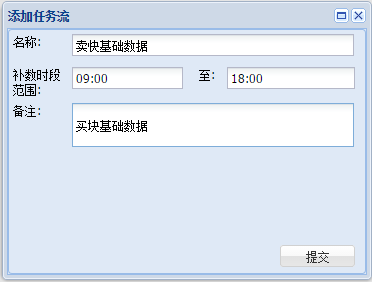
1.如果没有“历史补数”权限，首先得申请权限。



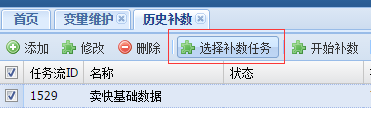
2.点击“历史补数”——“添加”



3.输入任务“名称”，“补数时段范围”选择默认的就好，指的是补数任务的执行时间，输入“备注”，点击“提交”。



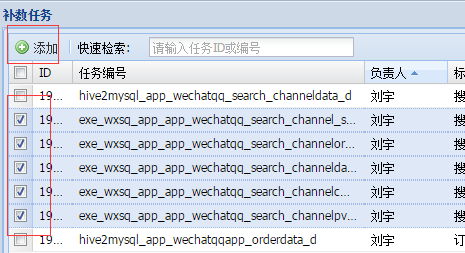
4.选中创建的任务点击“选择补数任务”



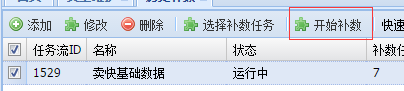
5.点击“选择补数任务”



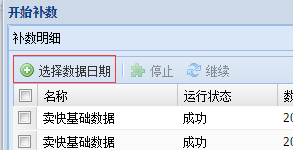
6.选择需要补数的任务后点击“添加”

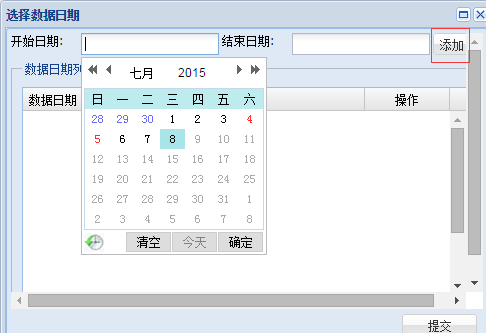


7.依次关闭窗口后返回主界面，选中任务点击“开始补数”

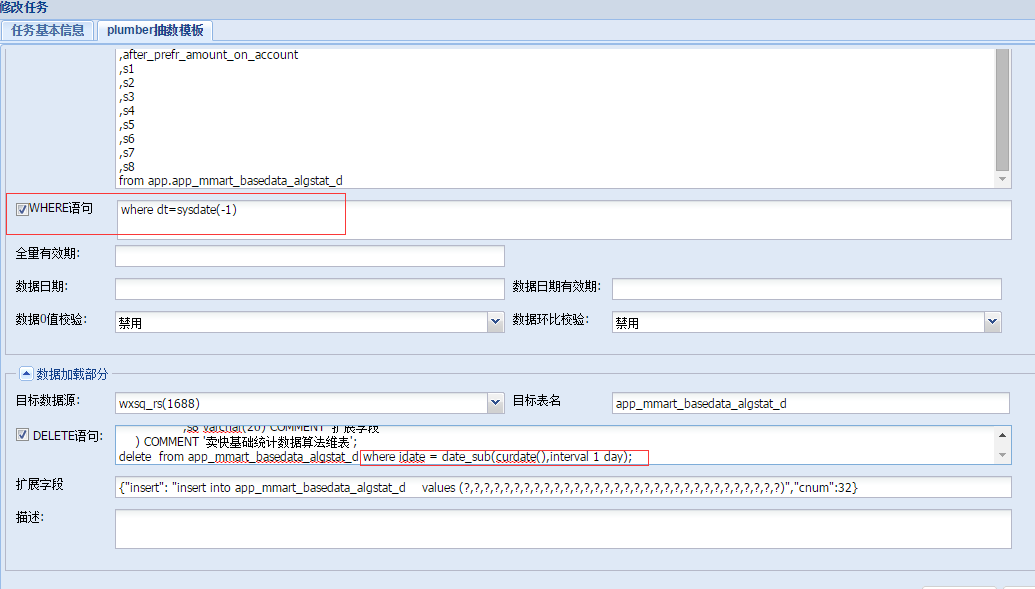


8.点击“选择数据日期”后选择“开始日期”、“结束日期”后点击“添加”即可开始数据重录。





9.重跑抽数任务，将where条件都去掉，将整个表重新导入一份。



10.重跑完抽数任务后记得将where条件加回来，切记！！

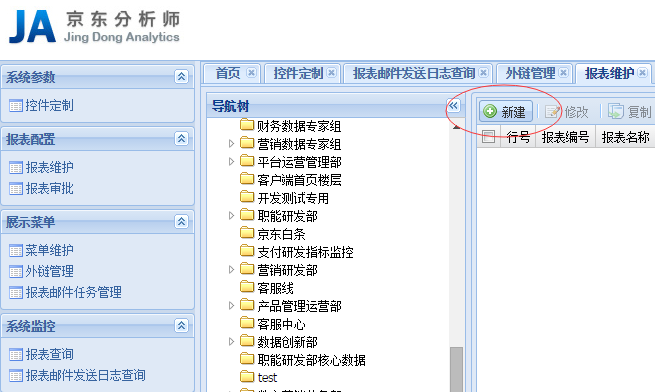
# JA报表展示

## 6.1 开发入口：<http://ja.jd.com/sys>

## 6.2 点击“报表维护”，选择目录：微信手Q事业部/类目运营产品开发组

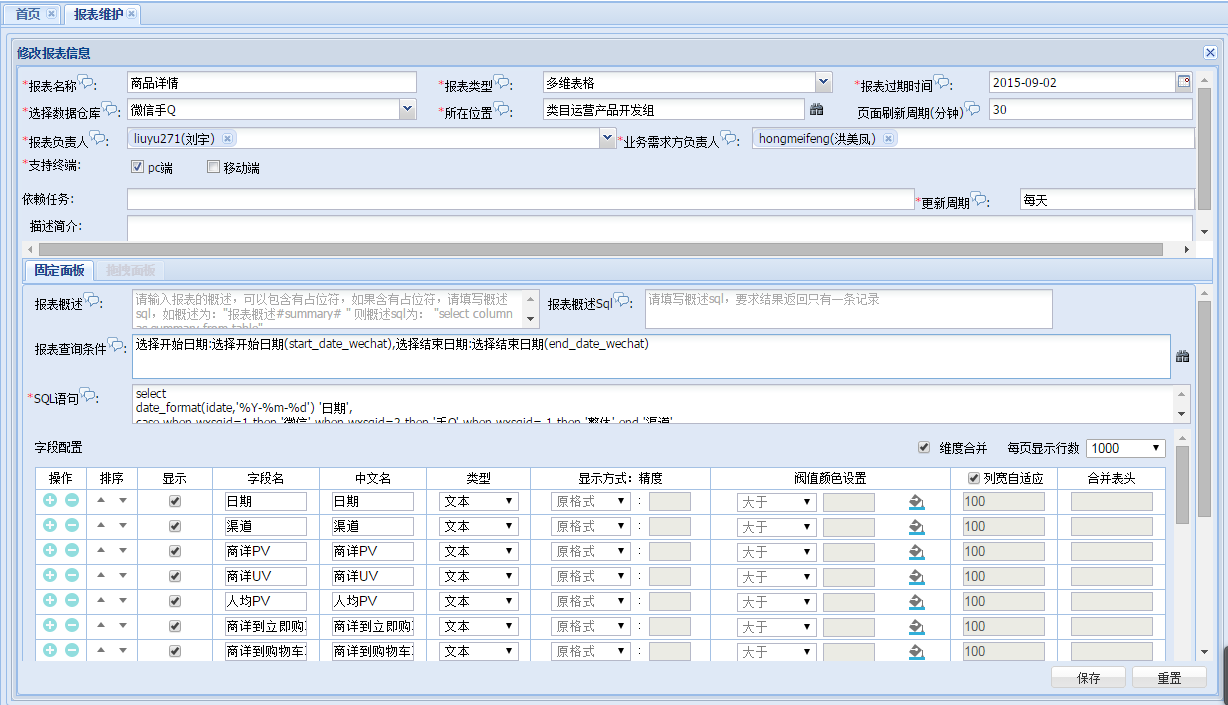


## 6.3 点击“新建”



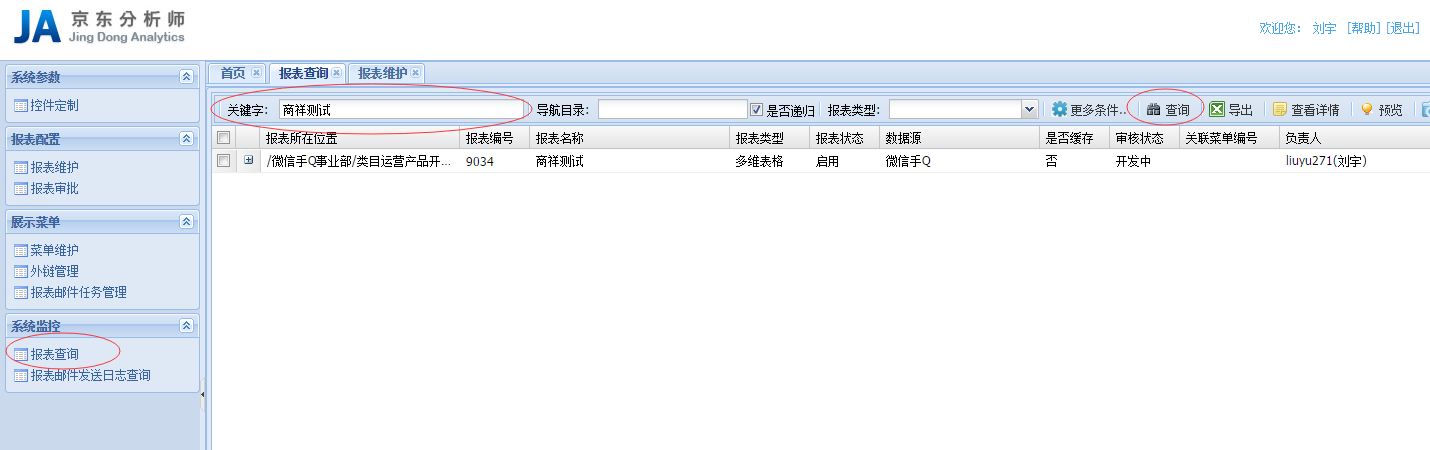
## 6.4 配置报表



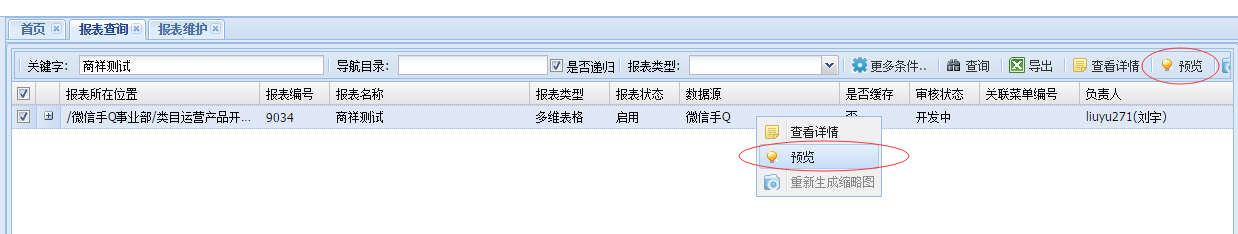


报表查询条件中可以添加控件，例如查询起始和结束日期，具体可看帮助，填写完配置后点击“保存”。

## 6.5 测试报表



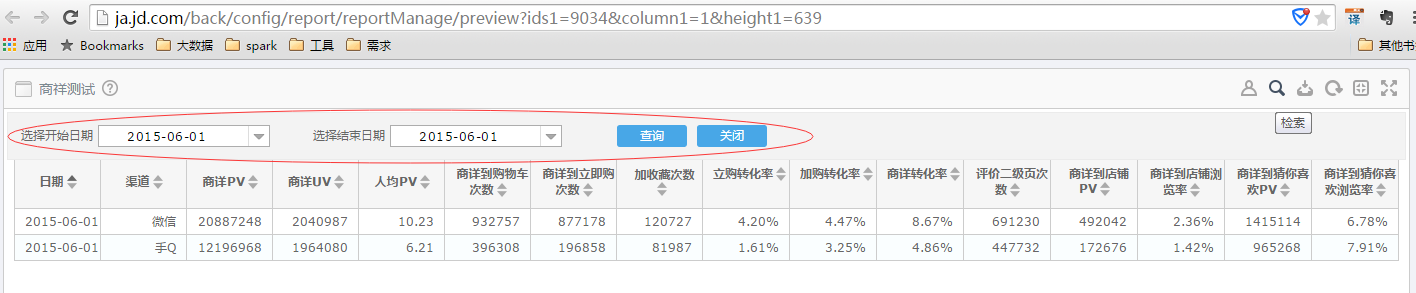
点击“报表查询”—输入关键字(如报表名)—点击“查询”。



选中报表—右键“预览”，或者点击上方工具条“预览”。

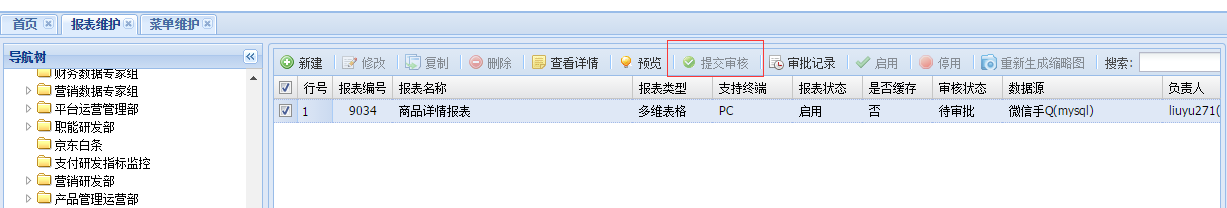


即可预览报表，点击右上方“检索”即可查看检索控件。



## 6.6 提交审批

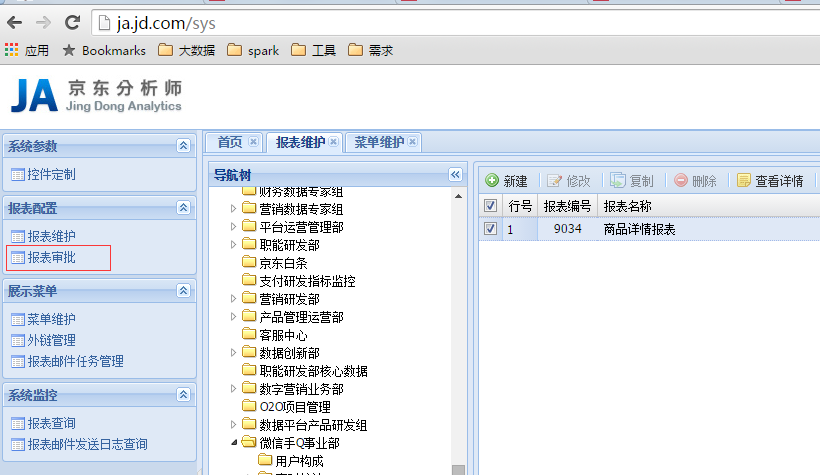
“报表维护”中点击“提交审批”。



## 6.7 审批报表

不知道为什么居然自己可以审批自己的报表。。。

1.点击“报表审批”。

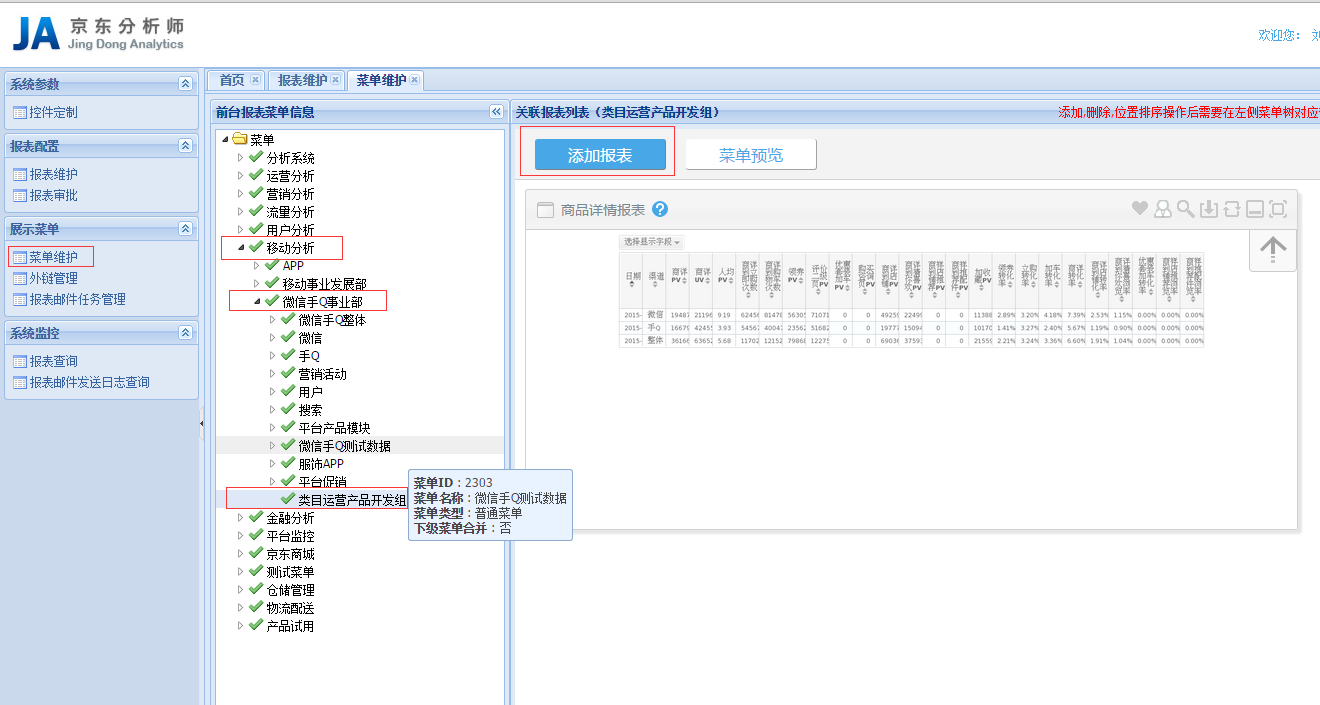


2.勾选自己提交审批的报表，点击“审批”即可。

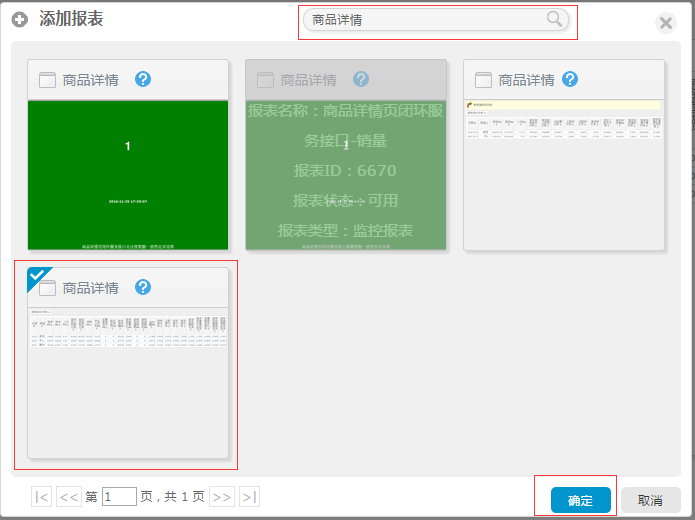


## 6.8 前台报表展示

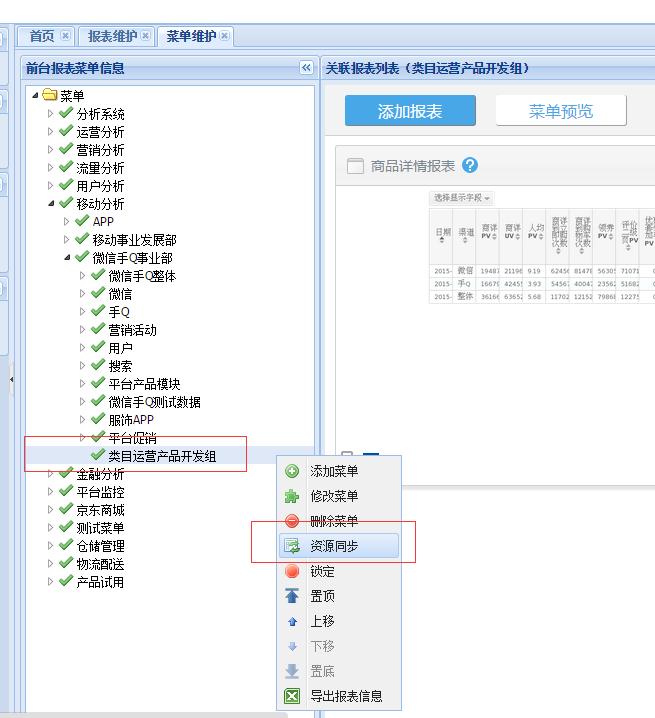
1.点击“菜单维护”，选择展示报表的目录，这里我选择的是“移动分析”——“微信手Q事业部”——“类目运营产品开发组”，点击添加报表。



2.找到自己的报表，勾选上以后点击“确定”。



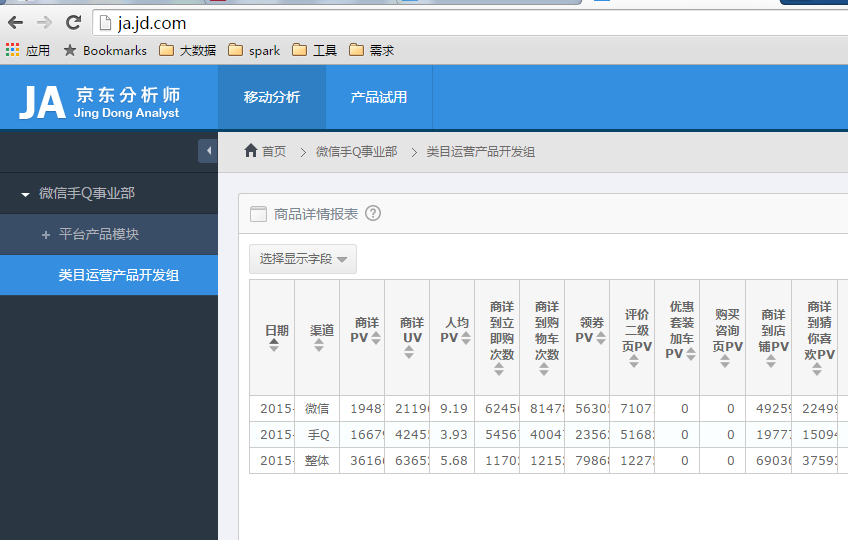
3.右键点击报表所在目录，点击“资源同步”。



## 6.9 报表前台浏览

1.前台浏览入口地址：<http://ja.jd.com/>

2.找到6.8中报表对应的目录即可浏览。

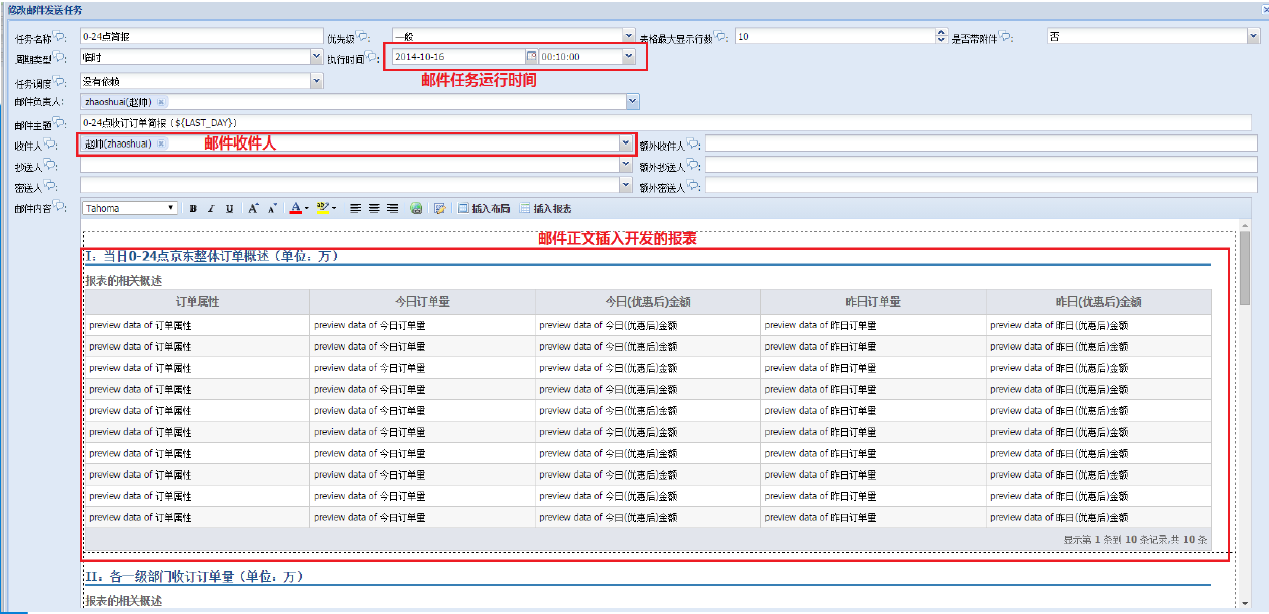


# 7. 报表邮件推送

## 7.1 报表邮件任务管理—微信手Q事业部—添加



## 配置任务



# 8. 数据录入JDW

## 8.1 数据直通车

### 8.1.1 权限申请

地址：

<http://biwiki.bdp.jd.com/w/index.php?title=%E6%95%B0%E6%8D%AE%E7%9B%B4%E9%80%9A%E8%BD%A6%E7%94%B3%E8%AF%B7%E6%B5%81%E7%A8%8B>



## 8.2 邮件申请



注明这个数据是做什么用的和申请原因，[发送给颜博yanbo1@jd.com](mailto:发送给颜博yanbo1@jd.com%20) 。

参考svn：

operation\trunk\operation\_docs\01 类目运营产品开发组\01 项目\03 统计分析系统\JDW报表开发\数据接入JDW申请单

# 9. 注意事项(踩过的坑)

## 9.1 建表的时候要留足够的扩展字段，否则要走流程改表结构

地址：<http://itsv.jd.com/forms/itsm-ap2/JD%3AWorkbench/Workbench/?cacheid=230c6be>

点击“服务目录”——“DDL变更申请”。



## 9.2 尽量不要修改Hive表的结构，因为Hive没有Mysql稳定，可能出现一些奇怪的问题。

例如：新增的字段插入不了值

Hive表出现问题咨询：马建军

## 9.3 不得不修改别人原来维护的Hive表结构时，一定要事先跟负责人沟通

否则如果别人是全删全导的方式，就有可能导致Mysql表数据全部丢失。

## 9.4 python脚本操作hive时的转义

在正则表达式需要转义时需要使用8个反斜杠，例如：

split(parsed\_url\_par['pps'],'\\\\\\\\.')

## 9.5 产品同学过来反应报表金额数目对不上

可能的原因之一是：



报表导出为Excel后，科学计数都默认为文本格式，因此产品同学将此列求和后会少计算这部分数据而误以为报表数据有问题。图中第一行已经转换为数字格式，虽然这不是技术问题但我们也别被产品同学坑了…

## 9.6 任务执行时间不得超过4小时

否则会被集群强制终止。

# 10. 附录：JDW手册-开发规范：

参考链接：http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E5.AD.97.E6.AE.B5.E5.91.BD.E5.90.8D

|  |
| --- |
| **目录**   * [1 层次命名](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E5.B1.82.E6.AC.A1.E5.91.BD.E5.90.8D)   + [1.1 数据缓冲数据层](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E6.95.B0.E6.8D.AE.E7.BC.93.E5.86.B2.E6.95.B0.E6.8D.AE.E5.B1.82)   + [1.2 基础数据层](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E5.9F.BA.E7.A1.80.E6.95.B0.E6.8D.AE.E5.B1.82)   + [1.3 通用数据层](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E9.80.9A.E7.94.A8.E6.95.B0.E6.8D.AE.E5.B1.82)   + [1.4 聚合数据层](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E8.81.9A.E5.90.88.E6.95.B0.E6.8D.AE.E5.B1.82)   + [1.5 临时层](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E4.B8.B4.E6.97.B6.E5.B1.82)   + [1.6 维度层](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E7.BB.B4.E5.BA.A6.E5.B1.82)   + [1.7 其他层次](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E5.85.B6.E4.BB.96.E5.B1.82.E6.AC.A1) * [2 字段命名](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E5.AD.97.E6.AE.B5.E5.91.BD.E5.90.8D) * [3 任务命名](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E4.BB.BB.E5.8A.A1.E5.91.BD.E5.90.8D)   + [3.1 落地方式ETL](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E8.90.BD.E5.9C.B0.E6.96.B9.E5.BC.8FETL)   + [3.2 plumber抽数方式](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#plumber.E6.8A.BD.E6.95.B0.E6.96.B9.E5.BC.8F)   + [3.3 bdm,fdm,gdm,adm等任务及表](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#bdm.2Cfdm.2Cgdm.2Cadm.E7.AD.89.E4.BB.BB.E5.8A.A1.E5.8F.8A.E8.A1.A8)   + [3.4 落地方式推数](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E8.90.BD.E5.9C.B0.E6.96.B9.E5.BC.8F.E6.8E.A8.E6.95.B0)   + [3.5 plumber推数方式](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#plumber.E6.8E.A8.E6.95.B0.E6.96.B9.E5.BC.8F)   + [3.6 集群推往集市任务](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E9.9B.86.E7.BE.A4.E6.8E.A8.E5.BE.80.E9.9B.86.E5.B8.82.E4.BB.BB.E5.8A.A1) * [4 建表规范](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E5.BB.BA.E8.A1.A8.E8.A7.84.E8.8C.83) * [5 数据类型标准](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E6.95.B0.E6.8D.AE.E7.B1.BB.E5.9E.8B.E6.A0.87.E5.87.86)   + [5.1 数据类型定义遵循原则](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E6.95.B0.E6.8D.AE.E7.B1.BB.E5.9E.8B.E5.AE.9A.E4.B9.89.E9.81.B5.E5.BE.AA.E5.8E.9F.E5.88.99)   + [5.2 HIVE支持的数据类型](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#HIVE.E6.94.AF.E6.8C.81.E7.9A.84.E6.95.B0.E6.8D.AE.E7.B1.BB.E5.9E.8B)   + [5.3 原子数据类型](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E5.8E.9F.E5.AD.90.E6.95.B0.E6.8D.AE.E7.B1.BB.E5.9E.8B)   + [5.4 复杂数据类型](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E5.A4.8D.E6.9D.82.E6.95.B0.E6.8D.AE.E7.B1.BB.E5.9E.8B) * [6 脚本规范](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#.E8.84.9A.E6.9C.AC.E8.A7.84.E8.8C.83)   + [6.1 python示例.py](http://biwiki.bdp.jd.com/w/index.php?title=JDW%E6%89%8B%E5%86%8C-%E5%BC%80%E5%8F%91%E8%A7%84%E8%8C%83#python.E7.A4.BA.E4.BE.8B.py) |

## 层次命名

命名规范包括对数据仓库数据架构中的各层次的命名规划。   
在命名过程中需要建立一个统一的中英文词库，在翻译过程中，首先到词库中查找相应的单词缩写，然后根据表名和字段名的命名规则进行命名。表名和字段名在用词、词缩写方式上都应保持一致，避免出现同一词有多种缩写的情况。

### 数据缓冲数据层

数据缓冲层规则：表名= BDM\_源库名称\_源表名

### 基础数据层

基础数据层规则：表名= FDM + 源库名称 +源表名+ 加载策略   
加载策略：拉链抽取\_chain，非拉链\_无

### 通用数据层

通用数据层规则：表名= GDM + 主题前缀 + 主体   
按照LDM中对主题的分类、主题编号，我们对每一主题中的表在物理命名时会加上一个前缀，以方便区分来自哪一个主题。主体是对业务分析范围或者源表名称的简写。   
表格 4通用语义层主题前缀及简称列表

|  |  |  |  |
| --- | --- | --- | --- |
| **主题前缀** | **主题名称** | **简称** | **业务覆盖范围** |
| M01 | 客户 | CUS | 供应商、POP商家、团购商家、合作伙伴、用户 |
| M02 | 组织机构 | ORG | 员工、部门 |
| M03 | 商品 | ITM |  |
| M04 | 销售订单 | ORD | 订单相关 |
| M05 | 账户 | ACT | 与账户关联的礼品卡、余额、积分、优惠券等 |
| M06 | 客户端 | CLI | 移动客户端、PC客户端、移动设备 |
| M07 | 营销 | CAM | 促销、活动、优惠券、礼品卡 |
| M08 | 仓储 | INV |  |
| M09 | 配送 | DIS |  |
| M10 | 客服 | CSC | 售后、备件库、呼叫中心、工单 |
| M11 | 地理区域 | LOC |  |
| M12 | 财务 | FIN |  |
| M13 | 社区 | COM | 用户的关注、订阅 |
| M14 | 流量 | TRA | Traffice |
| M15 | 供应链 | SCM | 采购、采购退货(退供应商) |

### 聚合数据层

聚合数据层规则：表名= ADM+ 主题英文简称 + 主体+后缀(日/周/月/季/年/其他)

* 主题英文简称：聚合层主题的简称(如DNST日百采销等)
* 主体：到词库中查找相应的单词缩写
* 后缀：D 日汇总 W周汇总 M 月汇总 Q季汇总 Y年汇总

命名格式举例： ADM\_DNST\_XXX\_D   
表格‑5聚合层主题分类

|  |  |
| --- | --- |
| **聚合层主题** | **主题英文简写** |
| 日百采销 | DNST |
| IT数码采销 | IT |
| 通讯汽车采销 | CC |
| 家电采销 | HEA |
| 图书采销 | BOOK |
| 采销 | SCM |
| 仓储 | STORE |
| 配送 | DIS |
| POP | POP |
| 售后 | AFS |
| 市场 | MKT |
| 信息部 | INFO |
| 发展战略 | DS |
| 财务 | FIN |
| 人资 | HR |
| 海外 | EPT |
| 自有品牌 | PB |

### 临时层

临时层规则：表名= TMP + 层次名 + [业务含义的表名]

### 维度层

DIM层规则：表名= DIM + 业务含义的表名（维度变化，命名区分，结构改造）

### 其他层次

在京东数据仓库中除上述分层之外，还存在APP层和集市层的特殊需求，这两层数据是在我们设计的数据仓库层次基础上，根据不同部门的需求，从数据仓库各层次中抽取需要的数据信息按照APP层和集市层的存储与命名规范生成的数据层次。   
集市层的命名规范与数据仓库各层的命名规范相对应。APP层命名规范如下：   
表名= APP + 主题英文简称+ 主体 + 后缀(日/周/月/季/年)  
主题英文简称参见：表格 4-2聚合层主题

## 字段命名

* 在数据仓库模型中，具有相同业务含义的字段在模型的不同业务范围、不同主题范围、不同的事实表中、不同的维度表中必须使用同一个中文和英文名，不允许出现同一业务含义的属性字段在数据仓库使用不同的中英文名称；
* 构成字段的英文名称中每个单词字母采用小写，单词间用下划线分割；
* 对于编号类字段，一般源系统有业务含义的编号用“××号”命名，无业务含义的编号用“××编号”命名，由数据仓库中生成的编号字段也用“××编号”命名。当中文用“号”时，英文名后缀一般用num或no，中文用“编号”时，英文名后缀一般用id。对于协议主题、内部机构主题和财务主题的标识符，我们习惯上用num或no，如：协议号agmt\_num、凭证号vou\_num、卡号card\_no、内部机构号int\_org\_num、会计科目号gl\_num等；其他主题的标识符则习惯上用id，例如：当事人编号pty\_id、商品编 item\_id、事件编号evt\_id、地址编号addr\_id、营销计划编号plan\_id、营销活动编号activity\_id、渠道编号chan\_id、申请号apply\_id等；修饰符习惯上用modif，例如：合同协议修饰符contract\_agmt\_modif；
* 字段属性的中文名称尽量保留实体的名称作为前缀，例如：“当事人编号”、“渠道编号”；
* 代码表的列属性中文名与实体的中文名保持一致，代码列属性英文名为实体名去掉前缀，每单词字母小写，相应的描述字段属性后缀是desc，如“婚姻状况描述 marg\_status\_desc”；
* 有一些按习惯称为“xx名称”，描述属性中文名可以用“xx名称”，英文名用“xx\_name”，比如“货币代码”实体，其两个字段分别为“货币代码curr\_cd”和“货币中文名称curr\_cn\_name”；
* 日期类型的字段，后缀应是dt，如“开户日期open\_dt”等；时间类型后缀应是tm，如“出库时间out\_wh\_tm”等；
* 时间拉链用“开始日期<a href="javascript:showColumn('b38fdd6404f60927ced9e0642e95e87f','start\_date');">start\_date</a>”和“结束日期end\_date”，此处的开始日期和结束日期不体现业务含义，只体现数据有效性的周期；
* 中文名称中尽量不加“和”、“的”等字眼以及标点符号、空格、斜线、减号和其他非规范文字；
* 英文名中不能出现标点符号、空格、斜线、减号等特殊字符，否则DDL语句在执行时会报错；
* 维表及字段命名，统一使用“dim\_”+名称(表名、字段名)的方式，在JDW模型各层维表命名规则原则上保持一致，衍生出的分支维表也应遵守维表命名原则

## 任务命名

### 落地方式ETL

一、ETL抽数sql管理

* 名称:sql\_dbname\_tablename

<p align="left">eg: sql\_afs\_sp\_basic\_onway\_transfer

* 文件名：dbname\_tablename

eg：afs\_sp\_basic\_onway\_transfer

* 目标表名：加载到bdm表

eg: bdm\_afs\_sp\_basic\_onway\_transfer  增量   
eg: bdm\_afs\_sp\_basic\_onway\_transfer\_da  全量

* 描述：抽取任务用途

二、ETL抽数模板

* 名称：template\_dbname\_tablename
* 是否压缩：是
* 描述:
* 模版作用
* 如果是分库，请描述抽取了多少库

三、任务管理(EDW)

1、抽数任务配置

* 任务编号：ext\_dbname\_tablename
* 任务类型：抽数任务
* 节点编号：查看节点编号分配
* 运行规则：根据抽数时间具体确定
* 如果是大量,运行规则不能一样;一样的视为不符合规范,自动禁用
* 任务描述：
* 任务作用
* 任务失败处理方法
* 抽数模版：模版名称

2、加载任务配置

* 任务编号：load\_dbname\_tablename
* 任务类型：加载任务
* 节点编号：查看节点编号分配
* 任务描述：
* 任务作用
* 任务失败处理方法
* 抽数模版：模版名称
* 增加父任务：选择依赖的抽取任务

### plumber抽数方式

一、plumber模版管理

* 名称：sql\_dbname\_tablename
* 如果原来抽取过该表，需重新抽取：

名称：sql\_dbname\_tablename\_new

* 描述：任务用途

二、 plumber抽取任务配置

* 任务编号：plumber\_dbname\_tablename
* 任务类型：plumber任务
* 节点编号：查看节点编号使用
* 运行规则：根据抽数时间具体确定
* Plumber版本更新说建附件《plumber版本迭代流程说明》

### bdm,fdm,gdm,adm等任务及表

* 表命名tablename=层级\_源系统库名\_源系统表名\_chain(非拉链不加chain)
* 任务编号：exe\_tablename  任务对应层表名
* 任务类型：普通任务
* 节点编号：专属虚拟节点
* 脚本命名: tablename.py   对应层表名
* 标准模板如下
* 调度执行路径：《》

|  |  |  |  |
| --- | --- | --- | --- |
| 默认路径 | 1级路径 | 2级路径 | 目前涉及业务 |
| $HOME/task | edw/etl | fdm/bin | 基础非拉链表 |
| gdm/bin | 宽表 |
| adm/bin | 主题汇总报表 |
| dim/bin | 维表 |
| app/bin | 应用报表 |
| edw/output/bin | | 其他脚本(同步，推送) |
| workspace/ods\_etl | | 基础拉链表 |

任务描述：

* 任务作用
* 任务失败处理方法
* 增加父任务：选择依赖的父任务

### 落地方式推数

* 从hive推往orcl：
* 抽数任务名称：ext\_hive\_adm\_s03\_item\_band\_stat
* 加载任务名称：load\_orcl\_adm\_s03\_item\_band\_stat
* 从hive推往mysql：
* 抽数任务名称：ext\_hive\_adm\_s03\_item\_band\_stat
* 加载任务名称：load\_mysql\_adm\_s03\_item\_band\_stat
* 从hive推往sqlserver：
* 抽数任务名称：ext\_hive\_adm\_s03\_item\_band\_stat
* 加载任务名称：load\_sqlserver\_adm\_s03\_item\_band\_stat
* 从hive推往jss：
* 抽数任务名称：ext\_hive\_adm\_s03\_item\_band\_stat
* 加载任务名称：load\_jss\_adm\_s03\_item\_band\_stat

### plumber推数方式

* 从hive推往orcl：      hive2orcl\_adm\_s03\_item\_band\_stat
* 从hive推往mysql：     hive2mysql\_adm\_s03\_item\_band\_stat
* 从hive推往sqlserver： hive2sqlserver\_adm\_s03\_item\_band\_stat
* 从hive推往jss：       hive2jss\_adm\_s03\_item\_band\_stat

### 集群推往集市任务

一、数据集市模版管理

* 名称：mart\_tablename   对应需要推送的集群表名
* 表名：tablename           对应推送的集群表名
* 库名：db                  对应推送的集市库名
* 同步分区：对应的集市推送的分区，
* 多个分区用enter键隔开

EG: dt=yesterday  
dt=4712-12-31

* 多层分区表：取全分区或最底层分区

EG: dp=EXPIRED/dt=yesterday/end\_date=yesterday

* 描述：模版用途
* 集市名称：勾选数据要推往的集市

二、任务管理（EDW）

* 任务编号：mart\_tablename  对应需要推送的集群表名
* 数据集市模版：模版名称
* 增加父任务：选择依赖的集群任务

## 建表规范

* 建表语句：

use gdm; create EXTERNAL table gdm\_sku\_pur\_attrib ( stat\_dt string comment '统计日期' ,ord\_item\_units double comment '下单商品件数' valid\_ord\_qtty bigint comment '有效订单量' ) comment '商品采销属性' PARTITIONED BY ( dt string ) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe' WITH SERDEPROPERTIES ( 'field.delim'='\t' ) STORED AS INPUTFORMAT "com.hadoop.mapred.DeprecatedLzoTextInputFormat" OUTPUTFORMAT "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat"

* 建表时需要加上 ‘表注释’，’字段注释’

## 数据类型标准

数据仓库系统会以多个源系统作为其数据来源，一般情况下源系统的数据类型定义会多种多样，差异较大，所以数据仓库在数据类型的定义上，一方面要参考源系统相关字段的定义，还要考虑到仓库自身的特点和要求，综合起来进行设计。 </p>

### 数据类型定义遵循原则

* 数据仓库中的字段长度尽量满足相应源系统字段中最大长度的要求，当然也会考虑字段的业务含义，对于一些源系统定义过长，而从实际业务含义又不可能有那么长的字段，由仓库自行选择一个合适的长度定义；
* 为了提高通用性，代码字段尽量不用数字型，建议多采用STRING类型；
* 数据仓库**不建议转换代码**，如必须在处理代码转换时，如果遇到一个例外取值，处理的的方式是在前面拼上一个特殊字符将其保留下来，以便后面知道这个例外取值的含义时可以进行相应恢复处理，这样就要求字段长度定义必须比源系统字段中最大长度还多2位；
* 日期类型字段由于格式多样，造成在信息加工处理过程中的格式转换复杂且易出错，因此对日期类型字段统一制定如下规范，在fdm层统一类型。
* 以“2013-01-01 13:35:23.71”为例说明日期字段类型规范内容：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期规范类型** | **格式规范** | **描述** | **举例** | **命名规范** |
| 时间戳字段 | yyyy-MM-dd HH:mm:ss | 带日期，精度到秒 | 2013-01-01 13:35:23 | xxx\_timestamp |
| 时间字段 | HH:mm:ss | 不带日期，精度到秒 | 13:35:23 | xxx\_time |
| 日期字段 | YYYY-MM-DD | 日期到天 | 2013-01-01 | xxx\_date |
| 周（年的第几周） | YYYYMMWW | 所在年份的第几周，由年份、月份和所在周数组成 | 20130101 | xxx\_yearweek |
| 周（月的第几周） | YYYYMMW | 由年份、月份和所在周组成 | 2013011 | xxx\_monthweek |
| 月份字段 | YYYYMM | 由年份和月份组成，无分隔符 | 201301 | xxx\_month |
| 季度字段 | YYYYQ | 由年份和所在季度组成 | 20131 | xxx\_quarterly |
| 年字段 | YYYY | 年份 | 2013 | xxx\_year |

* 命名中的“xxx”代表具体的业务操作说明
* Int类型的时间字段，默认日期代码
* 在默认情况下，JDW中的日期类型就以上述规范为标准，如果有特殊需求请向我们反馈，我们将完善到上述规则和数据仓库模型设计规范中
* 需要精度到毫秒级的日期类型字段，此时我们会补充如下规范内容（FDM）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期规范类型** | **格式规范** | **描述** | **举例** | **命名规范** |
| 时间戳字段  （到毫秒） | yyyy-MM-dd HH:mm:ss.fff | 带日期，  精度到3位数的毫秒 | 2013-01-01 13:35:23.713 | xxx\_3timestamp |

* 为了尽可能的保持仓库中数据类型的一致性以及规范性，建议仓库中的数据类型定义不宜过杂，使得仓库中的字段类型看起来比较整齐。

### HIVE支持的数据类型

Hive支持两种数据类型，一类是原子数据类型，一类是复杂数据类型。

### 原子数据类型

包括数值型、布尔型和字符串类型，具体如下表所示：   
表格 4-3 Hive原子数据类型：

|  |  |  |
| --- | --- | --- |
| **类型** | **描述** | **示例** |
| TINYINT | 1个字节（8位）有符号整数 | 1 |
| SMALLINT | 2字节（16位）有符号整数 | 1 |
| INT | 4字节（32位）有符号整数 | 1 |
| BIGINT | 8字节（64位）有符号整数 | 1 |
| FLOAT | 4字节（32位）单精度浮点数 | 1.0 |
| DOUBLE | 8字节（64位）双精度浮点数 | 1.0 |
| BOOLEAN | true/false | true |
| STRING | 字符串 | ‘sss’,”sss” |

### 复杂数据类型

包括数组（ARRAY）、映射（MAP）和结构体（STRUCT），具体如下表所示：   
表格 4-4 Hive复杂数据类型：

|  |  |  |
| --- | --- | --- |
| **类型** | **描述** | **示例** |
| ARRAY | 一组有序字段。字段的类型必须相同 | Array(1,2) |
| MAP | 一组无序的键/值对。键的类型必须是原子的，值可以是任何类型，同一个映射的键的类型必须相同，值得类型也必须相同 | Map(‘a’,1,’b’,2) |
| STRUCT | 一组命名的字段。字段类型可以不同 | Struct(‘a’,1,1,0) |

## 脚本规范

### python示例.py

* !/usr/bin/env python3
* ===============================================================================
* FILE: table\_name.py
* USAGE: ./table\_name.py
* DESCRIPTION:
* OPTIONS: ---
* REQUIREMENTS: ---
* BUGS: ---
* NOTES: ---
* AUTHOR: Liwenxue liwenxue@jd.com
* COMPANY: 360buy
* VERSION: 3.0
* CREATED: 05/22/2013 10:18:33 AM
* REVIEWER: ---
* REVISION: ---
* SRC\_TABLE: ---
* TGT\_TABLE: ---
* ===============================================================================

import sys

import os

sys.path.append(os.getenv('HIVE\_TASK'))

from HiveTask import HiveTask

ht = HiveTask()

if(len(sys.argv)>1):

data\_day\_str = sys.argv[1]

else:

data\_day\_str = ht.data\_day\_str

* ============================================================================================
* sys.argv: 可选 (default) sys.argv[1] '2013-03-14' '20130314' '2013-03-14 15:28:30' ...
* log\_lev: 可选 'INFO'(default) log日志级别
* ignore\_error: 可选 Ture慎重使用
* ht.data\_day: datetime.date(2013, 5, 4)
* ht.data\_day\_int: '20130504'
* ht.data\_day\_str: '2013-05-04'
* ht.logger.info('......')
* ht.UDF\_JAR
* -------------------------------------------------------------------------------------------
* ht.oneday( days=-3, sep = '|') '2013|05|04'
* =============================================================================================

str = """

drop table z\_tmp\_m04\_ord\_sum;

create table z\_tmp\_m04\_ord\_sum like dwa.dwa\_m04\_ord\_sum;

set hive.exec.dynamic.partition=true;

set hive.exec.dynamic.partition.mode=nonstrict;

INSERT overwrite table tmp.z\_tmp\_m04\_ord\_sum PARTITION (dp, dt)

select \* from dwa.dwa\_m04\_ord\_sum where dt >= '""" + ht.data\_day\_str +"""'

"""

* ht.exec\_sql(schema\_name = 'tmp', sql = str, table\_name = 'z\_tmp\_m04\_ord\_sum')

ht.exec\_sql(schema\_name = 'tmp', sql = str, table\_name = 'z\_tmp\_m04\_ord\_sum', lzo\_compress = True, lzo\_index\_path=['dp=ACTIVE/dt=4712-12-31','dp=HISTORY/dt=2013-05-04'], merge\_flag = True, merge\_part\_dir =

['dp=ACTIVE/dt=4712-12-31','dp=HISTORY/dt=2013-05-04'])

* ==============================================================================================
* schema\_name: 必选
* sql: 必选
* table\_name: 可选
* lzo\_compress: 可选 False (default)
* lzo\_index\_path: 依赖lzo\_compress可选，不需要warehouse，实例化了表后自动找到localtion
* *,[*] 压缩整个表
* Normal,
* /home/use/dd\_edw/db/table
* ['partition1','partition2']
* ['dir1','dir2']
* merge\_flag: False (default)
* merge\_part\_dir: [](default) 整个表都检测小文件
* [partition1,partition2]
* min\_size: 128Mb
* ----------------------------------------------------------------------------------------------
* max\_size: 250Mb
* ---------------------------------------------------------------------------------------------
* ht.merge\_small\_file(db, table, partition = [], min\_size = 128\*1024\*1024)
* ht.CreateIndex(db, table, path = 'Normal')
* ===============================================================================================