通用项目部署文档

1. 项目部署文件
2. 项目部署文件夹

在用户家目录下需要部署一下目录：

general/bin 存放项目部署的脚本和jar文件

general/data 存放导入集群的数据

general/log 存放项目的运行日志

general/bin/conf 存放项目运行加载的配置文件

1. 项目部署jar文件

transform-date.jar 日期转换jar

label.jar 标签jar

delete-useless.jar 删除标签jar

format.jar 分隔符转换jar

uniq.jar 数据去重jar

1. 项目部署脚本

claim.py 生成claim原始数据

cust\_order.py 生成customer原始数据和order原始数据

doss.py 生成doss原始数据

part.py 生成part原始数据,在生成part之前需要先生成order表的数据，part表需要关联order

auto\_check.sh 检查新增零件车型车系经销商信息

auto\_cretb.sh 创建需要的所有表

load\_data.sh 导入所有辅助表数据脚本

auto\_mr.sh mr计算的所有的运行程序

submit.sh 所有任务的提交脚本

zhanbi.py 计算占比类文件

fjp.py 计算非基盘类文件脚本

zjxj.py 计算直接相加类脚本

jpl.py 计算基盘类脚本

在general/bin/conf目录下部署的文件：

CHE 是auto\_mr.sh中打删除标签的时候使用文件，里面是品牌车系车型一级二级车型的文件，里面加有版本控制，一共是六列

doss\_asc mark\_doss doss表打一级二级车型一级asc\_code的时候需要使用的配置文件

engineoil filter 打维修标签使用的配置文件

mark 打剩余的出库日期等时间相关标签，以及一级二级车型使用的配置文件

1. 项目部署的辅助表数据

mon\_mapping 部署在general/data/dim

asc\_mapping,distributor,engine\_oil,local\_engineoil,local\_mark\_doss mapping,name,second\_level\_classification,CHE,doss\_asc,filter, local\_filter,maintenance,mark,primary\_classification,sexual,city enclosure,high\_flow\_parts,local\_mark,maint\_type1,mark\_doss,rovince type2 部署在general/data/dim每个对应的版本下面。

1. 项目执行

项目的执行顺序需要按照submit.sh提交顺序执行

#创建集群文件路径和本地文件路径

#sh auto\_cretb.sh mkdifile

#创建kpi计算之前的五张数据存储表

#sh auto\_cretb.sh cre\_tb\_fact

#创建kpi计算的辅助表

#sh auto\_cretb.sh cre\_tb\_dim

#创建kpi计算的表

#sh auto\_cretb.sh cre\_tb\_kpi\_tmp 8areas

#sh auto\_cretb.sh cre\_tb\_kpi\_tmp 8areas\_k1

#sh auto\_cretb.sh cre\_tb\_kpi\_other 8areas

#创建kpi计算的相关辅助表

#sh auto\_cretb.sh cre\_flow\_dim

#创建流水表

#sh auto\_cretb.sh cre\_flow

#创建数据检查表，在通用的即为生成原始数据的五张表

#sh auto\_cretb.sh cre\_check\_table

#导入辅助表数据，参数是当前的版本号

#sh auto\_cretb.sh load\_data\_dim 1

#给辅助表添加分区，分区为版本号

#sh auto\_cretb.sh add\_par 1

#为数据检测表添加分区

#sh auto\_cretb.sh add\_par\_check 201704\_201707

#存放原始数据的本地路径，通用不需要

#sh auto\_mr.sh mkdir\_local 201704\_201707

#format\_delimiter

#用于检测新增品牌车型车系，以及零件

#sh auto\_check.sh fun\_all

#导入辅助表数据，参数是当前的版本号

#sh load\_data.sh load\_data\_ff 3

#更新或者加载本地conf目录下的文件

#sh load\_data.sh load\_localfile 3

#用于再次检测新增品牌车型车系，以及零件

#sh auto\_check.sh fun\_all

#分隔符转换

#sh auto\_mr.sh format\_delimiter ${date} >${log\_path}/format\_delimiter\_part\_${date}.log 2>&1

#数据去重

#sh auto\_mr.sh uniq\_cus ${date} >${log\_path}/format\_delimiter\_cus\_${date}.log 2>&1

#sh auto\_mr.sh uniq\_doss ${date} >${log\_path}/format\_delimiter\_dosses\_${date}.log 2>&1

#sh auto\_mr.sh uniq\_order ${date} >${log\_path}/format\_delimiter\_ord\_${date}.log 2>&1

#sh auto\_mr.sh uniq\_claim ${date} >${log\_path}/format\_delimiter\_claims\_${date}.log 2>&1

#打标签

#sh auto\_mr.sh delete\_useless ${date} >${log\_path}/delete\_useless\_${date}.log 2>&1

#sh auto\_mr.sh label\_fm ${date} >${log\_path}/label\_fm\_${date}.log 2>&1

#sh auto\_mr.sh label\_fix ${date} >${log\_path}/label\_fix\_${date}\_2017-10-25\_1.log 2>&1

#sh auto\_mr.sh label\_other ${date} >${log\_path}/label\_other\_${date}.log 2>&1

#sh auto\_mr.sh label\_doss ${date} >${log\_path}/label\_doss\_${date}.log 2>&1

#sh auto\_mr.sh add\_partition 201704\_201707

source /etc/profile

startTime=`date +"%s.%N"`

#spark-submit --master yarn --driver-memory 3g --executor-memory 5g --num-executors 30 --executor-cores 1 /home/ipsos\_test4/general/bin/zjxj.py 2017-04 2017-07 ipsos\_test4 time\_record\_zjxj >../log/zjxj\_test 2>&1

#spark-submit --master yarn --driver-memory 3g --executor-memory 5g --num-executors 30 --executor-cores 1 /home/ipsos\_test4/general/bin/zhanbi.py 2017-04 2017-07 ipsos\_test4 time\_record\_zhanbi > ../log/zhanbi\_test 2>&1

#全国月度

spark-submit --master yarn --driver-memory 3g --executor-memory 5g --num-executors 30 --executor-cores 1 /home/ipsos\_test4/general/bin/fjp.py 2010-04 2017-07 ipsos\_test4 whole\_country everymonth >../log/fjp\_test 2>&1

spark-submit --master yarn --driver-memory 3g --executor-memory 5g --num-executors 30 --executor-cores 1 /home/ipsos\_test4/general/bin/jpl.py 2010-04 2017-07 ipsos\_test4 whole\_country everymonth >../log/jpl\_test 2>&1

#单店月度

endTime=`date +"%s.%N"`

echo `awk -v x1="$(echo $endTime | cut -d '.' -f 1)" -v x2="$(echo $startTime | cut -d '.' -f 1)" -v y1="$[$(echo $endTime | cut -d '.' -f 2) / 1000]" -v y2="$[$(echo $startTime | cut -d '.' -f 2) /1000]" 'BEGIN{printf "RunTime:%.1f s",(x1-x2)+(y1-y2)/1000000}'` >> ../log/time\_log