# 数据源表结构及源文件加工到数据主仓的脚本

## 数据源的文件结构

ODS结构化数据：以””为分割符的txt压缩文件，“”的ASCAII码是”\001”

O域流量数据：以”|”为分隔符的txt压缩文件。

## 源文件加工到数据主仓的脚本

hadoopfs -put /mnt/disk1/data/TG\_CDRMM\_GS/ /user/root/BONC/TG\_CDRMM\_GS#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TF\_B\_TRADE/ /user/root/BONC/TF\_B\_TRADE#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TD\_B\_ITEMS/ /user/root/BONC/TD\_B\_ITEMS#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TF\_F\_USER\_SVCSTATE/ /user/root/BONC/TF\_F\_USER\_SVCSTATE#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TF\_R\_TERMINAL\_ARCH/ /user/root/BONC/TF\_R\_TERMINAL\_ARCH#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TF\_CHL\_CHANNEL/ /user/root/BONC/TF\_CHL\_CHANNEL#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TF\_F\_USER/ /user/root/BONC/TF\_F\_USER#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TG\_CDRMM/ /user/root/BONC/TG\_CDRMM#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TG\_CDRMM\_SP/ /user/root/BONC/TG\_CDRMM\_SP#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TF\_F\_USER\_PURCHASE/ /user/root/BONC/TF\_F\_USER\_PURCHASE#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TF\_F\_USER\_PROD\_ITEM/ /user/root/BONC/TF\_F\_USER\_PROD\_ITEM#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TF\_F\_USER\_PRODUCT/ /user/root/BONC/TF\_F\_USER\_PRODUCT#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TF\_F\_USER\_PURCHASE\_ITEM/ /user/root/BONC/TF\_F\_USER\_PURCHASE\_ITEM#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/TS\_B\_BILL/ /user/root/BONC/TS\_B\_BILL#CURDATEYYYYMMDD#/

hadoopfs -put /mnt/disk1/data/gnfiles/ /user/root/BONC/gnfiles#CURDATEYYYYMMDD#/

# ODS建表语句及加工语句

ODS建表及加载，是通过建表的同时指定数据的来源路径，实现自动加载。

## GPRS清单表TG\_CDRMM\_GS

create external table IF NOT EXISTS TG\_CDRMM\_GS (

PROV\_ID STRING ,

SOURCE\_TYPE STRING ,

BIZ\_TYPE STRING ,

FID STRING ,

RR\_FLAG STRING ,

FILE\_NAME STRING ,

DEAL\_TIME STRING ,

RECORD\_TYPE STRING ,

NI\_PDP STRING ,

MSISDN STRING ,

IMSI\_NUMBER STRING ,

SGSN STRING ,

MSNC STRING ,

LAC STRING ,

RA STRING ,

CELL\_ID STRING ,

CHARGING\_ID STRING ,

GGSN STRING ,

APNNI STRING ,

APNOI STRING ,

PDP\_TYPE STRING ,

SPA STRING ,

SGSN\_CHANGE STRING ,

SGSN\_PLMN\_ID STRING ,

CAUSE\_CLOSE STRING ,

RESULT STRING ,

HOME\_AREA\_CODE STRING ,

VISIT\_AREA\_CODE STRING ,

CITY\_CODE STRING ,

VISIT\_AREA\_HOMETYPE STRING ,

USER\_TYPE STRING ,

FEE\_TYPE STRING ,

ROAM\_TYPE STRING ,

SERVICE\_TYPE STRING ,

IMEI STRING ,

START\_DATE STRING ,

START\_TIME STRING ,

CALL\_DURATION STRING ,

SERV\_ID STRING ,

SERV\_GROUP STRING ,

SERV\_DURATION bigint ,

DATA\_UP1 STRING ,

DATA\_DOWN1 STRING ,

DATA\_UP2 bigint ,

DATA\_DOWN2 bigint ,

CHARGED\_ITEM STRING ,

CHARGED\_OPERATION STRING ,

CHARGED\_UNITS STRING ,

FREE\_CODE STRING ,

BILL\_ITEM STRING ,

CFEE\_ORG bigint ,

CFEE bigint ,

DIS\_CFEE bigint ,

DFEE\_ORG bigint ,

DFEE bigint ,

DIS\_DFEE bigint ,

RECORDSEQNUM STRING ,

FILE\_NO STRING ,

ERROR\_CODE bigint ,

CUST\_ID bigint ,

USER\_ID STRING ,

A\_PRODUCT\_ID STRING ,

A\_SERV\_TYPE STRING ,

CHANNEL\_NO STRING ,

OFFICE\_CODE STRING ,

DOUBLEMODE STRING ,

OPEN\_DATETIME STRING ,

A\_USER\_STAT STRING ,

INTER\_GPRSGROUP STRING ,

APN\_GROUP STRING ,

APN\_TYPE STRING ,

TARIFF\_FEE bigint ,

RATE\_TIMES bigint ,

INDB\_TIME STRING ,

RESERVER1 STRING ,

RESERVER2 STRING ,

RESERVER3 STRING ,

RESERVER4 STRING ,

RESERVER5 STRING ,

RESERVER6 STRING ,

RESERVER7 STRING ,

RESERVER8 STRING ,

PARTITION\_ID STRING ,

UNI\_HOME\_AREA\_CODE STRING ,

UNI\_VISIT\_AREA\_CODE STRING ,

UNI\_ROAM\_CITY\_CODE STRING,

UNI\_SERVICE\_TYPE STRING ,

UNI\_SERV\_ID STRING,

UNI\_ROAM\_TYPE STRING,

UNI\_NI\_PDP STRING,

UNI\_SERV\_TYPE STRING,

UNI\_SP\_SRVTYPE STRING,

UNI\_GS\_SOURCE\_TYPE STRING

)

row format delimited fields terminated by '\001'

location '/user/root/BONC/TG\_CDRMM\_GS#CURDATEYYYYMMDD#/'

## 业务台帐主表TF\_B\_TRADE

create external table IF NOT EXISTS TF\_B\_TRADE (

TRADE\_ID STRING ,

SUBSCRIBE\_ID STRING ,

BPM\_ID STRING ,

TRADE\_TYPE\_CODE STRING ,

IN\_MODE\_CODE STRING ,

PRIORITY STRING ,

SUBSCRIBE\_STATE STRING ,

NEXT\_DEAL\_TAG STRING ,

PRODUCT\_ID STRING ,

BRAND\_CODE STRING ,

USER\_ID bigint ,

CUST\_ID bigint ,

USECUST\_ID STRING ,

ACCT\_ID STRING ,

USER\_DIFF\_CODE STRING ,

NET\_TYPE\_CODE STRING ,

SERIAL\_NUMBER STRING ,

CUST\_NAME STRING ,

ACCEPT\_DATE STRING ,

ACCEPT\_MONTH bigint ,

TRADE\_STAFF\_ID STRING ,

TRADE\_DEPART\_ID STRING ,

TRADE\_CITY\_CODE STRING ,

TRADE\_EPARCHY\_CODE STRING ,

TERM\_IP STRING ,

EPARCHY\_CODE STRING ,

CITY\_CODE STRING ,

OLCOM\_TAG STRING ,

EXEC\_TIME STRING ,

FINISH\_DATE STRING ,

OPER\_FEE bigint ,

FOREGIFT bigint ,

ADVANCE\_PAY bigint ,

INVOICE\_NO STRING ,

UNI\_FEE\_STATE STRING ,

FEE\_TIME STRING ,

FEE\_STAFF\_ID STRING ,

CANCEL\_TAG STRING ,

CANCEL\_DATE STRING ,

CANCEL\_STAFF\_ID STRING ,

CANCEL\_DEPART\_ID STRING ,

CANCEL\_CITY\_CODE STRING ,

CANCEL\_EPARCHY\_CODE STRING ,

CHECK\_TYPE\_CODE STRING ,

CHK\_TAG STRING ,

AUDIT\_TAG STRING ,

AUDIT\_BATCH\_NO STRING ,

ACTOR\_NAME STRING ,

ACTOR\_CERTTYPEID STRING ,

ACTOR\_PHONE STRING ,

ACTOR\_CERTNUM STRING ,

CONTACT STRING ,

CONTACT\_PHONE STRING ,

CONTACT\_ADDRESS STRING ,

REMARK STRING ,

IF\_MAINTENANCE STRING ,

ORDER\_ID STRING ,

SUB\_ORDER\_ID bigint ,

MAIN\_DISCNT\_CODE bigint ,

PRODUCT\_SPEC STRING ,

PROV\_ID STRING ,

UNI\_TRADE\_EPARCHY\_CODE STRING ,

UNI\_EPARCHY\_CODE STRING ,

UNI\_CANCEL\_EPARCHY\_CODE STRING ,

UNI\_BRAND\_CODE STRING ,

UNI\_SERVICE\_TYPE STRING ,

UNI\_SUBSCRIBE\_STATE STRING ,

UNI\_IN\_MODE\_CODE STRING )

row format delimited fields terminated by '\001'

location '/user/root/BONC/TF\_B\_TRADE#CURDATEYYYYMMDD#/'

## 纵表属性参数表TD\_B\_ITEMS

create external table IF NOT EXISTS TD\_B\_ITEMS (

ATTR\_CODE STRING ,

ATTR\_NAME STRING ,

VALUE\_DESC STRING ,

UPDATE\_TIME STRING ,

UPDATE\_STAFF\_ID STRING ,

UPDATE\_DEPART\_ID STRING ,

PROV\_ID STRING

)

row format delimited fields terminated by '\001'

location '/user/root/BONC/TD\_B\_ITEMS#CURDATEYYYYMMDD#/'

## 用户-服务状态TF\_F\_USER\_SVCSTATE

create external table IF NOT EXISTS TF\_F\_USER\_SVCSTATE (

USER\_ID bigint ,

SERVICE\_ID STRING ,

UNI\_MAIN\_TAG STRING ,

STATE\_CODE STRING ,

START\_DATE STRING ,

END\_DATE STRING ,

UPDATE\_TIME STRING ,

PROV\_ID STRING ,

UNI\_STATE\_CODE STRING )

row format delimited fields terminated by '\001'

location '/user/root/BONC/TF\_F\_USER\_SVCSTATE#CURDATEYYYYMMDD#/'

## 终端资源资料表TF\_R\_TERMINAL\_ARCH

create external table IF NOT EXISTS TF\_R\_TERMINAL\_ARCH (

TERMINAL\_ID STRING ,

TERMINAL\_TYPE\_CODE STRING ,

BRAND\_CODE STRING ,

TERMINAL\_MODEL\_CODE STRING ,

PROPERTY\_ID STRING ,

SUPPLY\_COOP\_ID STRING ,

FUND\_COOP\_ID STRING ,

TERMINAL\_STATE STRING ,

PRE\_TERMINAL\_STATE STRING ,

EPARCHY\_CODE STRING ,

CITY\_CODE STRING ,

STOCK\_ID STRING ,

STOCK\_ID\_O STRING ,

STOCK\_ID\_P STRING ,

STOCK\_LEVEL STRING ,

STAFF\_ID STRING ,

AGENT\_ID STRING ,

DEPOT\_CODE STRING ,

ASSIGN\_TAG STRING ,

CONFIRM\_TAG STRING ,

ASSIGN\_TIME STRING ,

ASSIGN\_STAFF\_ID STRING ,

TERMINAL\_SRC\_CODE STRING ,

STOCK\_ID\_IN STRING ,

RECORD\_DATE STRING ,

TIME\_IN STRING ,

STAFF\_ID\_IN STRING ,

BATCH\_IN\_STOCK STRING ,

PURCHASE\_PRICE bigint ,

INSTOCK\_PRICE bigint ,

PRE\_ACPT\_ORG STRING ,

SALE\_LOG\_ID STRING ,

SALE\_TAG STRING ,

SALE\_FEE bigint ,

SALE\_TIME STRING ,

SALE\_STAFF\_ID STRING ,

SALE\_DEPART\_ID STRING ,

AGENT\_SALE\_FLAG STRING ,

AGENT\_PURCHASE\_FLAG STRING ,

SALE\_POLICY STRING ,

SALE\_RETURN\_TYPE STRING ,

SALE\_RETURN\_DATE STRING ,

VALID\_DATE STRING ,

CONTRACT\_ID STRING ,

ORDER\_ID STRING ,

APPLY\_ID STRING ,

USE\_TYPE STRING ,

USR\_NUMBER STRING ,

USR\_ID STRING ,

COPARTNER\_NO STRING ,

PROTECT\_TAG STRING ,

TERMINAL\_PRICE bigint ,

TERMINAL\_COST bigint ,

PACKAGE\_FLAG STRING ,

PACKAGE\_CODE STRING ,

PACKAGE\_TIME STRING ,

BOX\_CODE STRING ,

BOX\_TIME STRING ,

SETTLE\_PRICE bigint ,

REMARK STRING ,

RSRV\_TAG1 STRING ,

RSRV\_TAG2 STRING ,

RSRV\_TAG3 STRING ,

RSRV\_TAG4 STRING ,

RSRV\_NUMBER1 bigint ,

RSRV\_NUMBER2 bigint ,

RSRV\_NUMBER3 bigint ,

RSRV\_TIME1 STRING ,

RSRV\_TIME2 STRING ,

RSRV\_TIME3 STRING ,

RSRV\_STR1 STRING ,

RSRV\_STR2 STRING ,

RSRV\_STR3 STRING ,

RSRV\_STR4 STRING ,

RSRV\_STR5 STRING ,

PROV\_ID STRING ,

UNI\_BRAND\_CODE STRING ,

UNI\_TERMINAL\_MODEL\_CODE STRING ,

UNI\_EPARCHY\_CODE STRING )

row format delimited fields terminated by '\001'

location '/user/root/BONC/TF\_R\_TERMINAL\_ARCH#CURDATEYYYYMMDD#/'

## 渠道基本资料TF\_CHL\_CHANNEL

create external table IF NOT EXISTS TF\_CHL\_CHANNEL (

PROV\_ID STRING ,

CHNL\_ID STRING ,

CHNL\_CODE STRING ,

CHNL\_NAME STRING ,

CHNL\_TYPE\_ID STRING ,

CHNL\_KIND\_ID STRING ,

CHNL\_KIND\_FRAME STRING ,

CHNL\_LEVEL STRING ,

CHNL\_LAYER STRING ,

CHNL\_CREDIT STRING ,

CHNL\_DESC STRING ,

EPARCHY\_CODE STRING ,

CITY\_CODE STRING ,

AREA\_CODE STRING ,

SUPER\_CHNL\_ID STRING ,

AREA\_TYPE STRING ,

OPER\_TYPE STRING ,

IN\_MODE STRING ,

USER\_CHNL\_CODE STRING ,

ERP\_CODE STRING ,

ERP\_AREA\_CODE STRING ,

SOURCE\_TYPE STRING ,

SOURCE\_ID STRING ,

COPARTNER\_ID STRING ,

STATE STRING ,

PARENT\_DEPART\_ID STRING ,

MANAGE\_CHNL\_ID STRING ,

IF\_MANAGE\_CHNL STRING ,

ENTITY\_CODE STRING ,

CHNL\_APPLY\_ID bigint ,

CANCEL\_TAG STRING ,

REMARK STRING ,

AUDIT\_DATE STRING ,

AUDIT\_NOTE STRING ,

AUDIT\_STAFF\_ID STRING ,

AUDIT\_STATE STRING ,

CREATE\_DATE STRING ,

CREATE\_STAFF\_ID STRING ,

REMOVE\_DATE STRING ,

REMOVE\_STAFF\_ID STRING ,

REMOVE\_CHANGE STRING ,

LOCK\_DATE STRING ,

LOCK\_STAFF\_ID STRING ,

LOCK\_CHANGE STRING ,

UNLOCK\_DATE STRING ,

UNLOCK\_STAFF\_ID STRING ,

UNLOCK\_CHANGE STRING ,

REUSE\_DATE STRING ,

REUSE\_STAFF\_ID STRING ,

REUSE\_CHANGE STRING ,

UPDATE\_DATE STRING ,

UPDATE\_STAFF\_ID STRING ,

UPDATE\_TRADE\_ID STRING ,

RSRV\_NUM1 bigint ,

RSRV\_NUM2 bigint ,

RSRV\_NUM3 bigint ,

RSRV\_STR1 STRING ,

RSRV\_STR2 STRING ,

RSRV\_STR3 STRING ,

RSRV\_STR4 STRING ,

RSRV\_STR6 STRING ,

RSRV\_STR7 STRING ,

RSRV\_STR8 STRING ,

RSRV\_STR9 STRING ,

RSRV\_STR10 STRING ,

RSRV\_DATE1 STRING ,

RSRV\_DATE2 STRING ,

RSRV\_DATE3 STRING ,

RSRV\_TAG1 STRING ,

RSRV\_TAG2 STRING ,

RSRV\_TAG3 STRING ,

CHNL\_MANAGER STRING ,

FILE\_LIST STRING ,

REMOVE\_TAG STRING ,

IS\_RWD\_CNT STRING ,

DEPART\_CODE STRING ,

MANANGE\_DEPART\_ID STRING ,

NET\_FORM STRING ,

SCALE\_TYPE STRING ,

KPI\_LEVEL STRING ,

RSRV\_STR5 STRING ,

COVER\_AREA\_RATE STRING ,

CHNL\_FUN STRING ,

DEPART\_ID STRING ,

COMPANY\_ID STRING ,

UNI\_CHNL\_KIND\_ID STRING ,

UNI\_CHNL\_LEVEL STRING ,

UNI\_AREA\_TYPE STRING ,

UNI\_EPARCHY\_CODE STRING )

row format delimited fields terminated by '\001'

location '/user/root/BONC/TF\_CHL\_CHANNEL#CURDATEYYYYMMDD#/'

## 用户主表TF\_F\_USER

create external table IF NOT EXISTS TF\_F\_USER

(

USER\_ID bigint ,

CUST\_ID bigint ,

USECUST\_ID bigint ,

BRAND\_CODE STRING ,

PRODUCT\_ID STRING ,

PRODUCT\_TYPE\_CODE STRING ,

EPARCHY\_CODE STRING ,

CITY\_CODE STRING ,

IN\_NET\_MODE STRING ,

USER\_PASSWD STRING ,

USER\_DIFF\_CODE STRING ,

USER\_TYPE\_CODE STRING ,

SERIAL\_NUMBER STRING ,

NET\_TYPE\_CODE STRING ,

SCORE\_VALUE bigint ,

CREDIT\_CLASS STRING ,

BASIC\_CREDIT\_VALUE bigint ,

CREDIT\_VALUE bigint ,

ACCT\_TAG STRING ,

PREPAY\_TAG STRING ,

IN\_DATE STRING ,

OPEN\_DATE STRING ,

OPEN\_MODE STRING ,

OPEN\_DEPART\_ID STRING ,

OPEN\_STAFF\_ID STRING ,

IN\_DEPART\_ID STRING ,

IN\_STAFF\_ID STRING ,

REMOVE\_TAG STRING ,

DESTROY\_TIME STRING ,

REMOVE\_EPARCHY\_CODE STRING ,

REMOVE\_CITY\_CODE STRING ,

REMOVE\_DEPART\_ID STRING ,

REMOVE\_REASON\_CODE STRING ,

PRE\_DESTROY\_TIME STRING ,

FIRST\_CALL\_TIME STRING ,

LAST\_STOP\_TIME STRING ,

USER\_STATE\_CODESET STRING ,

MPUTE\_MONTH\_FEE STRING ,

MPUTE\_DATE STRING ,

UPDATE\_TIME STRING ,

ASSURE\_CUST\_ID bigint ,

ASSURE\_TYPE\_CODE STRING ,

ASSURE\_DATE STRING ,

DEVELOP\_STAFF\_ID STRING ,

DEVELOP\_DATE STRING ,

DEVELOP\_EPARCHY\_CODE STRING ,

DEVELOP\_CITY\_CODE STRING ,

DEVELOP\_DEPART\_ID STRING ,

DEVELOP\_NO STRING ,

REMARK STRING ,

CREDIT\_RULE\_ID bigint ,

CONTRACT\_ID STRING ,

CHANGEUSER\_DATE STRING ,

MAIN\_DISCNT\_CODE bigint ,

PRODUCT\_SPEC STRING ,

ACCOUNT\_ID STRING ,

GRID\_ID STRING ,

PROV\_ID STRING ,

UNI\_BRAND\_CODE STRING ,

UNI\_EPARCHY\_CODE STRING ,

UNI\_USER\_TYPE\_CODE STRING ,

UNI\_PREPAY\_TAG STRING ,

service\_type STRING ,

UNI\_USER\_STATE\_CODESET STRING ,

UNI\_ASSURE\_TYPE\_CODE STRING ,

UNI\_OPEN\_MODE STRING ,

UNI\_REMOVE\_TAG STRING )

row format delimited fields terminated by '\001'

location '/user/root/BONC/TF\_F\_USER#CURDATEYYYYMMDD#/'

G网语音清单表TG\_CDRMM

create external table IF NOT EXISTS TG\_CDRMM

(

PROV\_ID STRING ,

SOURCE\_TYPE STRING ,

BIZ\_TYPE STRING ,

FID STRING ,

RR\_FLAG STRING ,

DEAL\_TIME STRING ,

START\_DATE STRING ,

START\_TIME STRING ,

END\_DATE STRING ,

END\_TIME STRING ,

CALL\_TYPE STRING ,

USER\_ID bigint ,

IMSI\_NUMBER STRING ,

MSISDN STRING ,

OTHER\_PARTY STRING ,

MSRN STRING ,

MSC STRING ,

LAC1 STRING ,

CELL\_ID1 STRING ,

OTHER\_LAC1 STRING ,

OTHER\_CELL\_ID1 STRING ,

LAC2 STRING ,

CELL\_ID2 STRING ,

OTHER\_LAC2 STRING ,

OTHER\_CELL\_ID2 STRING ,

TRUNK\_GROUPOUT STRING ,

TRUNK\_GROUPIN STRING ,

SERVICE\_TYPE STRING ,

SERVICE\_CODE STRING ,

IMEI STRING ,

NET\_TYPE STRING ,

FCI STRING ,

MNS\_TYPE STRING ,

ISMULTIMEDIA STRING ,

HOME\_AREA\_CODE STRING ,

VISIT\_AREA\_CODE STRING ,

B\_MSISDN STRING ,

CALLED\_HOME\_CODE STRING ,

CALLED\_CODE STRING ,

A\_ASP STRING ,

A\_SERV\_TYPE STRING ,

B\_ASP STRING ,

B\_SERV\_TYPE STRING ,

B\_HOME\_TYPE STRING ,

CALLED\_TYPE STRING ,

C\_SERV\_TYPE STRING ,

CARRIER\_TYPE STRING ,

TRANS\_TYPE STRING ,

LONG\_TYPE1 STRING ,

LONG\_GROUP1 STRING ,

LONG\_TYPE2 STRING ,

LONG\_GROUP2 STRING ,

ROAM\_TYPE STRING ,

FILE\_NO STRING ,

STOP\_CAUSE STRING ,

ORIGIN\_NO STRING ,

FORWARDCAUSE STRING ,

EDGE\_FLAG STRING ,

OFFICE\_CODE STRING ,

RATE\_IND STRING ,

VPN\_FLAG STRING ,

IN\_CELL STRING ,

CHANNEL\_NO bigint ,

A\_PRODUCT\_ID STRING ,

CUST\_ID bigint ,

VPN\_ID STRING ,

OTHER\_VPN\_ID STRING ,

SVPN\_ID STRING ,

OTHER\_SVPN\_ID STRING ,

USER\_TYPE STRING ,

DOUBLEMODE STRING ,

OPEN\_DATETIME STRING ,

A\_USER\_STAT STRING ,

B\_BRANDTYPE STRING ,

B\_PRODUCT\_ID bigint ,

B\_OFFICE\_CODE STRING ,

ISNONEED\_TP STRING ,

SPECIAL\_PHONE\_GROUP STRING ,

FREE\_CODE STRING ,

BILL\_ITEM STRING ,

ERROR\_CODE STRING ,

CALL\_DURATION bigint ,

TIMES1 bigint ,

TIMES2 bigint ,

CFEE bigint ,

LFEE bigint ,

OTHERFEE bigint ,

DISCOUNT\_CFEE bigint ,

DISCOUNT\_LFEE bigint ,

DISCOUNT\_OTHERFEE bigint ,

ORG\_CFEE bigint ,

ORG\_LFEE bigint ,

ORG\_OTHERFEE bigint ,

TARIFF\_FEE bigint ,

RATE\_TIMES bigint ,

INDB\_TIME STRING ,

RESERVER1 STRING ,

RESERVER2 STRING ,

RESERVER3 STRING ,

RESERVER4 STRING ,

RESERVER5 STRING ,

RESERVER6 STRING ,

RESERVER7 STRING ,

RESERVER8 STRING ,

PARTITION\_ID STRING ,

RN\_INFO STRING ,

RESERVER9 STRING ,

RESERVER10 STRING ,

RESERVER11 STRING ,

RESERVER12 STRING ,

RESERVER13 STRING ,

RESERVER14 STRING ,

RESERVER15 STRING ,

UNI\_CALL\_TYPE STRING ,

UNI\_SERVICE\_CODE STRING ,

UNI\_CALLED\_HOME\_CODE STRING ,

UNI\_CALLED\_CODE STRING ,

UNI\_B\_ASP STRING ,

UNI\_B\_SERV\_TYPE STRING ,

UNI\_LONG\_TYPE STRING ,

UNI\_HOME\_AREA\_CODE STRING ,

UNI\_VISIT\_AREA\_CODE STRING ,

UNI\_SERVICE\_TYPE STRING,

UNI\_CARRIER\_TYPE STRING,

UNI\_ISMULTIMEDIA STRING,

UNI\_MNS\_TYPE STRING,

UNI\_TOLL\_CONNECTING\_TYPE STRING,

UNI\_ROAM\_TYPE STRING,

UNI\_RR\_FLAG STRING,

UNI\_SOURCE\_TYPE STRING

)

row format delimited fields terminated by '\001'

location '/user/root/BONC/TG\_CDRMM#CURDATEYYYYMMDD#/'

## SP服务提供清单表TG\_CDRMM\_SP

create external table IF NOT EXISTS TG\_CDRMM\_SP

(

PROV\_ID STRING ,

SOURCE\_TYPE STRING ,

BIZ\_TYPE STRING ,

FID STRING ,

RR\_FLAG STRING ,

FILE\_NAME STRING ,

DEAL\_TIME STRING ,

CDR\_SEQID STRING ,

CDR\_TYPE STRING ,

SP\_CODE STRING ,

SP\_LOGINNAME STRING ,

MSIP\_ADDRESS STRING ,

SP\_SRVTYPE STRING ,

SP\_SRVTYPENAME STRING ,

SP\_SRVKIND STRING ,

SP\_SRVID STRING ,

SP\_SRVNAME STRING ,

CONTENT\_CODE STRING ,

CONTENT\_NAME STRING ,

MSISDN STRING ,

IMSI\_NUMBER STRING ,

NAI\_NAME STRING ,

NAI\_DOMAIN STRING ,

START\_DATE STRING ,

START\_TIME STRING ,

END\_DATE STRING ,

END\_TIME STRING ,

QTY bigint ,

ACTIVE\_TIME bigint ,

UP\_COUNT bigint ,

DOWN\_COUNT bigint ,

TOTAL\_COUNT bigint ,

PRE\_DISCOUNTFEE bigint ,

POST\_DISCOUNTFEE bigint ,

DISCOUNTFEE bigint ,

RATE\_UNIT STRING ,

FEE\_TYPE STRING ,

FEE\_RATEDESCODE STRING ,

DISCOUNT\_DESCODE STRING ,

URL STRING ,

SYS\_ID STRING ,

SYS\_INTRAKEY STRING ,

USER\_TYPE STRING ,

OTHER\_PARTY STRING ,

RATE\_FLAG STRING ,

SEND\_STATUS STRING ,

FILE\_NO STRING ,

ERROR\_CODE STRING ,

NAI\_TYPE STRING ,

HOME\_AREA\_CODE STRING ,

VISIT\_AREA\_CODE STRING ,

ROAM\_TYPE STRING ,

CALLED\_HOME\_CODE STRING ,

CUST\_ID bigint ,

USER\_ID STRING ,

A\_PRODUCT\_ID STRING ,

A\_BRAND\_TYPE STRING ,

A\_SERV\_TYPE STRING ,

A\_USER\_STAT STRING ,

CHANNEL\_NO bigint ,

OFFICE\_CODE STRING ,

DOUBLEMODE STRING ,

OPEN\_DATETIME STRING ,

B\_ASP STRING ,

B\_SERV\_TYPE STRING ,

B\_HOME\_TYPE STRING ,

RATE\_TIMES bigint ,

FREE\_CODE STRING ,

BILL\_ITEM STRING ,

CFEE bigint ,

DISCOUNT\_CFEE bigint ,

INDB\_TIME STRING ,

RESERVER1 STRING ,

RESERVER2 STRING ,

RESERVER3 STRING ,

PARTITION\_ID STRING ,

M\_TIMES bigint ,

T\_TIMES bigint ,

MFEE bigint ,

DISCOUNT\_MFEE bigint ,

TFEE bigint ,

DISCOUNT\_TFEE bigint ,

PRODUCT\_ID STRING ,

PLAN\_ID STRING ,

UNI\_VISIT\_AREA\_CODE STRING,

UNI\_B\_SERV\_TYPE STRING,

UNI\_B\_ASP STRING,

UNI\_CDR\_TYPE STRING,

UNI\_SP\_SRVTYPE STRING,

UNI\_ROAM\_TYPE STRING,

UNI\_SP\_SOURCE\_TYPE STRING

)

row format delimited fields terminated by '\001'

location '/user/root/BONC/TG\_CDRMM\_SP#CURDATEYYYYMMDD#/'

## 用户购机信息表TF\_F\_USER\_PURCHASE

create external table IF NOT EXISTS TF\_F\_USER\_PURCHASE

(

USER\_ID bigint ,

BINDSALE\_ATTR STRING ,

EXTRA\_DEV\_FEE bigint ,

MPFEE bigint ,

FEEITEM\_CODE STRING ,

FOREGIFT bigint ,

FOREGIFT\_CODE STRING ,

FOREGIFT\_BACKMODE STRING ,

AGREEMENT\_MONTHS bigint ,

END\_MODE STRING ,

DEVICE\_TYPE STRING ,

MOBILE\_COST bigint ,

DEVICE\_NAME STRING ,

DEVICE\_BRAND STRING ,

IMEI STRING ,

LIST\_BANK STRING ,

LIST\_FEE bigint ,

LIST\_CODE STRING ,

CREDIT\_ORG STRING ,

CREDIT\_TYPE STRING ,

CREDIT\_CARD\_NUM STRING ,

AGREEMENT STRING ,

PRODUCT\_ID STRING ,

PACKAGE\_ID STRING ,

STAFF\_ID STRING ,

DEPART\_ID STRING ,

START\_DATE STRING ,

END\_DATE STRING ,

REMARK STRING ,

ITEM\_ID STRING ,

PROV\_ID STRING ,

UNI\_FOREGIFT\_CODE STRING )

row format delimited fields terminated by '\001'

location '/user/root/BONC/TF\_F\_USER\_PURCHASE#CURDATEYYYYMMDD#/'

## 用户产品属性子表TF\_F\_USER\_PROD\_ITEM

create external table IF NOT EXISTS TF\_F\_USER\_PROD\_ITEM

(

ITEM\_ID bigint ,

ATTR\_CODE STRING ,

ATTR\_VALUE STRING ,

START\_DATE STRING ,

END\_DATE STRING ,

PROV\_ID STRING

)

row format delimited fields terminated by '\001'

location '/user/root/BONC/TF\_F\_USER\_PROD\_ITEM#CURDATEYYYYMMDD#/'

## 用户产品子表TF\_F\_USER\_PRODUCT

create external table IF NOT EXISTS TF\_F\_USER\_PRODUCT

(

USER\_ID bigint ,

PRODUCT\_MODE STRING ,

PRODUCT\_ID STRING ,

BRAND\_CODE STRING ,

START\_DATE STRING ,

END\_DATE STRING ,

ITEM\_ID STRING ,

USER\_ID\_A STRING ,

PROV\_ID STRING ,

UNI\_PRODUCT\_MODE STRING

)

row format delimited fields terminated by '\001'

location '/user/root/BONC/TF\_F\_USER\_PRODUCT#CURDATEYYYYMMDD#/'

## 用户购机属性子表TF\_F\_USER\_PURCHASE\_ITEM

create external table IF NOT EXISTS TF\_F\_USER\_PURCHASE\_ITEM

(

ITEM\_ID STRING ,

ATTR\_CODE STRING ,

ATTR\_VALUE STRING ,

START\_DATE STRING ,

END\_DATE STRING ,

PROV\_ID STRING

)

row format delimited fields terminated by '\001'

location '/user/root/BONC/TF\_F\_USER\_PURCHASE\_ITEM#CURDATEYYYYMMDD#/'

## 帐单表TS\_B\_BILL

create external table IF NOT EXISTS TS\_B\_BILL

(

EPARCHY\_CODE STRING ,

CITY\_CODE STRING ,

NET\_TYPE\_CODE STRING ,

SERIAL\_NUMBER STRING ,

BILL\_ID STRING ,

ACCT\_ID bigint ,

USER\_ID bigint ,

CYCLE\_ID bigint ,

INTEGRATE\_ITEM\_CODE STRING ,

FEE STRING ,

BALANCE STRING ,

PRINT\_FEE bigint ,

B\_DISCNT STRING ,

A\_DISCNT STRING ,

ADJUST\_BEFORE STRING ,

ADJUST\_AFTER STRING ,

LATE\_FEE STRING ,

LATE\_BALANCE STRING ,

LATECAL\_DATE STRING ,

CANPAY\_TAG STRING ,

PAY\_TAG STRING ,

BILL\_PAY\_TAG STRING ,

DESC\_OWE\_TAG STRING ,

VERSION\_NO bigint ,

UPDATE\_TIME STRING ,

UPDATE\_DEPART\_ID STRING ,

UPDATE\_STAFF\_ID STRING ,

CHARGE\_ID STRING ,

WRITEOFF\_FEE1 bigint ,

WRITEOFF\_FEE2 bigint ,

WRITEOFF\_FEE3 bigint ,

BACKUP\_INFO STRING ,

ROLL\_BACK\_INFO STRING ,

RSRV\_FEE1 bigint ,

RSRV\_FEE2 bigint ,

RSRV\_FEE3 bigint ,

RSRV\_INFO1 STRING ,

RSRV\_INFO2 STRING ,

ZK\_FEE bigint ,

PROV\_ID STRING ,

UNI\_EPARCHY\_CODE STRING ,

UNI\_CITY\_CODE STRING ,

UNI\_BILL\_PAY\_TAG STRING ,

BALANCE\_BACKUP\_INFO1 bigint ,

BALANCE\_BACKUP\_INFO2 bigint ,

CREATE\_CYCLE\_ID bigint

)

row format delimited fields terminated by '\001'

location '/user/root/BONC/TS\_B\_BILL#CURDATEYYYYMMDD#/'

## O域流量详单表GN\_URL

create external table IF NOT EXISTS GN\_URL(

MSISDN string,

LAC string,

CI\_NUM string,

IMEI string,

GPRS\_TYPE string,

START\_TIME string,

END\_TIME string,

CONTINUE\_TIME string,

UP\_COUNT bigint,

DOWN\_COUNT bigint,

TOTAL\_COUNT bigint,

NET\_TYPE string,

TERMINAL\_IP string,

VISIT\_IP string,

STATUS string,

USER\_AGENT string,

APN string,

IMSI string,

SGSN\_IP string,

GGSN\_IP string,

CONTENT\_TYPE string,

SOURCE\_PORT string,

Destination\_port string,

RECORDSEQ\_ID string,

Merge\_Records string,

URL string,

REFERER\_INFO string

)

row format delimited fields terminated by '\001'

location '/user/root/BONC/gnfiles#CURDATEYYYYMMDD#/'

3：数据加工过程以及加工脚本。

     具体描述：如何将ODS源表结构加工成DWD、DWA数据表结构（具体表结构见验证方案附件），请提供相应的脚本或者工具；*[pingxiaogang]* --DWD\DWA数据加工脚本

# DWD加工脚本

## 业务订购实例（月）DWD\_M\_PRD\_USER\_INFO

### 建表语句

create table IF NOT EXISTS DWD\_M\_PRD\_USER\_INFO(MONTH\_ID string,

PROV\_ID string,

USER\_ID string,

CUST\_ID string,

ACCOUNT\_ID string,

DEVICE\_NUMBER string,

SERVICE\_TYPE string,

SERVICE\_TYPE\_PROV string,

PAY\_MODE string,

PAY\_MODE\_PROV string,

USER\_TYPE string,

USER\_TYPE\_PROV string,

PRODUCT\_ID string,

BRAND\_ID string,

INNET\_METHOD string,

INNET\_AREA\_ID string,

INNET\_CITY\_ID string,

AREA\_ID string,

AREA\_ID\_PROV string,

CITY\_ID string,

CITY\_ID\_PROV string,

TOWN\_ID string,

TOWN\_ID\_PROV string,

BLOCK\_ID string,

USER\_STATUS string,

USER\_STATUS\_PROV string,

UPDATE\_DATE timestamp,

OPER\_NO string,

CHANNEL\_ID string,

OPER\_CHANNEL\_ID string,

OPER\_DATE timestamp,

CREATE\_DATE timestamp,

INNET\_DATE timestamp,

CLOSE\_DATE timestamp,

START\_DATE timestamp,

END\_DATE timestamp,

LAST\_STOP\_DATE timestamp,

REMOVE\_TAG string,

IS\_INNET string,

IS\_THIS\_DEV string,

IS\_THIS\_BREAK string,

IS\_THIS\_ACCT string,

CHANGE\_STATUS string,

CHANGE\_EFF\_DATE timestamp

) partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

insert overwrite table DWD\_M\_PRD\_USER\_INFO PARTITION (month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

select

'#CURDATEYYYYMM#' MONTH\_ID,

'#PROVINCE#' PROV\_ID,

A.USER\_ID USER\_ID,

A.CUST\_ID CUST\_ID,

A.ACCOUNT\_ID ACCOUNT\_ID,

A.SERIAL\_NUMBER DEVICE\_NUMBER,

A.service\_type SERVICE\_TYPE,

A.NET\_TYPE\_CODE SERVICE\_TYPE\_PROV,

A.UNI\_PREPAY\_TAG PAY\_MODE,

A.PREPAY\_TAG PAY\_MODE\_PROV,

A.USER\_TYPE\_CODE USER\_TYPE,

A.USER\_TYPE\_CODE USER\_TYPE\_PROV,

A.PRODUCT\_ID PRODUCT\_ID,

A.BRAND\_CODE BRAND\_ID,

A.IN\_NET\_MODE INNET\_METHOD,

A.UNI\_EPARCHY\_CODE INNET\_AREA\_ID,

A.CITY\_CODE INNET\_CITY\_ID,

A.EPARCHY\_CODE AREA\_ID,

A.EPARCHY\_CODE AREA\_ID\_PROV,

A.CITY\_CODE CITY\_ID,

A.CITY\_CODE CITY\_ID\_PROV,

-1 TOWN\_ID,

-1 TOWN\_ID\_PROV,

-1 BLOCK\_ID,

A.USER\_STATE\_CODESET USER\_STATUS,

A.USER\_STATE\_CODESET USER\_STATUS\_PROV,

A.UPDATE\_TIME UPDATE\_DATE,

-1 OPER\_NO,

A.IN\_DEPART\_ID CHANNEL\_ID,

A.IN\_DEPART\_ID OPER\_CHANNEL\_ID,

A.IN\_DATE OPER\_DATE,

A.IN\_DATE CREATE\_DATE,

A.OPEN\_DATE INNET\_DATE,

A.DESTROY\_TIME CLOSE\_DATE,

-1 START\_DATE,

-1 END\_DATE,

A.LAST\_STOP\_TIME LAST\_STOP\_DATE,

A.UNI\_REMOVE\_TAG REMOVE\_TAG,

CASE WHEN A.UNI\_REMOVE\_TAG = '0'

AND (SUBSTR(A.OPEN\_DATE,1,6) <= '#CURDATEYYYYMM#' OR A.OPEN\_DATE IS NULL)

THEN 1 ELSE 0 END IS\_INNET,

case when substr(OPEN\_DATE,1,6)='#CURDATEYYYYMM#' then '1' else '0' end IS\_THIS\_DEV,

case when substr(DESTROY\_TIME,1,6)='#CURDATEYYYYMM#' then '1' else '0' end IS\_THIS\_BREAK,

case when (b.FEE is NOT null) and b.fee>0 then '1' else '0' end IS\_THIS\_ACCT,

-1 CHANGE\_STATUS,

-1 CHANGE\_EFF\_DATE

from TF\_F\_USER a

left outer join (select USER\_ID,sum(FEE) FEE from TS\_B\_BILL B group by USER\_ID) b

on (a.user\_id=b.user\_id)

## 产品订购信息（月）DWD\_M\_PRD\_AL\_USER\_PROD

### 建表语句

create table IF NOT EXISTS DWD\_M\_PRD\_AL\_USER\_PROD(MONTH\_ID string,

PROV\_ID string,

ITEM\_ID string,

USER\_ID string,

PRODUCT\_ID string,

IF\_VALID string,

CREATE\_DATE timestamp,

START\_DATE timestamp,

END\_DATE timestamp,

IF\_COMP string,

OPER\_DATE string,

OPER\_NO string,

OPER\_CHANNEL\_NO string

) partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

INSERT overwrite table DWD\_M\_PRD\_AL\_USER\_PROD PARTITION (month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

select '#CURDATEYYYYMM#' MONTH\_ID,

'#PROVINCE#' PROV\_ID,

-1 ITEM\_ID,

USER\_ID USER\_ID,

PRODUCT\_ID PRODUCT\_ID,

case when substr(START\_DATE,1,6)<='#CURDATEYYYYMM#' and substr(END\_DATE,1,6)>'#CURDATEYYYYMM#' then '1' else '0' end IF\_VALID,

START\_DATE CREATE\_DATE,

START\_DATE START\_DATE,

END\_DATE END\_DATE,

-1 IF\_COMP,

START\_DATE OPER\_DATE,

-1 OPER\_NO,

-1 OPER\_CHANNEL\_NO

FROM TF\_F\_USER\_PRODUCT

## 移动业务活动订购信息（月）DWD\_M\_PRD\_AL\_USER\_ACTIVE

### 建表语句

create table IF NOT EXISTS DWD\_M\_PRD\_AL\_USER\_ACTIVE(MONTH\_ID string,

PROV\_ID string,

ACTIVITY\_SUBS\_ID string,

USER\_ID string,

ACTIVE\_ID string,

ACTIVE\_ID\_PROV string,

SERVICE\_TYPE string,

SERVICE\_TYPE\_PROV string,

DEVICE\_NUMBER string,

PROD\_ID string,

ACTIVE\_TYPE string,

ACTIVE\_TYPE\_PROV string,

AGREEMENT\_MONTHS string,

MINIMUM\_CONSUME string,

RES\_TYPE string,

RES\_AREA\_ID string,

RES\_IMEI string,

RES\_BRAND string,

RES\_BRAND\_PROV string,

RES\_NUM INT,

RES\_MODEL string,

RES\_MODE\_PROV string,

RES\_COST INT,

RES\_PRICE INT,

ALLOWANCE\_FEE INT,

DEPOSIT\_FEE INT,

STORED\_FEE INT,

PRESENT\_FEE INT,

RETURN\_RATIO INT,

SPENDING\_SUM INT,

OPER\_DATE string,

START\_DATE string,

END\_DATE string

) partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

insert overwrite table DWD\_M\_PRD\_AL\_USER\_ACTIVE PARTITION(month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

select

'#CURDATEYYYYMM#' MONTH\_ID,--canshu

'#PROVINCE#' PROV\_ID,--canshu

-1 ACTIVITY\_SUBS\_ID,

A.USER\_ID USER\_ID,

-1 ACTIVE\_ID,

A.BINDSALE\_ATTR ACTIVE\_ID\_PROV,

B.SERVICE\_TYPE SERVICE\_TYPE,

B.NET\_TYPE\_CODE SERVICE\_TYPE\_PROV,

B.SERIAL\_NUMBER DEVICE\_NUMBER,

B.PRODUCT\_ID PROD\_ID,

-1 ACTIVE\_TYPE,

-1 ACTIVE\_TYPE\_PROV,

A.AGREEMENT\_MONTHS AGREEMENT\_MONTHS,

-1 MINIMUM\_CONSUME,

-1 RES\_TYPE,

-1 RES\_AREA\_ID,

A.IMEI RES\_IMEI,

-1 RES\_BRAND,

A.DEVICE\_BRAND RES\_BRAND\_PROV,

-1 RES\_NUM,

-1 RES\_MODEL,

A.DEVICE\_TYPE RES\_MODE\_PROV,

A.MOBILE\_COST RES\_COST,

A.MPFEE RES\_PRICE,

A.EXTRA\_DEV\_FEE ALLOWANCE\_FEE,

A.FOREGIFT DEPOSIT\_FEE,

-1 STORED\_FEE,

-1 PRESENT\_FEE,

-1 RETURN\_RATIO,

-1 SPENDING\_SUM,

A.START\_DATE OPER\_DATE,

A.START\_DATE START\_DATE,

A.END\_DATE END\_DATE

from TF\_F\_USER\_PURCHASE A

LEFT outer JOIN TF\_F\_USER B

ON A.USER\_ID=B.USER\_ID

## 明细帐单(分摊前)（月）DWD\_M\_ACC\_AL\_DIVIDE\_CHARGE

### 建表语句

create table IF NOT EXISTS DWD\_M\_ACC\_AL\_DIVIDE\_CHARGE(MONTH\_ID string,

PROV\_ID string,

USER\_ID string,

FEE\_MONTH string,

ACCT\_ITEM\_ID string,

ACCT\_ITEM\_ID\_PROV string,

SERVICE\_TYPE string,

BEF\_FEE int,

FEE int,

ACCT\_ID string,

SERVICE\_TYPE\_PROV string

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

INSERT OVERWRITE TABLE DWD\_M\_ACC\_AL\_DIVIDE\_CHARGE PARTITION (month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

SELECT '#CURDATEYYYYMM#',

'#PROVINCE#' PROV\_ID,

USER\_ID,

CREATE\_CYCLE\_ID,

INTEGRATE\_ITEM\_CODE,

INTEGRATE\_ITEM\_CODE,

-1,

FEE+A\_DISCNT+ADJUST\_BEFORE,

FEE,

ACCT\_ID,

-1

FROM TS\_B\_BILL

## 移动业务语音详单（月）DWD\_D\_USE\_MB\_VOICE

### 建表语句

create table IF NOT EXISTS DWD\_D\_USE\_MB\_VOICE(MONTH\_ID string,

PROV\_ID string,

TICKET\_ID string,

ORIGINAL\_FILE string,

ACQUISITION\_TIME string,

INSERT\_TIME string,

SUBS\_INSTANCE\_ID string,

DEVICE\_NUMBER string,

SERVICE\_TYPE string,

SERVICE\_TYPE\_PROV string,

HOME\_AREA\_CODE string,

VISIT\_AREA\_CODE string,

TRUNK\_GROUPOUT string,

TRUNK\_GROUPIN string,

MSC string,

LAC string,

CELL\_ID string,

CELL\_TYPE string,

IMSI string,

IMEI string,

PRODUCT\_ID string,

TARIFF\_PLANS\_ID string,

OPPOSE\_NUMBER string,

OPPOSE\_PARTY string,

OPPOSE\_HOME\_CODE string,

OPPOSE\_VISIT\_CODE string,

OPPOSE\_NET\_TYPE string,

OPPOSE\_OPERATOR string,

OPPOSE\_SERVICE\_TYPE string,

OPPOSE\_LAC string,

OPPOSE\_CELL\_ID string,

THIRD\_NUMBER string,

START\_DATE string,

START\_TIME string,

END\_DATE string,

END\_TIME string,

RATE\_TYPE string,

CALL\_TYPE string,

LONG\_TYPE string,

ROAM\_TYPE string,

LAND\_TYPE string,

NET\_TYPE string,

TICKET\_TYPE string,

MSRN string,

FREE\_FLAG string,

VPN\_FLAG string,

VPN\_TYPE string,

VPN\_NO string,

SP\_ID string,

SP\_COMPANY\_NO string,

SP\_PRODUCT\_NO string,

CALL\_DURATION INT,

BASE\_TIMES INT,

DISCOUNT\_TIMES INT,

LONG\_TIMES INT,

BEF\_BASE\_FEE INT,

BEF\_LONG\_FEE INT,

BEF\_INFO\_FEE INT,

BEF\_OTHER\_FEE INT,

BASE\_FEE INT,

LONG\_FEE INT,

INFO\_FEE INT,

OTHER\_FEE INT,

BASE\_DISCOUNT INT,

LONG\_DISCOUNT INT,

INFO\_DISCOUNT INT,

OTHER\_DISCOUNT INT,

IS\_FREE string

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

INSERT OVERWRITE TABLE DWD\_D\_USE\_MB\_VOICE PARTITION(month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

Select '#CURDATEYYYYMM#',

'#PROVINCE#' PROV\_ID,

-1,

FILE\_NO,

-1,

INDB\_TIME,

USER\_ID,

MSISDN,

SERVICE\_TYPE,

SERVICE\_TYPE,

UNI\_HOME\_AREA\_CODE,

UNI\_VISIT\_AREA\_CODE,

TRUNK\_GROUPOUT,

TRUNK\_GROUPIN,

MSC,

LAC1,

CELL\_ID1,

-1,

IMSI\_NUMBER,

IMEI,

A\_PRODUCT\_ID,

-1,

OTHER\_PARTY,

B\_MSISDN,

UNI\_CALLED\_HOME\_CODE,

UNI\_CALLED\_CODE,

-1,

-1,

-1,

OTHER\_LAC1,

OTHER\_CELL\_ID1,

-1,

START\_DATE,

START\_TIME,

END\_DATE,

END\_TIME,

-1,

UNI\_CALL\_TYPE,

UNI\_LONG\_TYPE,

ROAM\_TYPE,

-1,

NET\_TYPE,

ISMULTIMEDIA,

MSRN,

-1,

VPN\_FLAG,

-1,

VPN\_ID,

-1,

-1,

-1,

CALL\_DURATION,

TIMES1,

-1,

TIMES2,

ORG\_CFEE/1000,

ORG\_LFEE/1000,

-1,

ORG\_OTHERFEE/1000,

DISCOUNT\_CFEE/1000,

DISCOUNT\_LFEE/1000,

-1,

DISCOUNT\_OTHERFEE/1000,

(ORG\_CFEE-DISCOUNT\_CFEE)/1000,

(ORG\_LFEE-DISCOUNT\_LFEE)/1000,

-1,

(ORG\_OTHERFEE-DISCOUNT\_OTHERFEE)/1000,

-1

FROM TG\_CDRMM

## 移动业务短信详单（月）DWD\_D\_USE\_MB\_SMS

### 建表语句

create table IF NOT EXISTS DWD\_D\_USE\_MB\_SMS(MONTH\_ID string,

PROV\_ID string,

TICKET\_ID string,

ORIGINAL\_FILE string,

ACQUISITION\_TIME string,

INSERT\_TIME string,

SUBS\_INSTANCE\_ID string,

SERVICE\_TYPE string,

SERVICE\_TYPE\_PROV string,

DEVICE\_NUMBER string,

PRODUCT\_ID string,

IMSI string,

IMEI string,

OPPOSE\_NUMBER string,

OPPOSE\_PARTY string,

OPPOSE\_OPERATOR string,

OPPOSE\_HOME\_CODE string,

OPPOSE\_VISIT\_CODE string,

OPPOSE\_NET\_TYPE string,

OPPOSE\_SERVICE\_TYPE string,

OPPOSE\_LAC string,

OPPOSE\_CELL\_ID string,

SMS\_DIRECT string,

SMS\_TYPE string,

NET\_TYPE string,

ROAM\_TYPE string,

HOME\_AREA\_CODE string,

VISIT\_AREA\_CODE string,

START\_DATE string,

START\_TIME string,

SMS\_ISMG\_CODE string,

RATE\_TYPE string,

MESSAGE\_BYTES string,

STATUS string,

SP\_NO string,

SP\_COMPANY\_NO string,

SP\_SRV\_CODE string,

SP\_PRODUCT\_ID string,

SP\_ACCESS\_NO string,

CONTENT\_CODE string,

SMG\_CODE string,

VPN\_FLAG string,

VPN\_TYPE string,

BEF\_COMM\_FEE INT,

BEF\_INFO\_FEE INT,

BEF\_RENT\_FEE INT,

BEF\_OTHER\_FEE INT,

COMM\_FEE INT,

INFO\_FEE INT,

RENT\_FEE INT,

OTHER\_FEE INT,

COMM\_DISCOUNT INT,

INFO\_DISCOUNT INT,

RENT\_DISCOUNT INT,

OTHER\_DISCOUNT INT,

GIFT\_FEE INT,

TARIFF\_PLANS\_ID string,

COMPENSATION\_AMOUNT INT,

SMS\_FORWARD\_ISMG string,

SMS\_CENTER\_CODE string,

IS\_FREE string,

FREE\_FLAG string

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

insert overwrite table DWD\_D\_USE\_MB\_SMS PARTITION

(month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

select '#CURDATEYYYYMM#' MONTH\_ID,

'#PROVINCE#' PROV\_ID,

A.CDR\_TYPE TICKET\_ID,

A.FILE\_NO ORIGINAL\_FILE,

-1 ACQUISITION\_TIME,

A.INDB\_TIME INSERT\_TIME,

A.USER\_ID SUBS\_INSTANCE\_ID,

-1 SERVICE\_TYPE,

A.BIZ\_TYPE SERVICE\_TYPE\_PROV,

A.MSISDN DEVICE\_NUMBER,

A.PRODUCT\_ID PRODUCT\_ID,

A.IMSI\_NUMBER IMSI,

-1 IMEI,

A.OTHER\_PARTY OPPOSE\_NUMBER,

-1 OPPOSE\_PARTY,

-1 OPPOSE\_OPERATOR,

A.CALLED\_HOME\_CODE OPPOSE\_HOME\_CODE,

-1 OPPOSE\_VISIT\_CODE,

-1 OPPOSE\_NET\_TYPE,

-1 OPPOSE\_SERVICE\_TYPE,

-1 OPPOSE\_LAC,

-1 OPPOSE\_CELL\_ID,

-1 SMS\_DIRECT,

A.CDR\_TYPE SMS\_TYPE,

-1 NET\_TYPE,

A.ROAM\_TYPE ROAM\_TYPE,

A.HOME\_AREA\_CODE HOME\_AREA\_CODE,

A.VISIT\_AREA\_CODE VISIT\_AREA\_CODE,

A.START\_DATE START\_DATE,

A.START\_TIME START\_TIME,

-1 SMS\_ISMG\_CODE,

A.FEE\_TYPE RATE\_TYPE,

-1 MESSAGE\_BYTES,

A.SEND\_STATUS STATUS,

A.SP\_CODE SP\_NO,

A.SP\_CODE SP\_COMPANY\_NO,

A.SP\_SRVID SP\_SRV\_CODE,

-1 SP\_PRODUCT\_ID,

-1 SP\_ACCESS\_NO,

A.CONTENT\_CODE/1000 CONTENT\_CODE,

-1 SMG\_CODE,

-1 VPN\_FLAG,

-1 VPN\_TYPE,

A.PRE\_DISCOUNTFEE/1000 BEF\_COMM\_FEE,

0 BEF\_INFO\_FEE,

0 BEF\_RENT\_FEE,

0 BEF\_OTHER\_FEE,

A.POST\_DISCOUNTFEE/1000 COMM\_FEE,

0 INFO\_FEE,

0 RENT\_FEE,

0 OTHER\_FEE,

A.DISCOUNTFEE/1000 COMM\_DISCOUNT,

0 INFO\_DISCOUNT,

0 RENT\_DISCOUNT,

0 OTHER\_DISCOUNT,

0 GIFT\_FEE,

A.PLAN\_ID TARIFF\_PLANS\_ID,

-1 COMPENSATION\_AMOUNT,

-1 SMS\_FORWARD\_ISMG,

-1 SMS\_CENTER\_CODE,

-1 IS\_FREE,

-1 FREE\_FLAG

from TG\_CDRMM\_SP A

## 移动业务流量详单（月）DWD\_D\_USE\_MB\_GPRS

### 建表语句

create table IF NOT EXISTS DWD\_D\_USE\_MB\_GPRS(MONTH\_ID string,

PROV\_ID string,

TICKET\_ID string,

ORIGINAL\_FILE string,

ACQUISITION\_TIME string,

INSERT\_TIME string,

SUBS\_INSTANCE\_ID string,

SERVICE\_TYPE string,

SERVICE\_TYPE\_PROV string,

DEVICE\_NUMBER string,

PRODUCT\_ID string,

IMSI string,

IMEI string,

INCR\_TYPE string,

NET\_TYPE string,

SVC\_TYPE string,

START\_DATE string,

START\_TIME string,

USE\_DURATION int,

SGSN string,

GGSN string,

MSRN string,

MSC string,

RA string,

LAC string,

CELL\_ID string,

CELL\_TYPE string,

HOME\_AREA\_CODE string,

VISIT\_AREA\_CODE string,

ROAM\_TYPE string,

PDP\_CHARGING\_ID string,

PDP\_START\_TYPE string,

PDP\_TYPE string,

PDP\_ADDRESS string,

APN\_OPERATOR string,

APNNI string,

SGSN\_CHANGE string,

SGSN\_CLOSE\_CAUSE string,

IS\_FREE string,

ASCEND\_BYTES INT,

DOWN\_BYTES INT,

ASCEND\_CACULATE\_BYTES INT,

DOWN\_CACULATE\_BYTES INT,

DINNER\_OUT\_BYTES INT,

DINNER\_IN\_BYTES INT,

BEF\_COMM\_FEE INT,

BEF\_INFO\_FEE INT,

BEF\_OTHER\_FEE INT,

COMM\_FEE INT,

INFO\_FEE INT,

OTHER\_FEE INT,

COMM\_DISCOUNT INT,

INFO\_DISCOUNT INT,

OTHER\_DISCOUNT INT,

TARIFF\_PLANS\_ID string,

IP string,

USER\_ACCOUNT string,

FREE\_FLAG string,

CALL\_END\_TIME string,

FEE\_TYPE string

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

insert overwrite table DWD\_D\_USE\_MB\_GPRS PARTITION

(month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

select '#CURDATEYYYYMM#' MONTH\_ID,

'#PROVINCE#' PROV\_ID,

-1 TICKET\_ID,

-1 ORIGINAL\_FILE,

-1 ACQUISITION\_TIME,

-1 INSERT\_TIME,

-1 SUBS\_INSTANCE\_ID,

A.BIZ\_TYPE SERVICE\_TYPE,

A.BIZ\_TYPE SERVICE\_TYPE\_PROV,

-1 DEVICE\_NUMBER,

A\_PRODUCT\_ID PRODUCT\_ID,

IMSI\_NUMBER IMSI,

A.IMEI IMEI,

-1 INCR\_TYPE,

-1 NET\_TYPE,

-1 SVC\_TYPE,

A.START\_DATE START\_DATE,

A.START\_TIME START\_TIME,

round(A.CALL\_DURATION/60, 2) USE\_DURATION,

A.SGSN SGSN,

A.GGSN GGSN,

A.MSNC MSRN,

-1 MSC,

A.RA RA,

A.LAC LAC,

A.CELL\_ID CELL\_ID,

-1 CELL\_TYPE,

HOME\_AREA\_CODE HOME\_AREA\_CODE,

VISIT\_AREA\_CODE VISIT\_AREA\_CODE,

ROAM\_TYPE ROAM\_TYPE,

CHARGING\_ID PDP\_CHARGING\_ID,

-1 PDP\_START\_TYPE,

PDP\_TYPE PDP\_TYPE,

SPA PDP\_ADDRESS,

A.APNOI APN\_OPERATOR,

A.APNNI APNNI,

A.SGSN\_CHANGE SGSN\_CHANGE,

-1 SGSN\_CLOSE\_CAUSE,

-1 IS\_FREE,

(DATA\_UP1 + DATA\_UP2) / 1024 ASCEND\_BYTES,

(DATA\_DOWN1 + DATA\_DOWN2) / 1024 DOWN\_BYTES,

-1 ASCEND\_CACULATE\_BYTES,

-1 DOWN\_CACULATE\_BYTES,

-1 DINNER\_OUT\_BYTES,

-1 DINNER\_IN\_BYTES,

-1 BEF\_COMM\_FEE,

-1 BEF\_INFO\_FEE,

-1 BEF\_OTHER\_FEE,

A.CFEE / 1000 COMM\_FEE,

-1 INFO\_FEE,

-1 OTHER\_FEE,

A.DIS\_CFEE / 1000 COMM\_DISCOUNT,

-1 INFO\_DISCOUNT,

-1 OTHER\_DISCOUNT,

-1 TARIFF\_PLANS\_ID,

-1 IP,

A.USER\_ID USER\_ACCOUNT,

-1 FREE\_FLAG,

-1 CALL\_END\_TIME,

A.FEE\_TYPE FEE\_TYPE

from TG\_CDRMM\_GS A

# DWA加工脚本

## [用户基本衍生表(月)](file:///C:\Users\zhulf\AppData\Roaming\Microsoft\DWD、DWA测试模型_汇总_20130514.xlsx#'用户基本衍生表(月)'!A1)DWA\_V\_M\_CUS\_USER\_INFO

### 建表语句

create table IF NOT EXISTS DWA\_V\_M\_CUS\_USER\_INFO

(MONTH\_ID string,

PROV\_ID string,

USER\_ID string,

CUST\_ID string,

DEVICE\_NUMBER string,

SERVICE\_TYPE string,

PAY\_MODE string,

PAY\_MODE\_PROV string,

USER\_TYPE string,

USER\_TYPE\_PROV string,

PRODUCT\_ID string,

PRODUCT\_TYPE string,

BRAND\_ID string,

INNET\_AREA\_ID string,

AREA\_ID string,

AREA\_ID\_PROV string,

USER\_STATUS string,

USER\_STATUS\_PROV string,

UPDATE\_DATE string,

CHANNEL\_ID string,

CHANNEL\_TYPE string,

OPER\_CHANNEL\_ID string,

OPER\_DATE string,

CREATE\_DATE string,

INNET\_DATE string,

INNET\_MONTHS int,

CLOSE\_DATE string,

START\_DATE string,

END\_DATE string,

REMOVE\_TAG string,

IS\_THIS\_DEV string,

IS\_THIS\_BREAK string,

IS\_CARD string,

IS\_INNET string,

IS\_ADD string,

IS\_ADD\_TYPE string,

IS\_LOST string,

IS\_LOST\_TYPE string,

IS\_ACTIVE\_USER string,

IS\_THIS\_ACCT string,

IS\_THREE\_NOT string

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

INSERT overwrite table DWA\_V\_M\_CUS\_USER\_INFO PARTITION(month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

SELECT '#CURDATEYYYYMM#' MONTH\_ID,

t.PROV\_ID PROV\_ID,

T.USER\_ID USER\_ID,

T.CUST\_ID CUST\_ID,

T.DEVICE\_NUMBER,

T.SERVICE\_TYPE,

T.PAY\_MODE,

T.PAY\_MODE\_PROV,

T.USER\_TYPE,

T.USER\_TYPE\_PROV,

T.PRODUCT\_ID PRODUCT\_ID,

-1 RODUCT\_TYPE,

T.BRAND\_ID,

T.INNET\_AREA\_ID,

T.AREA\_ID,

T.AREA\_ID\_PROV,

T.USER\_STATUS,

T.USER\_STATUS\_PROV,

T.UPDATE\_DATE,

T.CHANNEL\_ID,

-1 CHANNEL\_TYPE,

T.OPER\_CHANNEL\_ID,

T.OPER\_DATE,

T.CREATE\_DATE,

T.INNET\_DATE,

CASE WHEN T.IS\_INNET='1' THEN

cast('#CURDATEYYYYMM#' as bigint)-cast(T.INNET\_DATE as bigint)+1

ELSE

cast(T.CLOSE\_DATE as bigint)-cast(T.INNET\_DATE as bigint)+1

END INNET\_MONTHS,

T.CLOSE\_DATE,

T.START\_DATE,

T.END\_DATE,

T.REMOVE\_TAG REMOVE\_TAG,

CASE WHEN T.IS\_THIS\_DEV IS NULL THEN '0' ELSE T.IS\_THIS\_DEV END IS\_THIS\_DEV,

CASE WHEN T.IS\_THIS\_BREAK IS NULL THEN '0' ELSE T.IS\_THIS\_BREAK END IS\_THIS\_BREAK,

-1 IS\_CARD,

T.IS\_INNET IS\_INNET,

CASE WHEN T.IS\_THIS\_DEV IS NULL THEN '0' ELSE T.IS\_THIS\_DEV END IS\_ADD,

-1 IS\_ADD\_TYPE,

CASE WHEN T.IS\_THIS\_BREAK IS NULL THEN '0' ELSE T.IS\_THIS\_BREAK END IS\_LOST,

-1 IS\_LOST\_TYPE,

CASE WHEN (A.SUBS\_INSTANCE\_ID IS NOT NULL) OR (B.SUBS\_INSTANCE\_ID IS NOT NULL) OR (C.SUBS\_INSTANCE\_ID IS NOT NULL) THEN '1' ELSE '0' END IS\_ACTIVE\_USER,

CASE WHEN T.IS\_THIS\_ACCT is null then '0' else T.IS\_THIS\_ACCT end IS\_THIS\_ACCT,

CASE WHEN (A.SUBS\_INSTANCE\_ID IS NULL) AND (B.SUBS\_INSTANCE\_ID IS NULL) AND (C.SUBS\_INSTANCE\_ID IS NULL) and (d. charge\_counts>0) THEN '1' ELSE '0' END IS\_THREE\_NOT

FROM (select \* from DWD\_M\_PRD\_USER\_INFO T WHERE T.month = '#CURDATEYYYYMM#'AND PRO=‘#PROVINCE#’) t

left outer join (select SUBS\_INSTANCE\_ID,count(\*) from DWD\_D\_USE\_MB\_VOICE A where A.month ='#CURDATEYYYYMM#' AND PRO=‘#PROVINCE#’GROUP BY SUBS\_INSTANCE\_ID) a

on (T.USER\_ID=A.SUBS\_INSTANCE\_ID)

left outer join (select SUBS\_INSTANCE\_ID,count(\*) from DWD\_D\_USE\_MB\_SMS B where B.month ='#CURDATEYYYYMM#' AND PRO=‘#PROVINCE#’GROUP BY SUBS\_INSTANCE\_ID) b

on (T.USER\_ID=B.SUBS\_INSTANCE\_ID)

left outer join (select SUBS\_INSTANCE\_ID,count(\*) from DWD\_D\_USE\_MB\_GPRS C where C.month ='#CURDATEYYYYMM#' AND PRO=‘#PROVINCE#’ GROUP BY SUBS\_INSTANCE\_ID) c

on (T.USER\_ID=C.SUBS\_INSTANCE\_ID)

left outer join (select user\_id,count(\*) charge\_counts from DWD\_M\_ACC\_AL\_DIVIDE\_CHARGEd where d.month ='#CURDATEYYYYMM#' AND PRO=‘#PROVINCE#’

group by user\_id) d

on (T.USER\_ID=d.user\_id);

## [用户活动信息表(月)](file:///C:\Users\zhulf\AppData\Roaming\Microsoft\DWD、DWA测试模型_汇总_20130514.xlsx#'用户活动信息表(月)'!A1)DWA\_V\_M\_CUS\_ACT\_INFO

### 建表语句

create table IF NOT EXISTS DWA\_V\_M\_CUS\_ACT\_INFO

(MONTH\_ID string,

PROV\_ID string,

AREA\_ID string,

AREA\_ID\_PROV string,

ACTIVITY\_SUBS\_ID string,

SERVICE\_TYPE string,

SERVICE\_TYPE\_PROV string,

SUBS\_INSTANCE\_ID string,

DEVICE\_NUMBER string,

PAY\_MODE string,

PAY\_MODE\_PROV string,

USER\_TYPE string,

USER\_TYPE\_PROV string,

CHANNEL\_ID string,

IS\_GRP\_MBR string,

GROUP\_ID string,

CUST\_TYPE string,

PRODUCT\_NO string,

PRODUCT\_MODE string,

ACTIVITY\_ID string,

ACTIVITY\_ID\_PROV string,

ACTIVITY\_TYPE string,

ACTIVITY\_TYPE\_PROV string,

PROTOCAL\_MONTH string,

MON\_LIMIT\_CHARGE string,

RES\_TYPE string,

RES\_BELONG\_TYPE string,

RES\_NUMBER string,

RES\_BRAND string,

RES\_BRAND\_PROV string,

RES\_MODE string,

RES\_MODE\_PROV string,

RES\_COUNTS string,

RES\_COST string,

RES\_PURCHASE\_CHARGE int,

SUBSIDY int,

DEPOSIT\_CHARGE int,

PREPAY\_CHARGE int,

PROM\_CHARGE int,

MON\_RETURN\_RATE int,

OPER\_DATE string,

EFF\_DATE string,

EXP\_DATE string,

IS\_VALID string,

IS\_THIS\_EXPIRE string,

IS\_THIS\_ACCT string,

IS\_THIS\_INNET string,

IS\_THIS\_ADD string,

IS\_THIS\_NEW string,

IS\_THIS\_BREAK string,

IS\_THIS\_LOST string,

IS\_CARD string

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

INSERT overwrite table DWA\_V\_M\_CUS\_ACT\_INFO PARTITION(month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

SELECT '#CURDATEYYYYMM#' MONTH\_ID,

a.PROV\_ID PROV\_ID,

C.AREA\_ID,

C.AREA\_ID\_PROV,

A.ACTIVITY\_SUBS\_ID,

A.SERVICE\_TYPE,

A.SERVICE\_TYPE\_PROV,

A.USER\_ID,

A.DEVICE\_NUMBER,

C.PAY\_MODE,

C.PAY\_MODE\_PROV,

C.USER\_TYPE,

C.USER\_TYPE\_PROV,

C.CHANNEL\_ID,

-1 IS\_GRP\_MBR,

-1 GROUP\_ID,

-1 CUST\_TYPE,

A.PROD\_ID,

-1 PRODUCT\_MODE,

A.ACTIVE\_ID,

A.ACTIVE\_ID\_PROV,

A.ACTIVE\_TYPE,

A.ACTIVE\_TYPE\_PROV,

A.AGREEMENT\_MONTHS,

A.MINIMUM\_CONSUME,

A.RES\_TYPE,

A.RES\_AREA\_ID,

A.RES\_IMEI,

A.RES\_BRAND,

A.RES\_BRAND\_PROV,

A.RES\_MODEL,

A.RES\_MODE\_PROV,

A.RES\_NUM,

A.RES\_COST,

(CASE WHEN A.RES\_COST IS NULL THEN 0 ELSE A.RES\_COST END) ,

(CASE WHEN A.ALLOWANCE\_FEE IS NULL THEN 0 ELSE A.ALLOWANCE\_FEE END) ,

(CASE WHEN A.DEPOSIT\_FEE IS NULL THEN 0 ELSE A.DEPOSIT\_FEE END),

(CASE WHEN A.STORED\_FEE IS NULL THEN 0 ELSE A.STORED\_FEE END),

(CASE WHEN A.PRESENT\_FEE IS NULL THEN 0 ELSE A.PRESENT\_FEE END),

(CASE WHEN A.RETURN\_RATIO IS NULL THEN 0 ELSE A.RETURN\_RATIO END),

A.OPER\_DATE,

A.START\_DATE EFF\_DATE,

A.END\_DATE EXP\_DATE,

'1' IS\_VALID,

(CASE WHEN SUBSTR(A.END\_DATE,1,6)='#CURDATEYYYYMM#' THEN '1' ELSE '0' END) IS\_THIS\_EXPIRE ,

(CASE WHEN C.IS\_THIS\_ACCT='1' THEN '1' ELSE '0' END),

(CASE WHEN C.IS\_INNET='1' THEN '1' ELSE '0' END) IS\_THIS\_INNET ,

(CASE WHEN SUBSTR(A.OPER\_DATE,1,6)='#CURDATEYYYYMM#' THEN '1' ELSE '0' END) IS\_THIS\_ADD,

(CASE WHEN IS\_THIS\_DEV='1' AND SUBSTR(A.OPER\_DATE,1,6)='#CURDATEYYYYMM#' THEN '1' ELSE '0' END) IS\_THIS\_NEW,

(CASE WHEN C.IS\_THIS\_BREAK='1' AND substr(A.END\_DATE,1,6)>='#CURDATEYYYYMM#' AND SUBSTR(A.START\_DATE,1,6)<='#CURDATEYYYYMM#' THEN '1' ELSE '0' END) IS\_THIS\_BREAK,

(CASE WHEN C.IS\_THIS\_BREAK='1' AND substr(A.END\_DATE,1,6)>='#CURDATEYYYYMM#' AND SUBSTR(A.START\_DATE,1,6)<='#CURDATEYYYYMM#' THEN '1' ELSE '0' END) IS\_THIS\_LOST,

-1 IS\_CARD

FROM DWD\_M\_PRD\_AL\_USER\_ACTIVE a

left outer join

(SELECT B.\*

FROM DWD\_M\_PRD\_USER\_INFO B

WHERE b.month = '#CURDATEYYYYMM#'

AND PRO=‘#PROVINCE#’) C

on (A.USER\_ID = C.USER\_ID)

WHERE a.month = '#CURDATEYYYYMM#'

AND PRO=‘#PROVINCE#’

## [移动出帐衍生信息(分摊前)-月](file:///C:\Users\zhulf\AppData\Roaming\Microsoft\DWD、DWA测试模型_汇总_20130514.xlsx#'移动出帐衍生信息(分摊前)-月'!A1)DWA\_V\_M\_CUS\_MB\_BCHARGE

### 建表语句

create table IF NOT EXISTS DWA\_V\_M\_CUS\_MB\_BCHARGE

(MONTH\_ID string,

PROV\_ID string,

SUBS\_INSTANCE\_ID string,

AREA\_ID string,

SERVICE\_TYPE string,

DEVICE\_NUMBER string,

CUST\_ID string,

NOMAL\_TOTAL\_FEE INT,

BASE\_RENT\_FEE INT,

DINNER\_RENT\_FEE INT,

FUNCATION\_RENT\_FEE INT,

OTHER\_RENT\_FEE INT,

BASE\_CALL\_FEE INT,

COUN\_LONG\_LFEE INT,

INTER\_LONG\_LFEE INT,

GAT\_LONG\_LFEE INT,

COUN\_ROAM\_BFEE INT,

INTER\_ROAM\_BFEE INT,

GAT\_RAOM\_BFEE INT,

OTHER\_CALL\_LFEE INT,

P2P\_FEE INT,

LTZX\_FEE INT,

GPRS\_FEE INT,

INCRE\_FEE INT,

GRANT\_FEE INT,

OTHER\_FEE INT

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

insert overwrite table DWA\_V\_M\_CUS\_MB\_BCHARGE partition (month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

select A.MONTH\_ID MONTH\_ID,

A.PROV\_ID PROV\_ID,

A.USER\_ID SUBS\_INSTANCE\_ID,

B.AREA\_ID AREA\_ID,

B.SERVICE\_TYPE SERVICE\_TYPE,

B.DEVICE\_NUMBER DEVICE\_NUMBER,

B.CUST\_ID CUST\_ID,

SUM(COALESCE(FEE,0)) NOMAL\_TOTAL\_FEE,

SUM(case when A.ACCT\_ITEM\_ID ='A10001A20001000000001' then COALESCE(A.FEE,0) else 0 end) BASE\_RENT\_FEE,

SUM(case when A.ACCT\_ITEM\_ID ='D4020000000000000000001' then COALESCE(A.FEE,0) else 0 end) DINNER\_RENT\_FEE,

SUM(case when A.ACCT\_ITEM\_ID ='B1010001P01010110999999' then COALESCE(A.FEE,0) else 0 end) FUNCATION\_RENT\_FEE,

0 OTHER\_RENT\_FEE,

SUM(case when A.ACCT\_ITEM\_ID ='B1010001P01010100020101' then COALESCE(A.FEE,0) else 0 end) BASE\_CALL\_FEE,

SUM(case when A.ACCT\_ITEM\_ID ='B1010001P01010130000003' then COALESCE(A.FEE,0) else 0 end) COUN\_LONG\_LFEE,

SUM(case when A.ACCT\_ITEM\_ID ='B1010001P01010130000004' then COALESCE(A.FEE,0) else 0 end) INTER\_LONG\_LFEE,

SUM(case when A.ACCT\_ITEM\_ID ='B1010001P01010130000005' then COALESCE(A.FEE,0) else 0 end) GAT\_LONG\_LFEE,

SUM(case when A.ACCT\_ITEM\_ID ='B1010001P01010100020119' then COALESCE(A.FEE,0) else 0 end) COUN\_ROAM\_BFEE,

SUM(case when A.ACCT\_ITEM\_ID ='B1010001P01010100020120' then COALESCE(A.FEE,0) else 0 end) INTER\_ROAM\_BFEE,

SUM(case when A.ACCT\_ITEM\_ID ='B1010001P01010100020121' then COALESCE(A.FEE,0) else 0 end) GAT\_RAOM\_BFEE,

SUM(case when A.ACCT\_ITEM\_ID ='B1010001P01010130999999' then COALESCE(A.FEE,0) else 0 end) OTHER\_CALL\_LFEE,

SUM(case when A.ACCT\_ITEM\_ID ='B4020201P07030100110001' then COALESCE(A.FEE,0) else 0 end) P2P\_FEE,

SUM(case when A.ACCT\_ITEM\_ID in ('B4020201P07030400110000','B4020201P07030400050000') then COALESCE(A.FEE,0) else 0 end) LTZX\_FEE,

SUM(case when A.ACCT\_ITEM\_ID ='B2020117P04020200030000' then COALESCE(A.FEE,0) else 0 end) GPRS\_FEE,

SUM(case when A.ACCT\_ITEM\_ID not in ('A10001A20001000000001','D4020000000000000000001','B1010001P01010110999999','B1010001P01010100020101','B1010001P01010130000003',

'B1010001P01010130000004','B1010001P01010130000005','B1010001P01010100020119','B1010001P01010100020120','B1010001P01010100020121',

'B1010001P01010130999999','B4020201P07030100110001','B4020201P07030400110000','B4020201P07030400050000','B2020117P04020200030000','D4020000000000000000023') then COALESCE(A.FEE,0) else 0 end) INCRE\_FEE,

SUM(case when A.ACCT\_ITEM\_ID ='D4020000000000000000023' then COALESCE(A.FEE,0) else 0 end) GRANT\_FEE,

0 OTHER\_FEE

from (select \* from DWD\_M\_ACC\_AL\_DIVIDE\_CHARGE A where a.month ='#CURDATEYYYYMM#' and pro=‘#PROVINCE#’) a

join

(select \* from DWD\_M\_PRD\_USER\_INFO B where b.month ='#CURDATEYYYYMM#' and pro=‘#PROVINCE#’) b

ona.user\_id = b.user\_id

GROUP BY A.MONTH\_ID,

A.PROV\_ID,

A.USER\_ID,

B.AREA\_ID,

B.SERVICE\_TYPE,

B.DEVICE\_NUMBER,

B.CUST\_ID

## [用户语音信息(月)](file:///C:\Users\zhulf\AppData\Roaming\Microsoft\DWD、DWA测试模型_汇总_20130514.xlsx#'用户语音信息(月)'!A1)DWA\_V\_M\_CUS\_SING\_VOICE

### 建表语句

create table IF NOT EXISTS DWA\_V\_M\_CUS\_SING\_VOICE

(MONTH\_ID string,

PROV\_ID string,

AREA\_ID string,

CITY\_ID string,

SUBS\_INSTANCE\_ID string,

SERVICE\_TYPE string,

TOTAL\_DURA INT,

LOCAL\_DURA INT,

LOCAL\_CALLING\_DURA INT,

LOCAL\_CALLED\_DURA INT,

TOLL\_DURA INT,

TOLL\_PROV\_CALLING\_DURA INT,

TOLL\_COUN\_CALLING\_DURA INT,

TOLL\_GAT\_CALLING\_DURA INT,

TOLL\_INTER\_CALLING\_DURA INT,

ROAM\_DURA INT,

ROAM\_PROV\_CALLING\_DURA INT,

ROAM\_PROV\_GATL\_DURA INT,

ROAM\_PROV\_INTER\_DURA INT,

ROAM\_COUN\_CALLING\_DURA INT,

ROAM\_COUN\_GATL\_DURA INT,

ROAM\_COUN\_INTER\_DURA INT,

ROAM\_GAT\_CALLING\_DURA INT,

ROAM\_INTER\_CALLING\_DURA INT,

ROAM\_PROV\_CALLED\_DURA INT,

ROAM\_COUN\_CALLED\_DURA INT,

ROAM\_GAT\_CALLED\_DURA INT,

ROAM\_INTER\_CALLED\_DURA INT,

TOTAL\_TIMES INT,

LOCAL\_TIMES INT,

LOCAL\_CALLING\_TIMES INT,

LOCAL\_CALLIED\_TIMES INT,

TOLL\_TIMES INT,

TOLL\_PROV\_CALLING\_TIMES INT,

TOLL\_COUN\_CALLING\_TIMES INT,

TOLL\_GAT\_CALLING\_TIMES INT,

TOLL\_INTER\_CALLING\_TIMES INT,

ROAM\_TIMES INT,

ROAM\_PROV\_CALLING\_TIMES INT,

ROAM\_PROV\_GATL\_TIMES INT,

ROAM\_PROV\_INTER\_TIMES INT,

ROAM\_COUN\_CALLING\_TIMES INT,

ROAM\_COUN\_GATL\_TIMES INT,

ROAM\_COUN\_INTER\_TIMES INT,

ROAM\_GAT\_CALLING\_TIMES INT,

ROAM\_INTER\_CALLING\_TIMES INT,

ROAM\_PROV\_CALLED\_TIMES INT,

ROAM\_COUN\_CALLED\_TIMES INT,

ROAM\_GAT\_CALLED\_TIMES INT,

ROAM\_INTER\_CALLED\_TIMES INT,

TOTAL\_L\_TIMES INT,

TOLL\_L\_TIMES INT,

TOLL\_PROV\_CALLING\_L\_TIMES INT,

TOLL\_COUN\_CALLING\_L\_TIMES INT,

TOLL\_GAT\_CALLING\_L\_TIMES INT,

TOLL\_INTER\_CALLING\_L\_TIMES INT,

TOTAL\_NUMS INT,

LOCAL\_NUMS INT,

LOCAL\_CALLING\_NUMS INT,

LOCAL\_CALLED\_NUMS INT,

TOLL\_NUMS INT,

TOLL\_PROV\_CALLING\_NUMS INT,

TOLL\_COUN\_CALLING\_NUMS INT,

TOLL\_GAT\_CALLING\_NUMS INT,

TOLL\_INTER\_CALLING\_NUMS INT,

ROAM\_NUMS INT,

ROAM\_PROV\_CALLING\_NUMS INT,

ROAM\_PROV\_GATL\_NUMS INT,

ROAM\_PROV\_INTER\_NUMS INT,

ROAM\_COUN\_CALLING\_NUMS INT,

ROAM\_COUN\_GATL\_NUMS INT,

ROAM\_COUN\_INTER\_NUMS INT,

ROAM\_GAT\_CALLING\_NUMS INT,

ROAM\_INTER\_CALLING\_NUMS INT,

ROAM\_PROV\_CALLED\_NUMS INT,

ROAM\_COUN\_CALLED\_NUMS INT,

ROAM\_GAT\_CALLED\_NUMS INT,

ROAM\_INTER\_CALLED\_NUMS INT,

LOCAL\_CALLING\_DURA\_3G INT,

TOLL\_CALLING\_DURA\_3G INT,

ROAM\_CALLING\_DURA\_3G INT,

LOCAL\_CALLED\_DURA\_3G INT,

ROAM\_CALLED\_DURA\_3G INT,

LOCAL\_CALLING\_DURA\_2G INT,

TOLL\_CALLING\_DURA\_2G INT,

ROAM\_CALLING\_DURA\_2G INT,

LOCAL\_CALLED\_DURA\_2G INT,

ROAM\_CALLED\_DURA\_2G INT,

LOCAL\_CALLING\_TIMES\_3G INT,

TOLL\_CALLING\_TIMES\_3G INT,

ROAM\_CALLING\_TIMES\_3G INT,

LOCAL\_CALLED\_TIMES\_3G INT,

ROAM\_CALLED\_TIMES\_3G INT,

LOCAL\_CALLING\_TIMES\_2G INT,

TOLL\_CALLING\_TIMES\_2G INT,

ROAM\_CALLING\_TIMES\_2G INT,

LOCAL\_CALLED\_TIMES\_2G INT,

ROAM\_CALLED\_TIMES\_2G INT,

LOCAL\_CALLING\_NUM\_3G INT,

TOLL\_CALLING\_NUM\_3G INT,

ROAM\_CALLING\_NUM\_3G INT,

LOCAL\_CALLED\_NUM\_3G INT,

ROAM\_CALLED\_NUM\_3G INT,

LOCAL\_CALLING\_NUM\_2G INT,

TOLL\_CALLING\_NUM\_2G INT,

ROAM\_CALLING\_NUM\_2G INT,

LOCAL\_CALLED\_NUM\_2G INT,

ROAM\_CALLED\_NUM\_2G INT,

BEF\_BASE\_FEE INT,

BEF\_TOLL\_FEE INT,

BEF\_INFO\_FEE INT,

BEF\_OTHER\_FEE INT,

BASE\_FEE INT,

TOLL\_FEE INT,

INFO\_FEE INT,

OTHER\_FEE INT,

IN\_CNT INT,

OUT\_CNT INT,

IN\_DURA INT,

OUT\_DURA INT,

JF\_TIMES INT,

IN\_JF\_TIMES INT,

ROAM\_IN\_JF\_TIMES INT,

NOROAM\_LOCAL\_JF\_TIMES INT,

NOROAM\_LONG\_JF\_TIMES INT,

NOROAM\_GN\_LONG\_JF\_TIMES INT,

NOROAM\_GJ\_LONG\_JF\_TIMES INT,

NOROAM\_GAT\_LONG\_JF\_TIMES INT,

ROAM\_LOCAL\_JF\_TIMES INT,

ROAM\_GN\_LONG\_JF\_TIMES INT,

ROAM\_GJ\_LONG\_JF\_TIMES INT,

ROAM\_GAT\_LONG\_JF\_TIMES INT,

NOROAM\_LONG\_CNT INT,

NOROAM\_LOCAL\_CNT INT,

ROAM\_LOCAL\_CNT INT,

NOROAM\_GN\_LONG\_CNT INT,

ROAM\_GN\_LONG\_CNT INT,

ROAM\_GJ\_LONG\_CNT INT,

ROAM\_GAT\_LONG\_CNT INT,

NOROAM\_GAT\_LONG\_CNT INT,

NOROAM\_GJ\_LONG\_CNT int,

IS\_YW\_CS string

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

insert overwrite table DWA\_V\_M\_CUS\_SING\_VOICE partition (month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

select

A.MONTH\_ID MONTH\_ID ,

A.PROV\_ID PROV\_ID ,

B.AREA\_ID AREA\_ID ,

B.CITY\_ID CITY\_ID ,

A.SUBS\_INSTANCE\_ID SUBS\_INSTANCE\_ID ,

B.SERVICE\_TYPE SERVICE\_TYPE ,

sum(COALESCE(CALL\_DURATION ,0)) TOTAL\_DURA ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' then COALESCE(A.CALL\_DURATION ,0) else 0 end) LOCAL\_DURA ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) LOCAL\_CALLING\_DURA ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01202' then COALESCE(A.CALL\_DURATION ,0) else 0 end) LOCAL\_CALLED\_DURA ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' then COALESCE(A.CALL\_DURATION ,0) else 0 end) TOLL\_DURA ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH005020101' and A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) TOLL\_PROV\_CALLING\_DURA ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050201' and A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) TOLL\_COUN\_CALLING\_DURA ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050203' and A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) TOLL\_GAT\_CALLING\_DURA ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050202' and A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) TOLL\_INTER\_CALLING\_DURA ,

sum(case when A.ROAM\_TYPE='KH00602' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060201' and A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_PROV\_CALLING\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060201' and A.LONG\_TYPE='KH0050203' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_PROV\_GATL\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060201' and A.LONG\_TYPE='KH0050202' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_PROV\_INTER\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060202' and A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_COUN\_CALLING\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060202' and A.LONG\_TYPE='KH0050203' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_COUN\_GATL\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060202' and A.LONG\_TYPE='KH0050202' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_COUN\_INTER\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060203' and A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_GAT\_CALLING\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060204' and A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_INTER\_CALLING\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060201' and A.CALL\_TYPE='CP01202' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_PROV\_CALLED\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060202' and A.CALL\_TYPE='CP01202' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_COUN\_CALLED\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060203' and A.CALL\_TYPE='CP01202' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_GAT\_CALLED\_DURA ,

sum(case when A.ROAM\_TYPE='KH0060204' and A.CALL\_TYPE='CP01202' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_INTER\_CALLED\_DURA ,

sum( COALESCE(A.BASE\_TIMES+A.LONG\_TIMES ,0)) TOTAL\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' then COALESCE(A.BASE\_TIMES ,0) else 0 end) LOCAL\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) LOCAL\_CALLING\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01202' then COALESCE(A.BASE\_TIMES ,0) else 0 end) LOCAL\_CALLIED\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' then COALESCE(A.BASE\_TIMES ,0) else 0 end) TOLL\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH005020101' and A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) TOLL\_PROV\_CALLING\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050201' and A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) TOLL\_COUN\_CALLING\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050203' and A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) TOLL\_GAT\_CALLING\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050202' and A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) TOLL\_INTER\_CALLING\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00602' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060201' and A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_PROV\_CALLING\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060201' and A.LONG\_TYPE='KH0050203' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_PROV\_GATL\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060201' and A.LONG\_TYPE='KH0050202' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_PROV\_INTER\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060202' and A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_COUN\_CALLING\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060202' and A.LONG\_TYPE='KH0050203' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_COUN\_GATL\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060202' and A.LONG\_TYPE='KH0050202' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_COUN\_INTER\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060203' and A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_GAT\_CALLING\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060204' and A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_INTER\_CALLING\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060201' and A.CALL\_TYPE='CP01202' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_PROV\_CALLED\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060202' and A.CALL\_TYPE='CP01202' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_COUN\_CALLED\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060203' and A.CALL\_TYPE='CP01202' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_GAT\_CALLED\_TIMES ,

sum(case when A.ROAM\_TYPE='KH0060204' and A.CALL\_TYPE='CP01202' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_INTER\_CALLED\_TIMES ,

sum( COALESCE(A.LONG\_TIMES ,0)) TOTAL\_L\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' then COALESCE(A.LONG\_TIMES ,0) else 0 end) TOLL\_L\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH005020101' and A.CALL\_TYPE='CP01201' then COALESCE(A.LONG\_TIMES ,0) else 0 end) TOLL\_PROV\_CALLING\_L\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050201' and A.CALL\_TYPE='CP01201' then COALESCE(A.LONG\_TIMES ,0) else 0 end) TOLL\_COUN\_CALLING\_L\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050203' and A.CALL\_TYPE='CP01201' then COALESCE(A.LONG\_TIMES ,0) else 0 end) TOLL\_GAT\_CALLING\_L\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050202' and A.CALL\_TYPE='CP01201' then COALESCE(A.LONG\_TIMES ,0) else 0 end) TOLL\_INTER\_CALLING\_L\_TIMES ,

count(1) TOTAL\_NUMS ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' then '1' else null end) LOCAL\_NUMS ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01201' then '1' else null end) LOCAL\_CALLING\_NUMS ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01202' then '1' else null end) LOCAL\_CALLED\_NUMS ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' then '1' else null end) TOLL\_NUMS ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH005020101' and A.CALL\_TYPE='CP01201' then '1' else null end) TOLL\_PROV\_CALLING\_NUMS ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050201' and A.CALL\_TYPE='CP01201' then '1' else null end) TOLL\_COUN\_CALLING\_NUMS ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050203' and A.CALL\_TYPE='CP01201' then '1' else null end) TOLL\_GAT\_CALLING\_NUMS ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050202' then '1' else null end) TOLL\_INTER\_CALLING\_NUMS ,

count(case when A.ROAM\_TYPE='KH00602' then '1' else null end) ROAM\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060201' and A.CALL\_TYPE='CP01201' then '1' else null end) ROAM\_PROV\_CALLING\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060201' and A.LONG\_TYPE='KH0050203' then '1' else null end) ROAM\_PROV\_GATL\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060201' and A.LONG\_TYPE='KH0050202' then '1' else null end) ROAM\_PROV\_INTER\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060202' and A.CALL\_TYPE='CP01201' then '1' else null end) ROAM\_COUN\_CALLING\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060202' and A.LONG\_TYPE='KH0050203' then '1' else null end) ROAM\_COUN\_GATL\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060202' and A.LONG\_TYPE='KH0050202' then '1' else null end) ROAM\_COUN\_INTER\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060203' and A.CALL\_TYPE='CP01201' then '1' else null end) ROAM\_GAT\_CALLING\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060204' and A.CALL\_TYPE='CP01201' then '1' else null end) ROAM\_INTER\_CALLING\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060201' and A.CALL\_TYPE='CP01202' then '1' else null end) ROAM\_PROV\_CALLED\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060201' and A.CALL\_TYPE='CP01202' then '1' else null end) ROAM\_COUN\_CALLED\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060203' and A.CALL\_TYPE='CP01202' then '1' else null end) ROAM\_GAT\_CALLED\_NUMS ,

count(case when A.ROAM\_TYPE='KH0060204' and A.CALL\_TYPE='CP01202' then '1' else null end) ROAM\_INTER\_CALLED\_NUMS ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.CALL\_DURATION ,0) else 0 end) LOCAL\_CALLING\_DURA\_3G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.CALL\_DURATION ,0) else 0 end) TOLL\_CALLING\_DURA\_3G ,

sum(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_CALLING\_DURA\_3G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.CALL\_DURATION ,0) else 0 end) LOCAL\_CALLED\_DURA\_3G ,

sum(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_CALLED\_DURA\_3G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.CALL\_DURATION ,0) else 0 end) LOCAL\_CALLING\_DURA\_2G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.CALL\_DURATION ,0) else 0 end) TOLL\_CALLING\_DURA\_2G ,

sum(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_CALLING\_DURA\_2G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.CALL\_DURATION ,0) else 0 end) LOCAL\_CALLED\_DURA\_2G ,

sum(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.CALL\_DURATION ,0) else 0 end) ROAM\_CALLED\_DURA\_2G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.BASE\_TIMES ,0) else 0 end) LOCAL\_CALLING\_TIMES\_3G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.BASE\_TIMES ,0) else 0 end) TOLL\_CALLING\_TIMES\_3G ,

sum(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_CALLING\_TIMES\_3G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.BASE\_TIMES ,0) else 0 end) LOCAL\_CALLED\_TIMES\_3G ,

sum(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010102' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_CALLED\_TIMES\_3G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.BASE\_TIMES ,0) else 0 end) LOCAL\_CALLING\_TIMES\_2G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.BASE\_TIMES ,0) else 0 end) TOLL\_CALLING\_TIMES\_2G ,

sum(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_CALLING\_TIMES\_2G ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.BASE\_TIMES ,0) else 0 end) LOCAL\_CALLED\_TIMES\_2G ,

sum(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010101' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_CALLED\_TIMES\_2G ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010102' then '1' else null end) LOCAL\_CALLING\_NUM\_3G ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010102' then '1' else null end) TOLL\_CALLING\_NUM\_3G ,

count(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010102' then '1' else null end) ROAM\_CALLING\_NUM\_3G ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010102' then '1' else null end) LOCAL\_CALLED\_NUM\_3G ,

count(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010102' then '1' else null end) ROAM\_CALLED\_NUM\_3G ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010101' then '1' else null end) LOCAL\_CALLING\_NUM\_2G ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010101' then '1' else null end) TOLL\_CALLING\_NUM\_2G ,

count(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01201' and A.SERVICE\_TYPE='CP0010101' then '1' else null end) ROAM\_CALLING\_NUM\_2G ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010101' then '1' else null end) LOCAL\_CALLED\_NUM\_2G ,

count(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01202' and A.SERVICE\_TYPE='CP0010101' then '1' else null end) ROAM\_CALLED\_NUM\_2G ,

sum(COALESCE(A.BEF\_BASE\_FEE ,0)) BEF\_BASE\_FEE ,

sum(COALESCE(A.BEF\_LONG\_FEE ,0)) BEF\_TOLL\_FEE ,

sum(COALESCE(A.BEF\_INFO\_FEE ,0)) BEF\_INFO\_FEE ,

sum(COALESCE(A.BEF\_OTHER\_FEE ,0)) BEF\_OTHER\_FEE ,

sum(COALESCE(A.BASE\_FEE ,0)) BASE\_FEE ,

sum(COALESCE(A.LONG\_FEE ,0)) TOLL\_FEE ,

sum(COALESCE(A.INFO\_FEE ,0)) INFO\_FEE ,

sum(COALESCE(A.OTHER\_FEE ,0)) OTHER\_FEE ,

count(case when A.CALL\_TYPE='CP01202' then '1' else null end) IN\_CNT ,

count(case when A.CALL\_TYPE='CP01201' then '1' else null end) OUT\_CNT ,

sum(case when A.CALL\_TYPE='CP01202' then COALESCE(A.CALL\_DURATION ,0) else 0 end) IN\_DURA ,

sum(case when A.CALL\_TYPE='CP01201' then COALESCE(A.CALL\_DURATION ,0) else 0 end) OUT\_DURA ,

sum(case when A.CALL\_TYPE='CP01201' then COALESCE(A.BASE\_TIMES ,0) else 0 end) JF\_TIMES ,

sum(case when A.CALL\_TYPE='CP01202' then COALESCE(A.BASE\_TIMES ,0) else 0 end) IN\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00602' and A.CALL\_TYPE='CP01202' then COALESCE(A.BASE\_TIMES ,0) else 0 end) ROAM\_IN\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' then COALESCE(A.BASE\_TIMES ,0) else 0 end) NOROAM\_LOCAL\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' then COALESCE(A.LONG\_TIMES ,0) else 0 end) NOROAM\_LONG\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050201' then COALESCE(A.LONG\_TIMES ,0) else 0 end) NOROAM\_GN\_LONG\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050202' then COALESCE(A.LONG\_TIMES ,0) else 0 end) NOROAM\_GJ\_LONG\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050203' then COALESCE(A.LONG\_TIMES ,0) else 0 end) NOROAM\_GAT\_LONG\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00602' and A.LONG\_TYPE='KH00501' then COALESCE(A.LONG\_TIMES ,0) else 0 end) ROAM\_LOCAL\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00602' and A.LONG\_TYPE='KH0050201' then COALESCE(A.LONG\_TIMES ,0) else 0 end) ROAM\_GN\_LONG\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00602' and A.LONG\_TYPE='KH0050202' then COALESCE(A.LONG\_TIMES ,0) else 0 end) ROAM\_GJ\_LONG\_JF\_TIMES ,

sum(case when A.ROAM\_TYPE='KH00602' and A.LONG\_TYPE='KH0050203' then COALESCE(A.LONG\_TIMES ,0) else 0 end) ROAM\_GAT\_LONG\_JF\_TIMES ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00502' then '1' else null end) NOROAM\_LONG\_CNT ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH00501' then '1' else null end) NOROAM\_LOCAL\_CNT ,

count(case when A.ROAM\_TYPE='KH00602' and A.LONG\_TYPE='KH00501' then '1' else null end) ROAM\_LOCAL\_CNT ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050201' then '1' else null end) NOROAM\_GN\_LONG\_CNT ,

count(case when A.ROAM\_TYPE='KH00602' and A.LONG\_TYPE='KH0050201' then '1' else null end) ROAM\_GN\_LONG\_CNT ,

count(case when A.ROAM\_TYPE='KH00602' and A.LONG\_TYPE='KH0050202' then '1' else null end) ROAM\_GJ\_LONG\_CNT ,

count(case when A.ROAM\_TYPE='KH00602' and A.LONG\_TYPE='KH0050203' then '1' else null end) ROAM\_GAT\_LONG\_CNT ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050203' then '1' else null end) NOROAM\_GAT\_LONG\_CNT ,

count(case when A.ROAM\_TYPE='KH00601' and A.LONG\_TYPE='KH0050202' then '1' else null end) NOROAM\_GJ\_LONG\_CNT ,

case when yd\_call\_counts>0 and dx\_call\_counts>0 then '3'

whendx\_call\_counts>0 then '2'

whenyd\_call\_counts>0 then '1'

else '0' end IS\_YW\_CS

from (select \* from DWD\_D\_USE\_MB\_VOICE A where a.month ='#CURDATEYYYYMM#'and pro=‘#PROVINCE#’) A

join

(select \* from DWD\_M\_PRD\_USER\_INFO B where b.month ='#CURDATEYYYYMM#'and pro=‘#PROVINCE#’) B

join

(select SUBS\_INSTANCE\_ID,

sum(case when OPPOSE\_NUMBER='10086' then 1 else 0 end) yd\_call\_counts,

sum(case when OPPOSE\_NUMBER='10000' then 1 else 0 end) dx\_call\_counts

from DWD\_D\_USE\_MB\_VOICE a

wherea.month ='#CURDATEYYYYMM#'

and pro=‘#PROVINCE#’

group by SUBS\_INSTANCE\_ID) c

on A.SUBS\_INSTANCE\_ID = B.user\_id

and A.SUBS\_INSTANCE\_ID = c.SUBS\_INSTANCE\_ID

group by

A.MONTH\_ID,

A.PROV\_ID,

B.AREA\_ID,

B.CITY\_ID,

A.SUBS\_INSTANCE\_ID,

B.SERVICE\_TYPE ,

case when yd\_call\_counts>0 and dx\_call\_counts>0 then '3'

whendx\_call\_counts>0 then '2'

whenyd\_call\_counts>0 then '1'

else '0' end

## [用户流量表(月)](file:///C:\Users\zhulf\AppData\Roaming\Microsoft\DWD、DWA测试模型_汇总_20130514.xlsx#'用户流量表(月)'!A1)DWA\_V\_M\_CUS\_SING\_FLUX

### 建表语句

create table IF NOT EXISTS DWA\_V\_M\_CUS\_SING\_FLUX

(MONTH\_ID string,

PROV\_ID string,

SUBS\_INSTANCE\_ID string,

DEVICE\_NUMBER string,

SERVICE\_TYPE string,

FLUX\_NUM INT,

FREE\_NUM INT,

BILL\_NUM INT,

LOCAL\_NUM INT,

ROAM\_PROV\_NUM INT,

ROAM\_CONT\_NUM INT,

ROAM\_GAT\_NUM INT,

ROAM\_INT\_NUM INT,

FLUX\_2G\_NUM INT,

FLUX\_3G\_NUM INT,

TOTAL\_FLUX INT,

FREE\_FLUX INT,

BILL\_FLUX INT,

UP\_FLUX INT,

UP\_LOCAL\_FLUX INT,

UP\_ROAM\_PROV\_FLUX INT,

UP\_ROAM\_CONT\_FLUX INT,

UP\_ROAM\_GAT\_FLUX INT,

UP\_ROAM\_INT\_FLUX INT,

UP\_2G\_FLUX INT,

UP\_3G\_FLUX INT,

DOWN\_FLUX INT,

DOWN\_LOCAL\_FLUX INT,

DOWN\_ROAM\_PROV\_FLUX INT,

DOWN\_ROAM\_CONT\_FLUX INT,

DOWN\_ROAM\_GAT\_FLUX INT,

DOWN\_ROAM\_INT\_FLUX INT,

DOWN\_2G\_FLUX INT,

DOWN\_3G\_FLUX INT,

TOTAL\_DURA INT,

FREE\_DURA INT,

BILL\_DURA INT,

FLUX\_2G\_DURA INT,

FLUX\_3G\_DURA INT,

TOTAL\_FEE INT,

PROD\_IN\_LOCAL\_FLUX INT,

PROD\_IN\_ROAM\_CONT\_FLUX INT,

PROD\_IN\_ROAM\_GAT\_FLUX INT,

PROD\_IN\_ROAM\_INT\_FLUX INT

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工语句

insert overwrite table DWA\_V\_M\_CUS\_SING\_FLUX partition (month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

select

A.MONTH\_ID MONTH\_ID ,

A.PROV\_ID PROV\_ID ,

A.SUBS\_INSTANCE\_ID SUBS\_INSTANCE\_ID ,

A.DEVICE\_NUMBER DEVICE\_NUMBER ,

A.SERVICE\_TYPE SERVICE\_TYPE ,

count(\*) FLUX\_NUM ,

count(case when A.COMM\_FEE+A.INFO\_FEE+A.OTHER\_FEE=0 then '1' else null end) FREE\_NUM ,

count(case when A.COMM\_FEE+A.INFO\_FEE+A.OTHER\_FEE<>0 then '1' else null end) BILL\_NUM ,

count(case when A.ROAM\_TYPE='KH00601' then '1' else null end) LOCAL\_NUM ,

count(case when A.ROAM\_TYPE='KH0060201' then '1' else null end) ROAM\_PROV\_NUM ,

count(case when A.ROAM\_TYPE='KH0060202' then '1' else null end) ROAM\_CONT\_NUM ,

count(case when A.ROAM\_TYPE='KH0060203' then '1' else null end) ROAM\_GAT\_NUM ,

count(case when A.ROAM\_TYPE='KH0060204' then '1' else null end) ROAM\_INT\_NUM ,

count(case when A.ROAM\_TYPE='KH00601' and A.SERVICE\_TYPE='CP0010101' then '1' else null end) FLUX\_2G\_NUM ,

count(case when A.SERVICE\_TYPE='CP0010102' then '1' else null end) FLUX\_3G\_NUM ,

sum(COALESCE( A.ASCEND\_BYTES+A.DOWN\_BYTES ,0)) TOTAL\_FLUX ,

sum(case when A.COMM\_FEE+A.INFO\_FEE+A.OTHER\_FEE=0 then COALESCE( A.ASCEND\_BYTES+A.DOWN\_BYTES ,0) else 0 end) FREE\_FLUX ,

sum(case when A.COMM\_FEE+A.INFO\_FEE+A.OTHER\_FEE<>0 then COALESCE( A.ASCEND\_BYTES+A.DOWN\_BYTES ,0) else 0 end) BILL\_FLUX ,

sum(COALESCE( A.ASCEND\_BYTES ,0)) UP\_FLUX ,

sum(case when A.ROAM\_TYPE='KH00601' then COALESCE( A.ASCEND\_BYTES ,0) else 0 end) UP\_LOCAL\_FLUX ,

sum(case when A.ROAM\_TYPE='KH0060201' then COALESCE( A.ASCEND\_BYTES ,0) else 0 end) UP\_ROAM\_PROV\_FLUX ,

sum(case when A.ROAM\_TYPE='KH0060202' then COALESCE( A.ASCEND\_BYTES ,0) else 0 end) UP\_ROAM\_CONT\_FLUX ,

sum(case when A.ROAM\_TYPE='KH0060203' then COALESCE( A.ASCEND\_BYTES ,0) else 0 end) UP\_ROAM\_GAT\_FLUX ,

sum(case when A.ROAM\_TYPE='KH0060204' then COALESCE( A.ASCEND\_BYTES ,0) else 0 end) UP\_ROAM\_INT\_FLUX ,

sum(case when A.SERVICE\_TYPE='CP0010101' then COALESCE( A.ASCEND\_BYTES ,0) else 0 end) UP\_2G\_FLUX ,

sum(case when A.SERVICE\_TYPE='CP0010102' then COALESCE( A.ASCEND\_BYTES ,0) else 0 end) UP\_3G\_FLUX ,

sum( COALESCE( A.DOWN\_BYTES ,0)) DOWN\_FLUX ,

sum(case when A.ROAM\_TYPE='KH00601' then COALESCE( A.DOWN\_BYTES ,0) else 0 end) DOWN\_LOCAL\_FLUX ,

sum(case when A.ROAM\_TYPE='KH0060201' then COALESCE( A.DOWN\_BYTES ,0) else 0 end) DOWN\_ROAM\_PROV\_FLUX ,

sum(case when A.ROAM\_TYPE='KH0060202' then COALESCE( A.DOWN\_BYTES ,0) else 0 end) DOWN\_ROAM\_CONT\_FLUX ,

sum(case when A.ROAM\_TYPE='KH0060203' then COALESCE( A.DOWN\_BYTES ,0) else 0 end) DOWN\_ROAM\_GAT\_FLUX ,

sum(case when A.ROAM\_TYPE='KH0060204' then COALESCE( A.DOWN\_BYTES ,0) else 0 end) DOWN\_ROAM\_INT\_FLUX ,

sum(case when A.SERVICE\_TYPE='CP0010101' then COALESCE( A.DOWN\_BYTES ,0) else 0 end) DOWN\_2G\_FLUX ,

sum(case when A.SERVICE\_TYPE='CP0010102' then COALESCE( A.DOWN\_BYTES ,0) else 0 end) DOWN\_3G\_FLUX ,

sum(COALESCE( A.USE\_DURATION ,0)) TOTAL\_DURA ,

sum(case when A.COMM\_FEE+A.INFO\_FEE+A.OTHER\_FEE=0 then COALESCE( A.USE\_DURATION ,0) else 0 end) FREE\_DURA ,

sum(case when A.COMM\_FEE+A.INFO\_FEE+A.OTHER\_FEE<>0 then COALESCE( A.USE\_DURATION ,0) else 0 end) BILL\_DURA ,

sum(case when A.SERVICE\_TYPE='CP0010101' then COALESCE( A.USE\_DURATION ,0) else 0 end) FLUX\_2G\_DURA ,

sum(case when A.SERVICE\_TYPE='CP0010102' then COALESCE( A.USE\_DURATION ,0) else 0 end) FLUX\_3G\_DURA ,

sum(COALESCE( A.COMM\_FEE+A.INFO\_FEE+A.OTHER\_FEE ,0)) TOTAL\_FEE ,

sum(case when A.ROAM\_TYPE='KH00601' then COALESCE( A.DINNER\_IN\_BYTES ,0) else 0 end) PROD\_IN\_LOCAL\_FLUX ,

sum(case when A.ROAM\_TYPE='KH00601' then COALESCE( A.DINNER\_IN\_BYTES ,0) else 0 end) PROD\_IN\_ROAM\_CONT\_FLUX ,

sum(case when A.ROAM\_TYPE='KH00601' then COALESCE( A.DINNER\_IN\_BYTES ,0) else 0 end) PROD\_IN\_ROAM\_GAT\_FLUX ,

sum(case when A.ROAM\_TYPE='KH00601' then COALESCE( A.DINNER\_IN\_BYTES ,0) else 0 end) PROD\_IN\_ROAM\_INT\_FLUX

from

(select \* from DWD\_D\_USE\_MB\_GPRS A where a.month ='#CURDATEYYYYMM#' and pro=‘#PROVINCE#’) A

group by

A.MONTH\_ID,

A.PROV\_ID,

A.SUBS\_INSTANCE\_ID,

A.DEVICE\_NUMBER,

A.SERVICE\_TYPE

## [用户短信表(月)](file:///C:\Users\zhulf\AppData\Roaming\Microsoft\DWD、DWA测试模型_汇总_20130514.xlsx#'用户短信表(月)'!A1)DWA\_V\_M\_CUS\_SING\_SMS

### 建表语句

create table IF NOT EXISTS DWA\_V\_M\_CUS\_SING\_SMS

(MONTH\_ID string,

PROV\_ID string,

SUBS\_INSTANCE\_ID string,

DEVICE\_NUMBER string,

SERVICE\_TYPE string,

TOTAL\_SMS\_NUM INT,

PTP\_SMS\_NUM INT,

SP\_SMS\_NUM INT,

GROUP\_SMS\_NUM INT,

OTH\_SMS\_NUM INT,

OUT\_SMS\_NUM INT,

IN\_SMS\_NUM INT,

FREE\_SMS\_NUM INT,

BILL\_SMS\_NUM INT,

PTP\_OUT\_FREE\_NUM INT,

PTP\_OUT\_BILL\_NUM INT,

PTP\_IN\_FREE\_NUM INT,

PTP\_IN\_BILL\_NUM INT,

SP\_OUT\_FREE\_NUM INT,

SP\_OUT\_BILL\_NUM INT,

SP\_IN\_FREE\_NUM INT,

SP\_IN\_BILL\_NUM INT,

PTP\_OUT\_CMCC\_NUM INT,

PTP\_OUT\_UNI\_NUM INT,

PTP\_OUT\_CTC\_NUM INT,

PTT\_OUT\_INT\_NUM INT,

PTT\_OUT\_OTH\_NUM INT,

PROD\_IN\_SMS\_NUM INT,

PROD\_PTP\_IN\_SMS\_NUM INT,

PROD\_PTP\_CONT\_IN\_SMS\_NUM INT,

PROD\_PTPIN\_CONT\_IN\_SMS\_NUM INT,

PROD\_PTPOUT\_CONT\_IN\_SMS\_NUM INT,

PROD\_PTP\_INT\_SMS\_NUM INT,

PROD\_PTP\_OTH\_SMS\_NUM INT,

PTP\_SMS\_FEE INT,

SP\_SMS\_FEE INT,

TOTAL\_SMS\_FEE INT

)partitioned by (month string,pro string)

row format delimited fields terminated by '\001'

### 加工数据

INSERT OVERWRITE TABLE DWA\_V\_M\_CUS\_SING\_SMS PARTITION (month ='#CURDATEYYYYMM#',pro='#PROVINCE#')

SELECT

MONTH\_ID,

PROV\_ID,

SUBS\_INSTANCE\_ID,

DEVICE\_NUMBER,

SERVICE\_TYPE,

count(\*) TOTAL\_SMS\_NUM,

count(CASE WHEN SMS\_TYPE='1' THEN '1' ELSE NULL END) PTP\_SMS\_NUM,

count(CASE WHEN SMS\_TYPE='2' THEN '1' ELSE NULL END) SP\_SMS\_NUM,

count(CASE WHEN SMS\_TYPE='3' THEN '1' ELSE NULL END) GROUP\_SMS\_NUM,

count(CASE WHEN SMS\_TYPE='0' THEN '1' ELSE NULL END) OTH\_SMS\_NUM,

count(\*) OUT\_SMS\_NUM,

0 IN\_SMS\_NUM,

count(CASE WHEN COMM\_FEE+INFO\_FEE+RENT\_FEE+OTHER\_FEE=0 THEN '1' ELSE NULL END) FREE\_SMS\_NUM,

count(CASE WHEN COMM\_FEE+INFO\_FEE+RENT\_FEE+OTHER\_FEE>0 THEN '1' ELSE NULL END) BILL\_SMS\_NUM,

count(CASE WHEN IS\_FREE='0' and SMS\_TYPE='1' THEN '1' ELSE NULL END) PTP\_OUT\_FREE\_NUM,

count(CASE WHEN IS\_FREE='1' and SMS\_TYPE='1' THEN '1' ELSE NULL END) PTP\_OUT\_BILL\_NUM,

0 PTP\_IN\_FREE\_NUM,

0 PTP\_IN\_BILL\_NUM,

count(CASE WHEN COMM\_FEE+INFO\_FEE+RENT\_FEE+OTHER\_FEE=0 and SMS\_TYPE='2' THEN '1' ELSE NULL END) SP\_OUT\_FREE\_NUM,

count(CASE WHEN COMM\_FEE+INFO\_FEE+RENT\_FEE+OTHER\_FEE>0 and SMS\_TYPE='2' THEN '1' ELSE NULL END) SP\_OUT\_BILL\_NUM,

0 SP\_IN\_FREE\_NUM,

0 SP\_IN\_BILL\_NUM,

count(CASE WHEN OPPOSE\_OPERATOR='HZ0080302' and SMS\_TYPE='1' THEN '1' ELSE NULL END) PTP\_OUT\_CMCC\_NUM,

count(CASE WHEN OPPOSE\_OPERATOR='HZ0080301' and SMS\_TYPE='1' THEN '1' ELSE NULL END) PTP\_OUT\_UNI\_NUM,

count(CASE WHEN OPPOSE\_OPERATOR='HZ0080303' and SMS\_TYPE='1' THEN '1' ELSE NULL END) PTP\_OUT\_CTC\_NUM,

count(CASE WHEN OPPOSE\_OPERATOR='HZ00801' and SMS\_TYPE='1' THEN '1' ELSE NULL END) PTT\_OUT\_INT\_NUM,

count(CASE WHEN OPPOSE\_OPERATOR='HZ00899' and SMS\_TYPE='1' THEN '1' ELSE NULL END) PTT\_OUT\_OTH\_NUM,

count(CASE WHEN COMM\_FEE=0 and INFO\_FEE=0 and RENT\_FEE=0 and OTHER\_FEE =0 THEN '1' ELSE NULL END) PROD\_IN\_SMS\_NUM,

count(CASE WHEN COMM\_FEE=0 and INFO\_FEE=0 and RENT\_FEE=0 and OTHER\_FEE =0 and SMS\_TYPE='1' THEN '1' ELSE NULL END) PROD\_PTP\_IN\_SMS\_NUM,

count(CASE WHEN COMM\_FEE=0 and INFO\_FEE=0 and RENT\_FEE=0 and OTHER\_FEE =0 and SMS\_TYPE='1' THEN '1' ELSE NULL END) PROD\_PTP\_CONT\_IN\_SMS\_NUM,

count(CASE WHEN COMM\_FEE=0 and INFO\_FEE=0 and RENT\_FEE=0 and OTHER\_FEE =0 and SMS\_TYPE='1' and OPPOSE\_OPERATOR='1' THEN '1' ELSE NULL END) PROD\_PTPIN\_CONT\_IN\_SMS\_NUM,

count(CASE WHEN COMM\_FEE=0 and INFO\_FEE=0 and RENT\_FEE=0 and OTHER\_FEE =0 and SMS\_TYPE='1' and OPPOSE\_OPERATOR<>'1' THEN '1' ELSE NULL END) PROD\_PTPOUT\_CONT\_IN\_SMS\_NUM,

0 PROD\_PTP\_INT\_SMS\_NUM,

0 PROD\_PTP\_OTH\_SMS\_NUM,

SUM(CASE WHEN SMS\_TYPE='1' THEN COMM\_FEE+INFO\_FEE+RENT\_FEE+OTHER\_FEE ELSE 0 END) PTP\_SMS\_FEE,

SUM(CASE WHEN SMS\_TYPE='2' THEN COMM\_FEE+INFO\_FEE+RENT\_FEE+OTHER\_FEE ELSE 0 END) SP\_SMS\_FEE,

SUM(COMM\_FEE+INFO\_FEE+RENT\_FEE+OTHER\_FEE) TOTAL\_SMS\_FEE

FROM DWD\_D\_USE\_MB\_SMS

WHERE month ='#CURDATEYYYYMM#'

And pro=‘#PROVINCE#’

GROUP BY MONTH\_ID,

PROV\_ID,

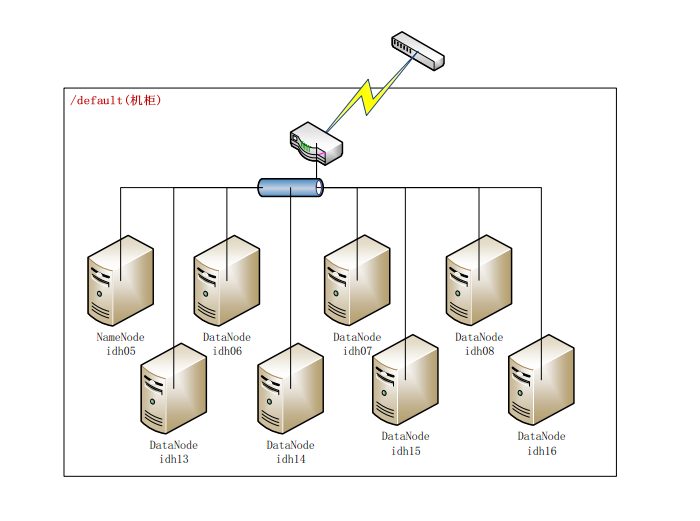
SUBS\_INSTANCE\_ID,

SERVICE\_TYPE,

DEVICE\_NUMBER

# 数据主仓的内部节点的结构

## 数据集群的拓扑结构



## 数据集群中的各节点分工

10.1.24.108 primary nameNode,jobTracker,GangliaServer,Management

10.1.24.109 DataNode,TaskTracker,HBaseMaster,HBaseRegionServer,ZooKeeper,Hive Thrift

10.1.24.110 DataNode,TaskTracker,HBaseMaster,HBaseRegionServer,ZooKeeper

10.1.24.111 DataNode,TaskTracker,HBaseMaster,HBaseRegionServer,ZooKeeper

10.1.24.112 DataNode,TaskTracker,HBaseRegionServer,Oozie Server

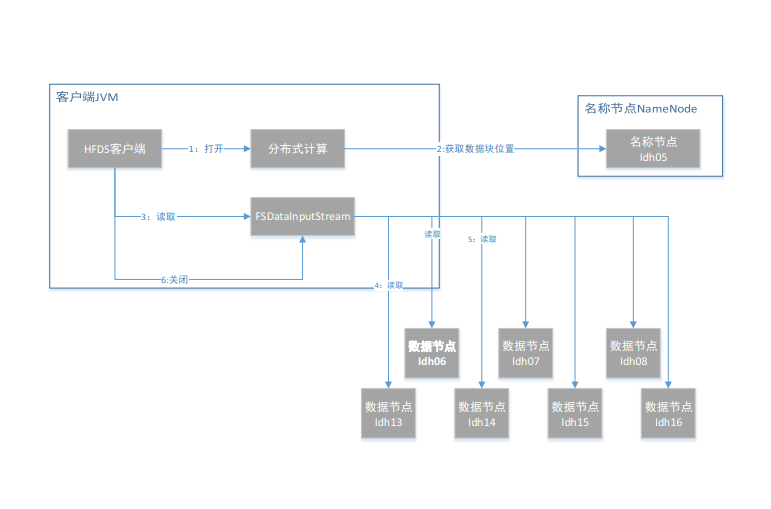
10.1.24.113 DataNode,TaskTracker,HBaseRegionServer

10.1.24.114 DataNode,TaskTracker,HBaseRegionServer

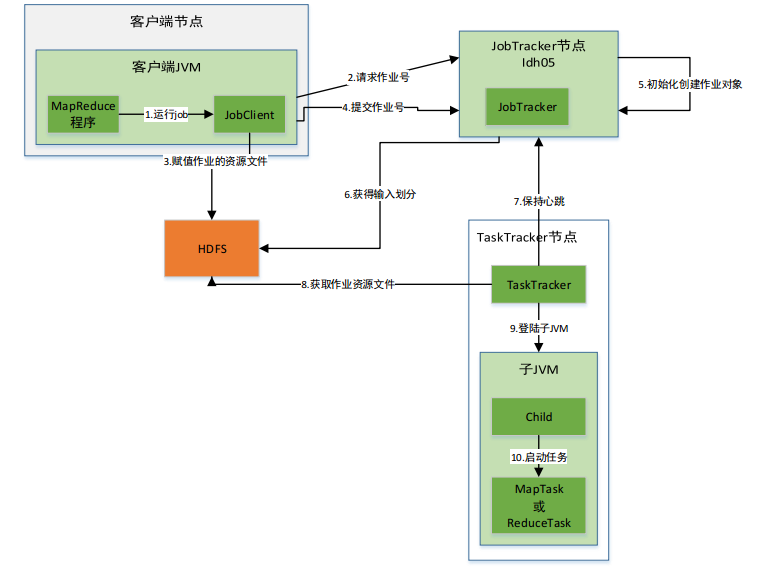
10.1.24.115 DataNode,TaskTracker,HBaseRegionServer

## 数据在集群中的流动情况，即数据加载加工过程中数据的流转

HDFS数据流



MapReduce流程图



# 附录1：《POC ODS 结构化数据\_实体表结构.xlsx》

