# 1、数据源表结构及源文件加载到Hive数据仓库脚本

## 1.1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”;

流量数据：以”|”为分隔符的txt压缩文件。

为了方便描述，本文以语音数据代表ODS结构化数据，以流量数据代表O域流量数据。

## 1.2语音数据加载到数据仓库

#### 1.2.1语音数据建表语句

create external table IF NOT EXISTS TG\_CDRMM\_SD\_GC

(

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' stored as textfile;

#### 1.2.2语音数据加载语句

load data local inpath "/mnt/disk1/10.1.24.233/sd\_cdrmm/sd\_cdrmm\_19-31/17\_tg\_cdrmm\_201307" into table TG\_CDRMM\_SD\_GC;

load data local inpath "/mnt/disk1/10.1.24.233/sd\_cdrmm1-18" into table TG\_CDRMM\_SD\_GC;

## 1.3 流量数据加载到数据仓库

#### 1.3.1流量数据建表语句

create external table IF NOT EXISTS gn\_url\_sd\_gc\_partition

(

user\_id string ,

start\_time string,

time string,

lac string,

ci\_num string,

imei string,

gprs\_type string,

end\_time string,

up\_count bigint,

down\_count bigint,

net\_type string,

status string,

net\_status string,

imsi string,

url string,

record\_flag string,

merge\_records string,

prov\_id string,

city\_id string,

ip string,

visite\_ip string,

sgsn\_ip string,

ggsn\_ip string,

source\_port string,

destination\_port string,

user\_agent string,

content\_type string

)partitioned by(segment String) row format delimited fields terminated by '|' stored as textfile;

#### 1.3.2流量数据加载语句

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc/130\*" into table gn\_url\_sd\_gc\_partition partition (segment = "130")；

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc/131\*" into table gn\_url\_sd\_gc\_partition partition (segment = "131")；

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc/132\*" into table gn\_url\_sd\_gc\_partition partition (segment = "132")；

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc/145\*" into table gn\_url\_sd\_gc\_partition partition (segment = "145")；

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc/155\*" into table gn\_url\_sd\_gc\_partition partition (segment = "155")；

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc/156\*" into table gn\_url\_sd\_gc\_partition partition (segment = "156")；

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc/185\*" into table gn\_url\_sd\_gc\_partition partition (segment = "185")；

由于‘186’开头数据较大，所以进行单独处理，‘186’开头数据导入HIVE表：

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc\_partition/segment=186/1860\*" into table gn\_url\_sd\_gc\_partition partition (segment = "1860")

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc\_partition/segment=186/1861\*" into table gn\_url\_sd\_gc\_partition partition (segment = "1861")

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc\_partition/segment=186/1865\*" into table gn\_url\_sd\_gc\_partition partition (segment = "1865")

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc\_partition/segment=186/1866\*" into table gn\_url\_sd\_gc\_partition partition (segment = "1866")

load data inpath "/user/hive/warehouse/gn\_url\_sd\_gc\_partition/segment=186/1867\*" into table gn\_url\_sd\_gc\_partition partition (segment = "1867")

# 2、山东三地市基站信息数据加载到Hive数据仓库

### 2.1基站信息数据建表语句

**济宁：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3G 基站信息表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create external table IF NOT EXISTS nodeb\_3g\_jining\_sd(

node\_NAME STRING ,

CELL\_NAME STRING ,

CELL\_ID STRING ,

node\_Longitude float ,

node\_Latitude float ,

node\_AREA float ,

node\_Radius float ,

LAC STRING ,

node\_ID STRING

)row format delimited fields terminated by ',' stored as textfile;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2G 基站信息表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create external table IF NOT EXISTS nodeb\_2g\_jining\_sd(

node\_NAME STRING ,

CELL\_NAME STRING ,

CELL\_ID STRING ,

node\_Longitude float ,

node\_Latitude float ,

node\_AREA float ,

node\_Radius float ,

LAC STRING ,

node\_ID STRING

)row format delimited fields terminated by ',' stored as textfile;

**枣庄：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3G 基站信息表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create external table IF NOT EXISTS nodeb\_3g\_zaozhuang\_sd(

node\_NAME STRING ,

CELL\_NAME STRING ,

CELL\_ID STRING ,

node\_Longitude float ,

node\_Latitude float ,

node\_AREA float ,

node\_Radius float ,

LAC STRING ,

node\_ID STRING

)row format delimited fields terminated by ',' stored as textfile;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2G 基站信息表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create external table IF NOT EXISTS nodeb\_2g\_zaozhuang\_sd(

node\_NAME STRING ,

CELL\_NAME STRING ,

CELL\_ID STRING ,

node\_Longitude float ,

node\_Latitude float ,

node\_AREA float ,

node\_Radius float ,

LAC STRING ,

node\_ID STRING

)row format delimited fields terminated by ',' stored as textfile;

**泰安：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3G 基站信息表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create external table IF NOT EXISTS nodeb\_3g\_taian\_sd(

node\_NAME STRING ,

CELL\_NAME STRING ,

CELL\_ID STRING ,

node\_Longitude float ,

node\_Latitude float ,

node\_AREA float ,

node\_Radius float ,

LAC STRING ,

node\_ID STRING

)row format delimited fields terminated by ',' stored as textfile;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2G 基站信息表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create external table IF NOT EXISTS nodeb\_2g\_taian\_sd(

node\_NAME STRING ,

CELL\_NAME STRING ,

CELL\_ID STRING ,

node\_Longitude float ,

node\_Latitude float ,

node\_AREA float ,

node\_Radius float ,

LAC STRING ,

node\_ID STRING

)row format delimited fields terminated by ',' stored as textfile;

### 2.2基站信息数据加载语句

**济宁：**

load data local inpath '/mnt/disk1/visitors\_flow/NodeB\_JN/nodeb\_3g\_jining\_sd.csv' into table nodeb\_3g\_jining\_sd;

load data local inpath '/mnt/disk1/visitors\_flow/NodeB\_JN/nodeb\_2g\_jining\_sd.csv' into table nodeb\_2g\_jining\_sd;

**枣庄：**

load data local inpath '/mnt/disk1/visitors\_flow/NodeB\_ZZ/nodeb\_3g\_zaozhuang\_sd.csv' into table nodeb\_3g\_zaozhuang\_sd;

load data local inpath '/mnt/disk1/visitors\_flow/NodeB\_ZZ/nodeb\_2g\_zaozhuang\_sd.csv' into table nodeb\_2g\_zaozhuang\_sd;

**泰安：**

load data local inpath '/mnt/disk1/visitors\_flow/NodeB\_TA/nodeb\_3g\_taian\_sd.csv' into table nodeb\_3g\_taian\_sd;

load data local inpath '/mnt/disk1/visitors\_flow/NodeB\_TA/nodeb\_2g\_taian\_sd.csv' into table nodeb\_2g\_taian\_sd;

# 3、人口热力图加工脚本

## 3.1数据源表部分字段抽取

#### 3.1.1语音建表语句

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*原始语音数据抽取部分字段\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create external table IF NOT EXISTS TG\_CDRMM\_EXTRACT\_SD\_GC

(

START\_DATE STRING ,

START\_TIME STRING ,

END\_DATE STRING ,

END\_TIME STRING ,

IMSI\_NUMBER STRING ,

MSISDN STRING ,

LAC1 STRING ,

CELL\_ID1 STRING ,

LAC2 STRING ,

CELL\_ID2 STRING ,

OTHER\_PARTY STRING ,

OTHER\_LAC1 STRING ,

OTHER\_CELL\_ID1 STRING ,

OTHER\_LAC2 STRING

)

partitioned by(ptDate String) row format delimited fields terminated by '\001' stored as RCFile;

#### 3.1.2语音加工语句

from TG\_CDRMM\_SD\_GC t

INSERT OVERWRITE TABLE TG\_CDRMM\_EXTRACT\_SD\_GC partition (ptDate='201307')

select t.START\_DATE,

t.START\_TIME,

t.END\_DATE,

t.END\_TIME,

t.IMSI\_NUMBER,

t.MSISDN,

t.LAC1,

t.CELL\_ID1,

t.LAC2,

t.CELL\_ID2,

t.OTHER\_PARTY,

t.OTHER\_LAC1,

t.OTHER\_CELL\_ID1,

t.OTHER\_LAC2;

#### 3.1.3 流量建表语句

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*原始流量数据抽取部分字段\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create external table IF NOT EXISTS GN\_URL\_extract\_SD\_GC\_partition(

START\_TIME string,

END\_TIME string,

MSISDN string,

IMSI string,

CI\_NUM string,

LAC string,

NET\_TYPE string,

UP\_COUNT bigint,

DOWN\_COUNT bigint,

VISIT\_IP string,

URL string

)partitioned by(segment String,ptDate string) row format delimited fields terminated by '|' stored as RCFile;

#### 3.1.4 流量加工语句

流量数据以130~185开头的数据：

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "130",ptDate="201307") select t.START\_TIME,t.END\_TIME,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "130" ;

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "131",ptDate="201307") select t.START\_TIME,t.END\_TIME,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "131" ;

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "132",ptDate="201307") select t.START\_TIME,t.END\_TIME,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "132" ;

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "145",ptDate="201307") select t.START\_TIME,t.END\_TIME,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "145" ;

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "155",ptDate="201307") select t.START\_TIME,t.END\_TIME,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "155" ;

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "156",ptDate="201307") select t.START\_TIME,t.END\_TIME,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "156" ;

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "185",ptDate="201307") select t.START\_TIME,t.END\_TIME,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "185" ;

由于‘186’开头数据较大，进行单独处理:

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "186",ptDate="201307") select t.START\_TIME,t.END\_TIME

,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "1860";

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "186",ptDate="201307") select t.START\_TIME,t.END\_TIME

,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "1861";

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "186",ptDate="201307") select t.START\_TIME,t.END\_TIME

,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "1865";

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "186",ptDate="201307") select t.START\_TIME,t.END\_TIME

,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "1866";

from gn\_url\_sd\_gc\_partition t INSERT OVERWRITE TABLE GN\_URL\_extract\_SD\_GC\_partition partition( segment = "186",ptDate="201307") select t.START\_TIME,t.END\_TIME

,t.user\_id,t.IMSI,t.CI\_NUM,t.LAC,t.NET\_TYPE,t.UP\_COUNT,t.DOWN\_COUNT,t.visite\_ip,t.URL where t.segment = "1867";

## 3.2人口热力图字段抽取

#### 3.2.1建表语句

create table HEATMAP\_SD\_CRDMM\_AND\_URL\_detail

(

START\_TIME STRING,

END\_TIME STRING,

MSISDN STRING,

IMSI STRING,

LAC STRING,

CELL\_ID STRING

)partitioned by(ptType String) row format delimited fields terminated by ',' stored as RCFile;

#### 3.2.2语音数据加工语句

from TG\_CDRMM\_EXTRACT\_SD\_GC t

INSERT OVERWRITE TABLE HEATMAP\_SD\_CRDMM\_AND\_URL\_detail partition(ptType='CDRMM201307')

select

concat(t.START\_DATE,t.START\_TIME),

concat(t.END\_DATE,t.END\_TIME),

t.MSISDN,

t.IMSI\_NUMBER,

t.LAC1,

t.CELL\_ID1;

#### 3.2.3 流量数据加工语句

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*‘130’开头数据处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from gn\_url\_extract\_sd\_gc\_partition t

INSERT INTO TABLE HEATMAP\_SD\_CRDMM\_AND\_URL\_detail partition( ptType='URL201307')

select

concat("201307",t.START\_TIME),

concat("201307",t.END\_TIME),

t.MSISDN,

t.IMSI,

t.LAC,

t.CI\_NUM

where t.segment='130' and t.ptdate='201307'

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*‘131’开头数据处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from gn\_url\_extract\_sd\_gc\_partition t

INSERT INTO TABLE HEATMAP\_SD\_CRDMM\_AND\_URL\_detail partition( ptType='URL201307')

select

concat("201307",t.START\_TIME),

concat("201307",t.END\_TIME),

t.MSISDN,

t.IMSI,

t.LAC,

t.CI\_NUM

where t.segment='131' and t.ptdate='201307'

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*‘132’开头数据处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from gn\_url\_extract\_sd\_gc\_partition t

INSERT INTO TABLE HEATMAP\_SD\_CRDMM\_AND\_URL\_detail partition( ptType='URL201307')

select

concat("201307",t.START\_TIME),

concat("201307",t.END\_TIME),

t.MSISDN,

t.IMSI,

t.LAC,

t.CI\_NUM

where t.segment='132' and t.ptdate='201307'

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*‘145’开头数据处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from gn\_url\_extract\_sd\_gc\_partition t

INSERT INTO TABLE HEATMAP\_SD\_CRDMM\_AND\_URL\_detail partition( ptType='URL201307')

select

concat("201307",t.START\_TIME),

concat("201307",t.END\_TIME),

t.MSISDN,

t.IMSI,

t.LAC,

t.CI\_NUM

where t.segment='145' and t.ptdate='201307'

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*‘155’开头数据处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from gn\_url\_extract\_sd\_gc\_partition t

INSERT INTO TABLE HEATMAP\_SD\_CRDMM\_AND\_URL\_detail partition( ptType='URL201307')

select

concat("201307",t.START\_TIME),

concat("201307",t.END\_TIME),

t.MSISDN,

t.IMSI,

t.LAC,

t.CI\_NUM

where t.segment='155' and t.ptdate='201307'

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*‘156’开头数据处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from gn\_url\_extract\_sd\_gc\_partition t

INSERT INTO TABLE HEATMAP\_SD\_CRDMM\_AND\_URL\_detail partition( ptType='URL201307')

select

concat("201307",t.START\_TIME),

concat("201307",t.END\_TIME),

t.MSISDN,

t.IMSI,

t.LAC,

t.CI\_NUM

where t.segment='156' and t.ptdate='201307'

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*‘185’开头数据处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from gn\_url\_extract\_sd\_gc\_partition t

INSERT INTO TABLE HEATMAP\_SD\_CRDMM\_AND\_URL\_detail partition( ptType='URL201307')

select

concat("201307",t.START\_TIME),

concat("201307",t.END\_TIME),

t.MSISDN,

t.IMSI,

t.LAC,

t.CI\_NUM

where t.segment='185' and t.ptdate='201307'

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*‘186’开头数据处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from gn\_url\_extract\_sd\_gc\_partition t

INSERT INTO TABLE HEATMAP\_SD\_CRDMM\_AND\_URL\_detail partition( ptType='URL201307')

select

concat("201307",t.START\_TIME),

concat("201307",t.END\_TIME),

t.MSISDN,

t.IMSI,

t.LAC,

t.CI\_NUM

where t.segment='186' and t.ptdate='201307'

## 3.3人口热力图字段抽取数据与基站信息表关联

#### 3.2.1语音数据建表语句

**济宁：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*济宁市语音数据建表语句\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_JN\_CRDMM\_detail

(

START\_TIME STRING,

END\_TIME STRING,

MSISDN STRING,

IMSI STRiNG,

CELL\_NAME STRING,

CELL\_ID STRING,

node\_NAME STRING,

node\_ID STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

node\_area float

)partitioned by(ptType String) row format delimited fields terminated by ',' stored as RCFile;

**枣庄：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*枣庄市语音数据建表语句\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_ZZ\_CRDMM\_detail

(

START\_TIME STRING,

END\_TIME STRING,

MSISDN STRING,

IMSI STRiNG,

CELL\_NAME STRING,

CELL\_ID STRING,

node\_NAME STRING,

node\_ID STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

node\_area float

)partitioned by(ptType String) row format delimited fields terminated by ',' stored as RCFile;

**泰安：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*泰安市语音数据建表语句\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_TA\_CRDMM\_detail

(

START\_TIME STRING,

END\_TIME STRING,

MSISDN STRING,

IMSI STRiNG,

CELL\_NAME STRING,

CELL\_ID STRING,

node\_NAME STRING,

node\_ID STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

node\_area float

)partitioned by(ptType String) row format delimited fields terminated by ',' stored as RCFile;

#### 3.2.2语音数据加工语句

**济宁：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3G语音数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_3g\_jining\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType='CDRMM201307')

INSERT OVERWRITE TABLE HEATMAP\_SD\_JN\_CRDMM\_detail partition( ptType='CDRMM3G201307')

select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2G语音数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_2g\_jining\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType='CDRMM201307')

INSERT OVERWRITE TABLE HEATMAP\_SD\_JN\_CRDMM\_detail partition( ptType='CDRMM2G201307')

select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area;

**枣庄市：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3G语音数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_3g\_zaozhuang\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType='CDRMM201307')

INSERT OVERWRITE TABLE HEATMAP\_SD\_ZZ\_CRDMM\_detail partition( ptType='CDRMM3G201307')

select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2G语音数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_2g\_zaozhuang\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType='CDRMM201307')

INSERT OVERWRITE TABLE HEATMAP\_SD\_ZZ\_CRDMM\_detail partition( ptType='CDRMM2G201307')

select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area;

**泰安市：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3G语音数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_3g\_taian\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType='CDRMM201307')

INSERT OVERWRITE TABLE HEATMAP\_SD\_TA\_CRDMM\_detail partition( ptType='CDRMM3G201307')

select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2G语音数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_2g\_taian\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType='CDRMM201307')

INSERT OVERWRITE TABLE HEATMAP\_SD\_TA\_CRDMM\_detail partition( ptType='CDRMM2G201307')

select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area;

#### 3.2.3流量数据建表语句

**济宁：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*流量数据建表语句\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_JN\_GNURL\_detail

(

START\_TIME STRING,

END\_TIME STRING,

MSISDN STRING,

IMSI STRiNG,

CELL\_NAME STRING,

CELL\_ID STRING,

node\_NAME STRING,

node\_ID STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

node\_area float

)partitioned by(ptType String) row format delimited fields terminated by "," stored as RCFile

**枣庄：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*流量数据建表语句\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_ZZ\_GNURL\_detail(

START\_TIME STRING,

END\_TIME STRING,

MSISDN STRING,

IMSI STRiNG,

CELL\_NAME STRING,

CELL\_ID STRING,

node\_NAME STRING,

node\_ID STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

node\_area float

)partitioned by(ptType String) row format delimited fields terminated by "," stored as RCFile;

**泰安：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*流量数据建表语句\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_TA\_GNURL\_detail(

START\_TIME STRING,

END\_TIME STRING,

MSISDN STRING,

IMSI STRiNG,

CELL\_NAME STRING,

CELL\_ID STRING,

node\_NAME STRING,

node\_ID STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

node\_area float

)partitioned by(ptType String) row format delimited fields terminated by "," stored as RCFile;

#### 3.2.4流量数据加工语句

**济宁：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3G流量数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_3g\_jining\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType= "URL201307") INSERT INTO TABLE

HEATMAP\_SD\_JN\_GNURL\_detail partition( ptType= "URL3G201307")

select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitu

de,t.node\_Latitude,

t.node\_Radius,

t.node\_area;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2G流量数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_2g\_jining\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType= "URL201307") INSERT INTO TABLE H

EATMAP\_SD\_JN\_GNURL\_detail partition( ptType="URL2G201307") select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area ;

**枣庄：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3G流量数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_3g\_zaozhuang\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType= "URL201307") INSERT INTO TAB

LE HEATMAP\_SD\_ZZ\_GNURL\_detail partition( ptType= "URL3G201307") select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2G流量数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_2g\_zaozhuang\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType= "URL201307") INSERT INTO TABLE

HEATMAP\_SD\_ZZ\_GNURL\_detail partition( ptType="URL2G201307") select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area

**泰安：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3G流量数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_3g\_taian\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType= "URL201307") INSERT INTO TABLE HE

ATMAP\_SD\_TA\_GNURL\_detail partition( ptType= "URL3G201307") select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2G流量数据进表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from nodeb\_2g\_taian\_sd t inner join HEATMAP\_SD\_CRDMM\_AND\_URL\_detail s on (t.LAC=s.LAC AND t.CELL\_ID=s.CELL\_ID AND s.ptType= "URL201307") INSERT INTO TABLE HEA

TMAP\_SD\_TA\_GNURL\_detail partition( ptType="URL2G201307") select

s.START\_TIME,

s.END\_TIME,

s.MSISDN,

s.IMSI,

t.CELL\_NAME,

t.CELL\_ID,

t.node\_NAME,

t.node\_ID,

t.node\_Longitude,

t.node\_Latitude,

t.node\_Radius,

t.node\_area;

## 3.4人口热力图展示数据抽取

#### 3.4.1语音数据建表语句

**济宁：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计per10min热力图展示数据表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_JN\_CRDMM\_ORACLE\_TOTAL(

node\_ID STRING,

start\_time string,

node\_NAME STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

customers\_count int,

customers\_density float,

node\_area float,

city String

)partitioned by(ptType String) row format delimited fields terminated by "," stored as textFile;

**枣庄：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计夜间停留表热力图展示数据表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_ZZ\_CRDMM\_ORACLE\_TOTAL(

node\_ID STRING,

start\_time string,

node\_NAME STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

customers\_count int,

customers\_density float,

node\_area float,

city String

)partitioned by(ptType String) row format delimited fields terminated by "," stored as textFile;

**泰安：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计日间停留表热力图展示数据表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_TA\_CRDMM\_ORACLE\_TOTAL(

node\_ID STRING,

start\_time string,

node\_NAME STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

customers\_count int,

customers\_density float,

node\_area float,

city String

)partitioned by(ptType String) row format delimited fields terminated by "," stored as textFile;

#### 3.4.2语音数据加工语句

**济宁：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计per10min热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_JN\_CRDMM\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_JN\_CRDMM\_ORACLE\_TOTAL partition(ptType = "per10min")

select

t.node\_ID,

concat(substr(t.start\_time,1,11),"0"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"jining"

where substr(t.start\_time,1,8) >= "20130701" and substr(t.start\_time,1,8)<= "20130731" group by t.node\_ID,substr(t.start\_time,1,11)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计夜间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_JN\_CRDMM\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_JN\_CRDMM\_ORACLE\_TOTAL partition(ptType = "rest")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"0600"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"jining"

where substr(t.start\_time,9,2) >= "00" and substr(t.start\_time,9,2) <= "06" or substr(t.start\_time,9,2) = "23" group by t.node\_ID,substr(t.start\_time,1,8)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计日间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_JN\_CRDMM\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_JN\_CRDMM\_ORACLE\_TOTAL partition(ptType = "work")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"1700"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),max(t.node\_area),

"jining"

where substr(t.start\_time,9,2) >= "09" and substr(t.start\_time,9,2) <= "18" group by t.node\_ID,substr(t.start\_time,1,8)

**枣庄：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计per10min热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_ZZ\_CRDMM\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_ZZ\_CRDMM\_ORACLE\_TOTAL partition(ptType = "per10min")

select

t.node\_ID,

concat(substr(t.start\_time,1,11),"0"),

MAX(t.node\_NAME), MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"zaozhuang"

where substr(t.start\_time,1,8)>= "20130701" and substr(t.start\_time,1,8)<= "20130731" group by t.node\_ID,substr(t.start\_time,1,11)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计夜间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_ZZ\_CRDMM\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_ZZ\_CRDMM\_ORACLE\_TOTAL partition(ptType = "rest")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"0600"),

MAX(t.node\_NAME), MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"zaozhuang"

where substr(t.start\_time,9,2)>="00" and substr(t.start\_time,9,2)<="06" or substr(t.start\_time,9,2)="23" group by t.node\_ID,substr(t.start\_time,1,8)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计日间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_ZZ\_CRDMM\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_ZZ\_CRDMM\_ORACLE\_TOTAL partition(ptType = "work")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"1700"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"zaozhuang"

where substr(t.start\_time,9,2)>="09" and substr(t.start\_time,9,2)<="18" group by t.node\_ID,substr(t.start\_time,1,8)

**泰安：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计per10min热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_TA\_CRDMM\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_TA\_CRDMM\_ORACLE\_TOTAL partition(ptType = "per10min")

select

t.node\_ID,

concat(substr(t.start\_time,1,11),"0"),

MAX(t.node\_NAME), MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"taian"

where substr(t.start\_time,1,8)>= "20130701" and substr(t.start\_time,1,8)<= "20130731" group by t.node\_ID,substr(t.start\_time,1,11)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计夜间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_TA\_CRDMM\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_TA\_CRDMM\_ORACLE\_TOTAL partition(ptType = "rest")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"0600"),

MAX(t.node\_NAME), MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"taian"

where substr(t.start\_time,9,2)>="00" and substr(t.start\_time,9,2)<="06" or substr(t.start\_time,9,2)="23" group by t.node\_ID,substr(t.start\_time,1,8)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计日间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_TA\_CRDMM\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_TA\_CRDMM\_ORACLE\_TOTAL partition(ptType = "work")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"1700"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"taian"

where substr(t.start\_time,9,2)>="09" and substr(t.start\_time,9,2)<="18" group by t.node\_ID,substr(t.start\_time,1,8)

#### 3.4.3流量数据建表语句

**济宁：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计per10min热力图展示数据表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_JN\_GNURL\_ORACLE\_TOTAL(

node\_ID STRING,

start\_time string,

node\_NAME STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

customers\_count int,

customers\_density float,

node\_area float,

city String

)partitioned by(ptType String) row format delimited fields terminated by "," stored as textFile

**枣庄：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计夜间停留区热力图展示数据表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_ZZ\_GNURL\_ORACLE\_TOTAL(

node\_ID STRING,

start\_time string,

node\_NAME STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

customers\_count int,

customers\_density float,

node\_area float,

city String

)partitioned by(ptType String) row format delimited fields terminated by "," stored as textFile

**泰安：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计日间停留区热力图展示数据表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table HEATMAP\_SD\_TA\_GNURL\_ORACLE\_TOTAL(

node\_ID STRING,

start\_time string,

node\_NAME STRING,

node\_Longitude float,

node\_Latitude float,

node\_Radius float,

customers\_count int,

customers\_density float,

node\_area float,

city String

)partitioned by(ptType String) row format delimited fields terminated by "," stored as textFile

#### 3.4.4流量数据加工语句

**济宁：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计per10min热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_JN\_GNURL\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_JN\_GNURL\_ORACLE\_TOTAL partition(ptType = "per10min")

select

t.node\_ID,

concat(substr(t.start\_time,1,11),"0"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"jining"

where substr(t.start\_time,1,8) >= "20130701" and substr(t.start\_time,1,8)<= "20130731" group by t.node\_ID,substr(t.start\_time,1,11)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计夜间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_JN\_GNURL\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_JN\_GNURL\_ORACLE\_TOTAL partition(ptType = "rest")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"0600"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"jining"

where substr(t.start\_time,9,2) >= "00" and substr(t.start\_time,9,2) <= "06" or substr(t.start\_time,9,2) = "23" group by t.node\_ID,substr(t.start\_time,1,8)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计日间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_JN\_GNURL\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_JN\_GNURL\_ORACLE\_TOTAL partition(ptType = "work")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"1700"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"jining"

where substr(t.start\_time,9,2) >= "09" and substr(t.start\_time,9,2) <= "18" group by t.node\_ID,substr(t.start\_time,1,8)

**枣庄：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计per10min热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_ZZ\_GNURL\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_ZZ\_GNURL\_ORACLE\_TOTAL partition(ptType = "per10min")

select

t.node\_ID,

concat(substr(t.start\_time,1,11),"0"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"zaozhuang"

where substr(t.start\_time,1,8) >= "20130701" and substr(t.start\_time,1,8)<= "20130731" group by t.node\_ID,substr(t.start\_time,1,11)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计夜间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_ZZ\_GNURL\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_ZZ\_GNURL\_ORACLE\_TOTAL partition(ptType = "rest")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"0600"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"zaozhuang"

where substr(t.start\_time,9,2) >= "00" and substr(t.start\_time,9,2) <= "06" or substr(t.start\_time,9,2) = "23" group by t.node\_ID,substr(t.st

art\_time,1,8)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计日间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_ZZ\_GNURL\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_ZZ\_GNURL\_ORACLE\_TOTAL partition(ptType = "work")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"1700"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"zaozhuang"

where substr(t.start\_time,9,2) >= "09" and substr(t.start\_time,9,2) <= "18" group by t.node\_ID,substr(t.start\_time,1,8)

**泰安：**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计per10min热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_TA\_GNURL\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_TA\_GNURL\_ORACLE\_TOTAL partition(ptType = "per10min")

select

t.node\_ID,

concat(substr(t.start\_time,1,11),"0"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"taian"

where substr(t.start\_time,1,8) >= "20130701" and substr(t.start\_time,1,8)<= "20130731" group by t.node\_ID,substr(t.start\_time,1,11)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计夜间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_TA\_GNURL\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_TA\_GNURL\_ORACLE\_TOTAL partition(ptType = "rest")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"0600"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"taian"

where substr(t.start\_time,9,2) >= "00" and substr(t.start\_time,9,2) <= "06" or substr(t.start\_time,9,2) = "23" group by t.node\_ID,substr(t.start\_

time,1,8)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按基站统计日间停留区热力图展示数据\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

from HEATMAP\_SD\_TA\_GNURL\_detail t INSERT OVERWRITE TABLE HEATMAP\_SD\_TA\_GNURL\_ORACLE\_TOTAL partition(ptType = "work")

select

t.node\_ID,

concat(substr(t.start\_time,1,8),"1700"),

MAX(t.node\_NAME),

MAX(t.node\_Longitude),

MAX(t.node\_Latitude),

MAX(t.node\_Radius),

count(distinct t.MSISDN),

count(distinct t.MSISDN)/max(t.node\_area),

max(t.node\_area),

"taian"

where substr(t.start\_time,9,2) >= "09" and substr(t.start\_time,9,2) <= "18" group by t.node\_ID,substr(t.start\_time,1,8)

# 4、轨迹分析加工脚本

轨迹分析原始数据来源于人口热力图字段抽取数据与基站信息表关联（3.3节）加工所得数据。

|  |  |
| --- | --- |
| 地市 | 轨迹分析原始数据 |
| 济宁 | HEATMAP\_SD\_JN\_CRDMM\_detail、HEATMAP\_SD\_JN\_GNURL\_detail |
| 枣庄 | HEATMAP\_SD\_ZZ\_CRDMM\_detail、HEATMAP\_SD\_ZZ\_GNURL\_detail |
| 泰安 | HEATMAP\_SD\_TA\_CRDMM\_detail、HEATMAP\_SD\_TA\_GNURL\_detail |

下面的加工脚本以济宁市为例，其他地市脚本可参考济宁市。

## 4.1数据预处理

**1、去重建表语句**

create table PERSON\_TRACE\_TEXTFILE

(

DAY\_TIME STRING,

IMSI STRiNG,

MSISDN STRING,

START\_TIME STRING,

DURATION STRING,

node\_ID STRING,

node\_NAME STRING,

CELL\_NAME STRING,

CITY STRING,

node\_Longitude float,

node\_Latitude float,

node\_area float

)partitioned by(ptType String,ptloc String) row format delimited fields terminated by ',' stored as textfile;

**2、去重加工语句**

from HEATMAP\_SD\_JN\_CRDMM\_detail s INSERT OVERWRITE TABLE PERSON\_TRACE\_TEXTFILE partition( ptType = "crdmm",ptloc = "jining")

select distinct

substring(s.START\_TIME,0,8),

s.IMSI,s.MSISDN,s.START\_TIME,

abs(unix\_timestamp(concat(substring(s.START\_TIME , 0 , 8) , " " , substring(s.START\_TIME , 9 , 2) , ":" , substring(s.START\_TIME , 11 , 2) , ":" , substring(s.START\_TIME , 13 , 2)),"yyyyMMdd HH:mm:ss") - unix\_timestamp(concat(substring(s.END\_TIME , 0 , 8) , " " , substring(s.END\_TIME , 9 , 2) , ":" , substring(s.END\_TIME , 11 , 2) , ":" , substring(s.END\_TIME , 13 , 2)),"yyyyMMdd HH:mm:ss") ) ,

s.node\_ID,

s.node\_NAME,

s.CELL\_NAME,

"jining",

s.node\_Longitude,

s.node\_Latitude,

s.node\_area;

from HEATMAP\_SD\_JN\_GNURL\_detail s INSERT INTO TABLE PERSON\_TRACE\_TEXTFILE partition( ptType = "gnurl",ptloc = "jining")

select distinct

substring(s.START\_TIME,0,8),s.IMSI,s.MSISDN,s.START\_TIME, abs(unix\_timestamp(concat(substring(s.START\_TIME , 0 , 8) , " " , substring(s.START\_TIME , 9 , 2) , ":" , substring(s.START\_TIME , 11 , 2) , ":" , substring(s.START\_TIME , 13 , 2)),"yyyyMMdd HH:mm:ss") - unix\_timestamp(concat(substring(s.END\_TIME , 0 , 8) , " " , substring(s.END\_TIME , 9 , 2) , ":" , substring(s.END\_TIME , 11 , 2) , ":" , substring(s.END\_TIME , 13 , 2)),"yyyyMMdd HH:mm:ss") ) ,

s.node\_ID,

s.node\_NAME,

s.CELL\_NAME,

"jining",

s.node\_Longitude,

s.node\_Latitude,

s.node\_area

where s.pttype= "URL2G201307";

from HEATMAP\_SD\_JN\_GNURL\_detail s INSERT INTO TABLE PERSON\_TRACE\_TEXTFILE partition( ptType = "gnurl",ptloc = "jining")

select distinct

substring(s.START\_TIME,0,8),

s.IMSI,

s.MSISDN,

s.START\_TIME,

abs(unix\_timestamp(concat(substring(s.START\_TIME , 0 , 8) , " " , substring(s.START\_TIME , 9 , 2) , ":" , substring(s.START\_TIME , 11 , 2) , ":" , substring(s.START\_TIME , 13 , 2)),"yyyyMMdd HH:mm:ss") - unix\_timestamp(concat(substring(s.END\_TIME , 0 , 8) , " " , substring(s.END\_TIME , 9 , 2) , ":" , substring(s.END\_TIME , 11 , 2) , ":" , substring(s.END\_TIME , 13 , 2)),"yyyyMMdd HH:mm:ss") ) ,

s.node\_ID,

s.node\_NAME,

s.CELL\_NAME,

"jining",

s.node\_Longitude,

s.node\_Latitude,

s.node\_area

where s.pttype= "URL3G201307";

**3、取duration最大值建表语句**

create table PERSON\_TRACE\_TEXTFILE\_ORACLE

(

DAY\_TIME STRING,

IMSI STRiNG,

MSISDN STRING,

START\_TIME STRING,

DURATION STRING,

node\_ID STRING,

node\_NAME STRING,

CELL\_NAME STRING,

CITY STRING,

node\_Longitude float,

node\_Latitude float,

node\_area float

)partitioned by(ptType String,ptloc String) row format delimited fields terminated by ',' stored as textfile;

**4、取duration最大值加工语句**

from PERSON\_TRACE\_TEXTFILE s INSERT OVERWRITE TABLE PERSON\_TRACE\_TEXTFILE\_ORACLE partition( ptType = "crdmm",ptloc = "jining") select

s.DAY\_TIME,

s.IMSI,

s.MSISDN,

s.START\_TIME,

max(s.duration),

s.node\_ID,

s.node\_NAME,

s.CELL\_NAME,

s.city,

s.node\_Longitude,

s.node\_Latitude,

s.node\_area

where s.ptType = "crdmm" and s.ptloc = "jining" group by s.DAY\_TIME,s.IMSI,s.MSISDN,s.START\_TIME,s.node\_ID,s.node\_NAME,s.city,s.cell\_name,s.node\_Longitude,s.node\_Latitude,s.node\_area

from PERSON\_TRACE\_TEXTFILE s INSERT OVERWRITE TABLE PERSON\_TRACE\_TEXTFILE\_ORACLE partition( ptType = "gnurl",ptloc = "jining") select

s.DAY\_TIME,

s.IMSI,

s.MSISDN,

s.START\_TIME,

max(s.duration),

s.node\_ID,

s.node\_NAME,

s.CELL\_NAME,

s.city,

s.node\_Longitude,

s.node\_Latitude,

s.node\_area

where s.ptType = "crdmm" and s.ptloc = "jining" group by s.DAY\_TIME,s.IMSI,s.MSISDN,s.START\_TIME,s.node\_ID,s.node\_NAME,s.city,s.cell\_name,s.node\_Longitude,s.node\_Latitude,s.node\_area

## 4.2 MSISDN\_LIST列表

#### 4.2.1建表语句

create table MSISDN\_LIST

(

MSISDN STRING,

DAY\_TIME STRING,

CITY STRING

)partitioned by(ptloc String) row format delimited fields terminated by ',' stored as textfile;

#### 4.2.2加工脚本

from PERSON\_TRACE\_TEXTFILE\_ORACLE s INSERT OVERWRITE TABLE MSISDN\_LIST partition(ptloc = "SD")

select

s.MSISDN,

s.DAY\_TIME,

s.CITY

group by s.MSISDN,s.DAY\_TIME,s.CITY;

## 4.3 key-person列表(针对语音数据)

#### 4.3.1建表语句

create table KEY\_PERSON

(

IMSI STRING,

MSISDN STRING,

DAY\_TIME STRING,

BSUNIESS\_COUNT bigint,

BUSINESS\_DURATION double

)partitioned by(ptloc String) row format delimited fields terminated by ',' stored as textfile;

#### 4.3.2加工脚本

from PERSON\_TRACE\_TEXTFILE\_ORACLE s INSERT OVERWRITE TABLE KEY\_PERSON partition(ptloc = "SD")

select

s.IMSI,

s.MSISDN,

s.DAY\_TIME,

count(s.IMSI),

sum(s.DURATION)

where s.ptType = "crdmm" group by s.IMSI,s.MSISDN,s.DAY\_TIME;

## 4.4停留点

#### 4.4.1建表语句

create table person\_trace\_key\_point

(

DAY\_TIME STRING,

IMSI STRiNG,

MSISDN STRING,

START\_TIME STRING,

DURATION STRING,

node\_ID STRING,

node\_NAME STRING,

CELL\_NAME STRING,

CITY STRING,

node\_Longitude float,

node\_Latitude float,

node\_area float

) row format delimited fields terminated by ',' stored as textfile;

#### 4.4.2加工语句

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*标序号\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

create table info\_number as

select DAY\_TIME,

IMSI,

MSISDN,

START\_TIME,

DURATION,

node\_ID,

node\_NAME,

CELL\_NAME,

CITY,

node\_Longitude ,

node\_Latitude,

node\_area ,

row\_number() over ( distribute by msisdn sort by msisdn,START\_TIME asc ) id

from person\_trace\_hive12 s where s.ptloc = "jining" ;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*统计person\_trace\_key\_point \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

insert overwrite table person\_trace\_key\_point

select

i1.day\_time,

i1.IMSI,

i1.msisdn,

i1.START\_TIME,

i1.DURATION,

i1.node\_ID,

i1.node\_NAME,

i1.CELL\_NAME,

i1.CITY,

i1.node\_Longitude ,

i1.node\_Latitude,

i1.node\_area

from info\_number i1 left join info\_number i2 on (i1.id=i2.id-1 and i1.msisdn = i2.msisdn ) where (i2.id is null or i1.node\_id<>i2.node\_id) ;

insert into table person\_trace\_key\_point

select

i1.day\_time,

i1.IMSI,

i1.msisdn ,

i1.START\_TIME,

i1.DURATION,

i1.node\_ID,

i1.node\_NAME,

i1.CELL\_NAME,

i1.CITY,

i1.node\_Longitude ,

i1.node\_Latitude,

i1.node\_area

from info\_number i1 left join info\_number i2 on (i1.id=i2.id+1 and i1.msisdn = i2.msisdn) where (i2.id is null or i1.node\_id<>i2.node\_id) ;

## 4.5停留点路径切换

#### 4.5.1建表语句

create table person\_trace\_info\_road

(

IMSI STRiNG,

MSISDN STRING,

DAY\_TIME STRING,

i1\_id bigint,

i2\_id bigint,

i1\_node\_id STRING,

i2\_node\_id STRING,

i1\_START\_TIME STRING,

i2\_START\_TIME STRING,

i1\_node\_NAME STRING,

i2\_node\_NAME STRING,

i1\_node\_Longitude float,

i2\_node\_Longitude float,

i1\_node\_Latitude float,

i2\_node\_Latitude float,

i1\_node\_area float,

i2\_node\_area float

) row format delimited fields terminated by ',' stored as textfile;

#### 4.5.2加工语句

Insert overwrite person\_trace\_info\_road table

select

i1.IMSI,

i1.msisdn,

i1.day\_time ,

i1.id as i1\_id,

i2.id as i2\_id,

i1.node\_id as i1\_node\_id,

i2.node\_id as i2\_node\_id,

i1.start\_time as i1\_start\_time,

i2.start\_time as i2\_start\_time,

i1.node\_name as i1\_node\_name,

i2.node\_name as i2\_node\_name,

i1.node\_longitude as i1\_node\_longitude,

i2.node\_longitude as i2\_node\_longitude,

i1.node\_latitude as i1\_node\_latitude,

i2.node\_latitude as i2\_node\_latitude,

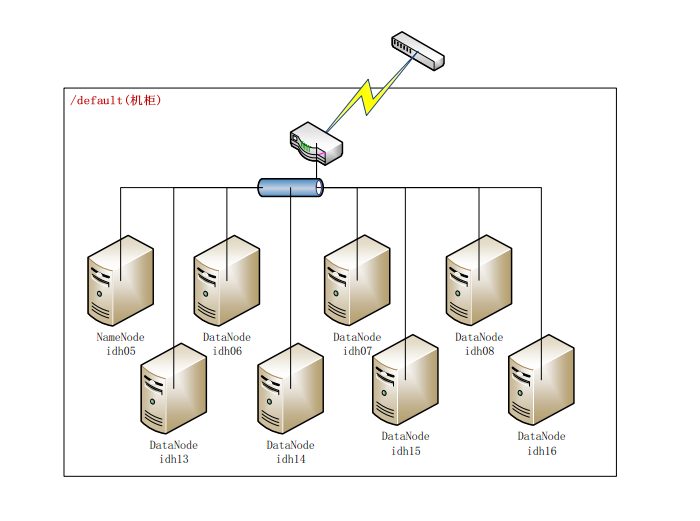
i1.node\_area as i1\_node\_area,

i2.node\_area as i2\_node\_area

from info\_number i1 left join info\_number i2 on （i1.id=i2.id-1 and i1.msisdn = i2.msisdn）where （i2.id is not null and i1.node\_id<>i2.node\_id）order by i1\_id;

# 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

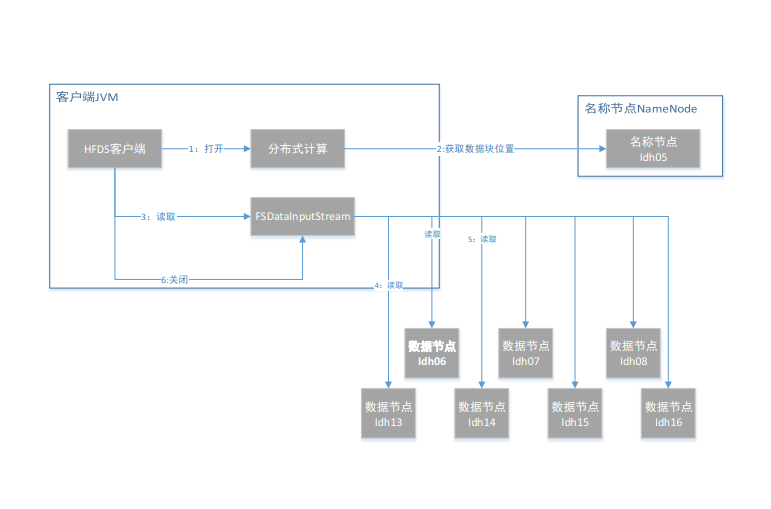
10.1.24.113 DataNode,TaskTracker,HBaseRegionServer

10.1.24.114 DataNode,TaskTracker,HBaseRegionServer

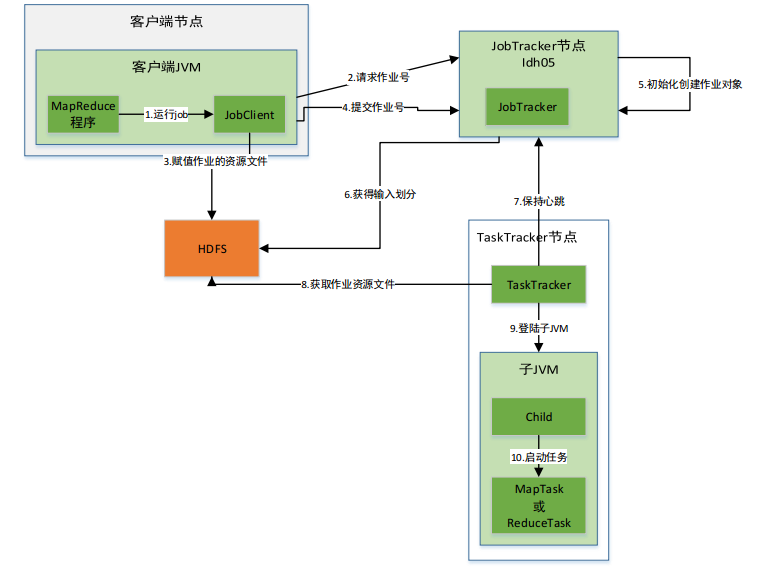
10.1.24.115 DataNode,TaskTracker,HBaseRegionServer

## 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:



# 数据仓库数据导入数据集市

## 6.1数据仓库表流向关系

数据仓库表流向关系如下图所示:



## 6.2数据仓库与数据集市主要表对应关系

|  |  |  |
| --- | --- | --- |
| 数据集市 | 数据仓库 | |
| 表名 | 表名 | 分区名 |
| HEATMAP\_SD\_CRDMM\_PER10MIN | heatmap\_sd\_jn\_crdmm\_oracle\_tota  heatmap\_sd\_zz\_crdmm\_oracle\_total  heatmap\_sd\_ta\_crdmm\_oracle\_total | pttype=per10min |
| HEATMAP\_SD\_CRDMM\_REST | heatmap\_sd\_jn\_crdmm\_oracle\_total  heatmap\_sd\_zz\_crdmm\_oracle\_total  heatmap\_sd\_ta\_crdmm\_oracle\_total | pttype=rest |
| HEATMAP\_SD\_CRDMM\_WORK | heatmap\_sd\_jn\_crdmm\_oracle\_total  heatmap\_sd\_zz\_crdmm\_oracle\_total  heatmap\_sd\_ta\_crdmm\_oracle\_total | pttype=work |
| HEATMAP\_SD\_GNURL\_PER10MIN | heatmap\_sd\_jn\_ gnurl\_oracle\_tota  heatmap\_sd\_zz\_ gnurl \_oracle\_total  heatmap\_sd\_ta\_ gnurl \_oracle\_total | pttype=per10min |
| HEATMAP\_SD\_GNURL\_REST | heatmap\_sd\_jn\_ gnurl \_oracle\_total  heatmap\_sd\_zz\_ gnurl \_oracle\_total  heatmap\_sd\_ta\_ gnurl \_oracle\_total | pttype=rest |
| HEATMAP\_SD\_GNURL\_WORK | heatmap\_sd\_jn\_ gnurl \_oracle\_total  heatmap\_sd\_zz\_ gnurl \_oracle\_total  heatmap\_sd\_ta\_ gnurl \_oracle\_total | pttype=work |
| PHONE\_LIST | msisdn\_list | ptloc=SD |
| KEY\_PERSON\_JN | key\_person\_jn | ptloc=SD |
| PERSON\_TRACE\_KEY\_POINT | person\_trace\_key\_point\_oracle | 无 |
| PERSON\_TRACE\_INFO\_ROAD | person\_trace\_info\_road\_oracle | 无 |

## 6.3 数据仓库导入数据集市sqoop对应语句

sqoop export --connect jdbc:oracle:thin:@10.1.24.211:1521:cupoc --table 数据仓库表名 --username HEATMAP --password heatmap --input-fields-terminated-by ',' --export-dir hdfs://idh05:8020/user/hive/warehouse/数据集市表名/数据集市表分区名 --input-null-string '\\N' --input-null-non-string '\\N' -m slotnum