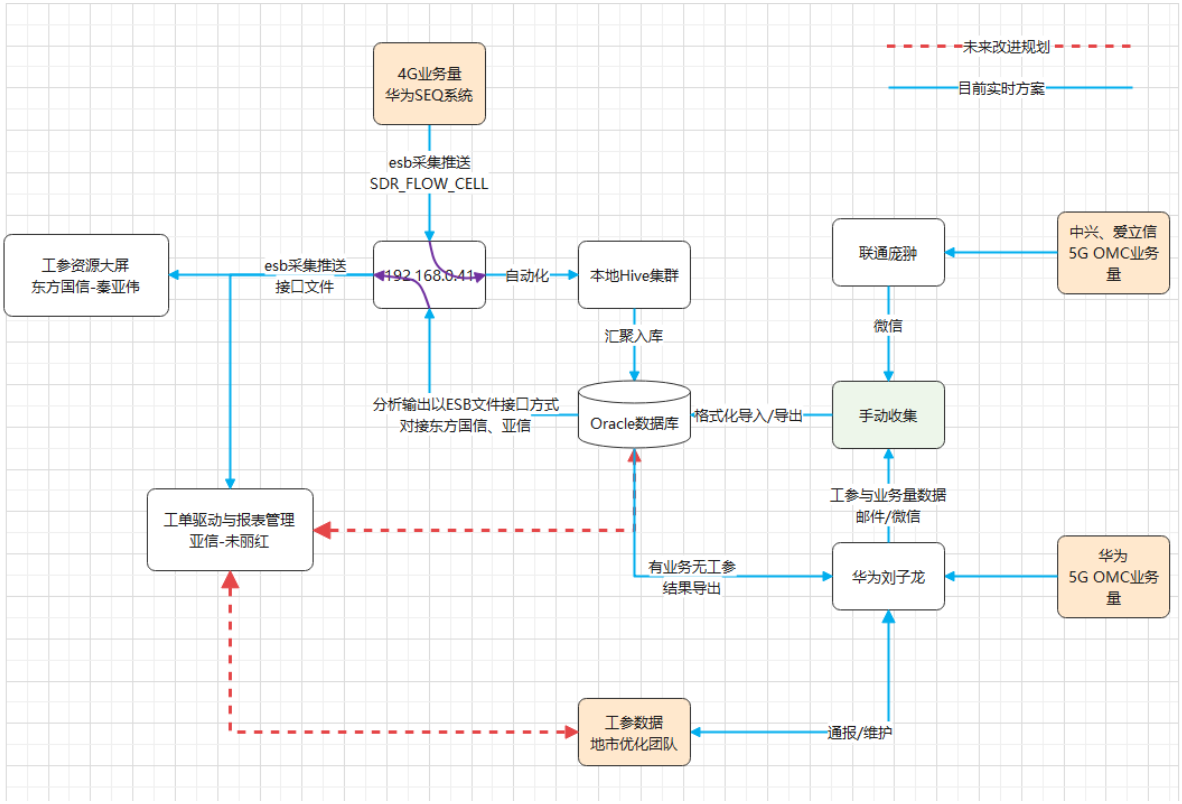
工参治理大屏总结

1、数据流图



2、数据源说明

```
# 1、工参数据
工参数据目前包括45G工参，由陈学安排华为手动从地市收集。

# 2、4G业务量数据
华为SEQ定时输出-->省内ESB-->省内ESB服务器

# 3、5G业务量数据
从无线网管收集，华为区由刘子龙每周5通过邮件发送上周六至本周五的数据；中兴、爱立信由庞翀每周一发送上周一至上周末的数据（通常不准时）。
```

2、数据入库操作

2.1 工参数据

```
# 1、4G工参数据
由陈学安排华为刘子龙从地市收集4G工参数据，在入库前需要进行相应的数据规整。

# 2、工参异常字段检查 收集到的工参为Excel格式，在入库前需要进行：
1）工参字段与数据库字段映射（参考附表1）
```

```
2）检查需要按照数字格式存储在数据库中的字段是否存在异常，如使用isNumber函数在Excel中检查经纬度、enbid、cell_id、下倾角等字段是否为数字，不为数字的需要进行处理。通常经纬度会存在以文本格式在Excel存储的问题。

3）选中整个工作表，将所有逗号替换为|，因为入库采用csv格式导入，而csv默认的分割是逗号，这样会导致数据无法入库。

4）将处理后的Excel工单另存为csv格式的文件，将文件复制到137服务器C:\Users\zhanglei\Desktop\工参治理\工参数据（地址可以根据实际修改）

5）备份数据库工参
SELECT * FROM seq_4g_siteinfo --4G工参数据表

# 执行后，会将seq_4g_siteinfo中数据完全插入SEQ_4G_SITEINFO_backup，并清空seq_4g_siteinfo
call pkg_basedata_governance_v4.backup_4g_siftinfo();
# 查看历史备份
SELECT sdate,count(*) t FROM SEQ_4G_SITEINFO_backup group by sdate

6）修改seq_4g_siteinfo_0712.ctl文件中INFILE后的文件内容

7）执行如下命令即可
C:\Users\zhanglei\Desktop\工参治理\工参数据>sqlldr c##fast491/\"F@st491*321\"@192.168.0.64:1521/fast control=C:\Users\zhanglei\Desktop\工参治理\工参数据\seq_4g_siteinfo_0712.ctl
```

```
# 2、5G工参数据
由陈学安排华为刘子龙从地市收集5G工参数据，在入库前需要进行相应的数据规整。

# 2、工参异常字段检查 收集到的工参为Excel格式，在入库前需要进行：
1）工参字段与数据库字段映射（参考附表1）
2）检查需要按照数字格式存储在数据库中的字段是否存在异常，如使用isNumber函数在Excel中检查经纬度、gnbid、cell_id、下倾角等字段是否为数字，不为数字的需要进行处理。通常经纬度会存在以文本格式在Excel存储的问题。

3）选中整个工作表，将所有逗号替换为|，因为入库采用csv格式导入，而csv默认的分割是逗号，这样会导致数据无法入库。

4）将处理后的Excel工单另存为csv格式的文件，将文件复制到137服务器C:\Users\zhanglei\Desktop\工参治理\工参数据

5）备份数据库工参
SELECT * FROM seq_5g_siteinfo --5G工参数据表

# 执行后，会将seq_5g_siteinfo中数据完全插入SEQ_5G_SITEINFO_backup，并清空seq_5g_siteinfo
call pkg_basedata_governance_v4.backup_5g_siftinfo();
# 查看历史备份
SELECT sdate,count(*) t FROM SEQ_5G_SITEINFO_backup group by sdate

6）修改seq_5g_siteinfo_0712.ctl文件中INFILE后的文件内容

7）执行如下命令即可
C:\Users\zhanglei\Desktop\工参治理\工参数据>sqlldr c##fast491/\"F@st491*321\"@192.168.0.64:1521/fast control=C:\Users\zhanglei\Desktop\工参治理\工参数据\seq_5g_siteinfo_0712.ctl
```

附表1

数据库字段	收集的Excel 4G工参字段映射
SDATE	日期（原本没有，按照数据日期补上）
CITY_CODE	地市编码
CITY_NAME	地市名称
DISTRICT_CODE	区县编码
DISTRICT_NAME	区县名称
NETWORK_NAME	网络名称
PHYSTATION_ADDRESS	准确物理站名
BBU_NAME	BBU名称
ENBID	ENODEBID
CELL_NAME	小区名称
CELL_ID	小区ID
CGI	CGISAI
LON	小区经度
LAT	小区纬度
DIRECTION	方位角
HEIGHT	天线挂高
M_DOWNTILT	机械下倾角
E_DOWNTILT	电子下倾角
STATION_TYPE	站型（宏站/室分）
ISDIGITALINDOOR	
DOWN_FREQ	下行频点
VENDER	所属厂家
OWN_SCHOOLYARD	高校名称
TOWERADDRESS_CODE	铁塔站址编码
PROPERTY	原产权
SCENE	场景1_区域归属
IS_SCENESITE	场景2_是否场景基站
MARKETING_NETWORK	营销网格
TERMINALAMOUNT_5G	5G终端数量
SECTOR_INCOMING	扇区收益

数据库字段	收集的Excel 4G工参字段映射
IS_BUSY	是否超忙
IS_ALIVE	
IS_ALIVE_UPDATE_TIME	
CONSTRUCTION	承建方（联通、电信）
IS_SAME_ADDRESS	是否共站址
SAME_ADDRESS_SITES	对应共站址基站GNBID
IS_SAME_ADDRESS_CT	

附表2

数据库字段	收集的Excel 5G工参字段映射
SDATE	日期（原本没有，按照数据日期补上）
CITY_CODE	地市编码
CITY_NAME	地市名称
DISTRICT_CODE	区县编码
DISTRICT_NAME	区县名称
NETWORK_NAME	网络名称
PHYSTATION_ADDRESS	准确物理站名
STATION_NAME	BBU名称
GNBID	GNBID
CELL_NAME	小区名称
CELL_ID	小区ID
CGISAI	CGISAI
LON	小区经度
LAT	小区纬度
DIRECTION	方位角
HEIGHT	天线挂高
M_DOWNTILT	机械下倾角
E_DOWNTILT	电子下倾角
STATION_TYPE	站型（宏站/室分）
ISDIGITALINDOOR	
DOWN_FREQ	下行频点
VENDER	所属厂家
OWN_SCHOOLYARD	高校名称
TOWERADDRESS_CODE	铁塔站址编码
PROPERTY	原产权
SCENE	场景1_区域归属
IS_SCENESITE	场景2_是否场景基站
MARKETING_NETWORK	营销网格
TERMINALAMOUNT_5G	5G终端数量
SECTOR_INCOMING	扇区收益

数据库字段	收集的Excel 5G工参字段映射
IS_BUSY	是否超忙
IS_ALIVE	
IS_ALIVE_UPDATE_TIME	
CONSTRUCTION	承建方（联通、电信）
IS_SAME_ADDRESS	是否共站址
SAME_ADDRESS_SITES	对应共站址基站ENODEBID
IS_SAME_ADDRESS_CT	共站址运营商

2.2 4G 业务量

```
# 4G业务量数据（以下步骤是自动化进行,正常情况不需要人工干预）
1）数据定时从192.168.0.41:/data/HUAWEI/SDR_FLOW_CELL目录入本地hive库（192.168.0.13）
sqmdb_local.sdr_flow_cell_15min  #入库后的原始XDR数据表名
sqmdb_local.sdr_flow_mapping_ran_ne_id  # 华为提供的关于配套的解析sdr_flow_cell_15min表中5G业务量数据所需的配置表，由于SDR数据中无法实现5G小区粒度的统计，因此不在从SDR中提取5G的业务量数据。
#该环节由李旋提供运维。

2）SDR业务量数据入库到hive后，执行初步过滤汇聚操作，生成如下表：
sqmdb_local.sdr_flow_4g_cell_day    #hive中汇聚出的4g小区天表
sqmdb_local.sdr_flow_4g_cell_hour   #hive中汇聚出的4g小区小时表
sqmdb_local.sdr_flow_4g_enb_day     #hive中汇聚出的4g基站天表
sqmdb_local.sdr_flow_4g_enb_hour    #hive中汇聚出的4g基站小时表

3）汇聚与入库命令 192.168.0.9 定时任务crontab
#sqmdb_local汇聚及入库
30 22 * * * /home/do/hj/hive2ora/sdr_flow.sh >> /home/do/hj/hive2ora/sdr_flow.log 2>&1 &

# 192.168.0.9 /home/do/hj/hive2ora/sdr_flow.sh文件内容
#############################################################################
# File Name: sdr_flow.sh
# Author: haojian
# mail: hjwiki@gmail.com
# Created Time: 2021-03-31 20:12:17
# 在hive汇聚数据并将结果转存到oracle
#############################################################################
#!/bin/bash
set -x
yesterday=`date -d "-1 day" "+%Y%m%d"`
cd /home/do/zhanglei
sh pkg.sh $yesterday > /home/do/hj/hive2ora/log/pkg$yesterday.log 2>&1
cd /home/do/hj/hive2ora
./hdfsget.sh $yesterday > /home/do/hj/hive2ora/log/hive2ora$yesterday.log 2>&1

# pkg.sh内容键附表3。
#该环节由郝建提供运维。
```

附表3 pkg.sh内容：用户hive业务量数据的初步汇聚。

```
[do@hbunicom-web zhanglei]$ cat pkg.sh
startDate=$1

hsql="
set hive.execution.engine=mr;
set mapreduce.job.queuename=hbase;
set io.sort.mb=200;
set hive.tez.container.size=2048;
SET mapreduce.reduce.memory.mb=8192;
SET mapreduce.map.memory.mb=4096;
set hive.input.format=org.apache.hadoop.hive.ql.io.CombineHiveInputFormat;
set mapred.max.split.size=256000000;
set mapred.min.split.size.per.node=100000000;
set mapred.min.split.size.per.rack=100000000;
set hive.merge.mapfiles = true;
set hive.merge.tezfiles=true;
set hive.merge.mapredfiles = true;
set hive.merge.size.per.task = 256000000;
set hive.merge.smallfiles.avgsize=16000000;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.exec.dynamic.partition =true;
set hive.exec.max.dynamic.partitions.pernode=100;
set hive.exec.max.dynamic.partitions =1000;
set hive.exec.max.created.files =100000;
use sqmdb_local;


insert overwrite table sdr_flow_4g_cell_hour partition(sdate)
select
enbid,
cell_id,
sum(l4_dw_throughput) l4_dw_throughput,
sum(l4_ul_throughput) l4_ul_throughput,
sum(throughput) throughput,
layer2id,
layer3id,
sdate
from (SELECT from_unixtime(cast(starttime as int),'yyyyMMddHH') sdate,
conv((case when length(cgisai) = 12 then substr(cgisai, 6, 5) else
substr(cgisai, 6, 4) end), 16,10) enbid,
conv((case when length(cgisai) = 12 then substr(cgisai, -2) else
substr(cgisai, -4) end), 16,10) cell_id,
l4_dw_throughput,
l4_ul_throughput,
(l4_dw_throughput + l4_ul_throughput) throughput,
layer2id,
layer3id
FROM sdr_flow_cell_15min t
WHERE T.RAT = 6
and length(cgisai) = 12
and p_hour >='"$startDate"00'
and p_hour <
from_unixtime(unix_timestamp(date_add(from_unixtime(unix_timestamp('$startD
ate','yyyymmdd'),'yyyy-mm-dd'),3),'yyyy-mm-dd'),'yyyymmddhh00')
and from_unixtime(cast(starttime as int),'yyyyMMdd')  = '"$startDate"' ) m
```

```
group by enbid, cell_id, sdate, layer2id, layer3id ;


insert overwrite table sdr_flow_4g_cell_day partition(sdate)
select
enbid,
cell_id,
sum(l4_dw_throughput) l4_dw_throughput,
sum(l4_ul_throughput) l4_ul_throughput,
sum(throughput) throughput,
layer2id,
layer3id,
sdate
from (SELECT from_unixtime(cast(starttime as int),'yyyyMMdd') sdate,
conv((case when length(cgisai) = 12 then substr(cgisai, 6, 5) else
substr(cgisai, 6, 4) end),16,10) enbid,
conv((case when length(cgisai) = 12 then substr(cgisai, -2) else
substr(cgisai, -4) end),16,10) cell_id,
l4_dw_throughput,
l4_ul_throughput,
(l4_dw_throughput + l4_ul_throughput) throughput,
layer2id,
layer3id
FROM sdr_flow_cell_15min t
WHERE T.RAT = 6
and length(cgisai) = 12
and p_hour >='"$startDate"00'
and p_hour <
from_unixtime(unix_timestamp(date_add(from_unixtime(unix_timestamp('$startD
ate','yyyymmdd'),'yyyy-mm-dd'),3),'yyyy-mm-dd'),'yyyymmddhh00')
and from_unixtime(cast(starttime as int),'yyyyMMdd')  = '"$startDate"') m
group by sdate, enbid, cell_id, layer2id, layer3id;

insert overwrite table sdr_flow_4g_enb_hour partition(sdate)
select enbid,
sum(l4_dw_throughput) l4_dw_throughput,
sum(l4_ul_throughput) l4_ul_throughput,
sum(throughput) throughput,
layer2id,
layer3id,
sdate
from (SELECT from_unixtime(cast(starttime as int),'yyyyMMddHH') sdate,
conv((case when length(cgisai) = 12 then substr(cgisai, 6, 5) else
substr(cgisai, 6, 4) end),16,10) enbid,
l4_dw_throughput,
l4_ul_throughput,
(l4_dw_throughput + l4_ul_throughput) throughput,
layer2id,
layer3id
FROM sdr_flow_cell_15min t
WHERE T.RAT = 6
and length(cgisai) = 12
and p_hour >='"$startDate"00'
and p_hour <
from_unixtime(unix_timestamp(date_add(from_unixtime(unix_timestamp('$startD
ate','yyyymmdd'),'yyyy-mm-dd'),3),'yyyy-mm-dd'),'yyyymmddhh00')
and from_unixtime(cast(starttime as int),'yyyyMMdd')  = '"$startDate"') m
group by sdate, enbid, layer2id, layer3id;
```

```

insert overwrite table sdr_flow_4g_enb_day partition(sdate)
select enbid,
sum(l4_dw_throughput) l4_dw_throughput,
sum(l4_ul_throughput) l4_ul_throughput,
sum(throughput) throughput,
layer2id,
layer3id,
sdate
from (SELECT from_unixtime(cast(starttime as int),'yyyyMMdd') sdate,
conv((case when length(cgisai) = 12 then substr(cgisai, 6, 5) else
substr(cgisai, 6, 4) end),16,10) enbid,
conv((case when length(cgisai) = 12 then substr(cgisai, -2) else
substr(cgisai, -4) end),16,10) cell_id,
l4_dw_throughput,
l4_ul_throughput,
(l4_dw_throughput + l4_ul_throughput) throughput,
layer2id,
layer3id
FROM sdr_flow_cell_15min t
WHERE T.RAT = 6
and length(cgisai) = 12
and p_hour >='"$startDate"00'
and p_hour <
from_unixtime(unix_timestamp(date_add(from_unixtime(unix_timestamp('$startD
ate','yyyymmdd'),'yyyy-mm-dd'),3),'yyyy-mm-dd'),'yyyymmddhh00')
and from_unixtime(cast(starttime as int),'yyyyMMdd')  = '"$startDate"') m
group by sdate, enbid, layer2id, layer3id;
"

#echo $hsql #>./h.sql
hive -e "$hsql"
```

2.3 5G业务量

```
# 5G的业务量数据，华为区由刘子龙每周5通过邮件发送上周六至本周五的数据；中兴、爱立信由庞翀每周一发送上周一至上周末的数据（通常不准时）。

# 1）5G业务量数据目前采用手工的方式收集，导致无法实现自动入库。目前在收到数据后，将数据Excel格式另存为utf-8格式的csv文件（utf8格式入库后不乱码），使用dbeaver导入。 注意：在另存为csv格式前，需要对相关数据进行检查，如日期的格式。

# 2）导入的目标表：sdr_flow_cell_throughput
```

2.4 业务量数据检查

```
--按地市统计
SELECT * FROM
(
SELECT sdate,city,COUNT(*) COUNT_NUM from sdr_flow_cell_throughput
where sdate>sysdate-20
GROUP BY sdate,city
)
```

```
PIVOT (SUM(COUNT_NUM) total FOR city IN ('张家口', '秦皇岛', '沧州', '唐山', '雄安', '保定','石家庄', '承德', '廊坊', '邢台', '衡水', '邯郸'))
ORDER BY sdate

--按厂家统计
SELECT * FROM
(
SELECT sdate,vender,COUNT(*) COUNT_NUM from sdr_flow_cell_throughput
where sdate>sysdate-20
GROUP BY sdate,vender
)
PIVOT (SUM(COUNT_NUM) total FOR vender IN ('华为', '中兴', '爱立信'))
ORDER BY sdate
```

按地市统计

```
SELECT * FROM
(
SELECT sdate,city,COUNT(*) COUNT_NUM from sdr_flow_cell_throughput
where sdate>sysdate-20
GROUP BY sdate,city
)
PIVOT (SUM(COUNT_NUM) total FOR city IN ('张家口', '秦皇岛', '沧州', '唐山', '雄安', '保定','石家庄', '承德', '廊坊', '邢台', '衡水', '邯郸'))
ORDER BY sdate
```

	SDATE	'张家口'_TOTAL	'秦皇岛'_TOTAL	'沧州'_TOTAL	'唐山'_TOTAL	'雄安'_TOTAL	'保定'_TOTAL	'石家庄'_TOTAL	'承德'_TOTAL	'廊坊'_TOTAL	'邢台'_TOTAL	'衡水'_TOTAL	'邯郸'_TOTAL
1	2021/7/21	2129	2365	3988	8435	1274	5051	9556	1131	3843	5100	1492	7320
2	2021/7/22	2139	2370	3965	8424	1286	5050	9454	1134	3904	5130	1522	7308
3	2021/7/23	2156	2363	3956	8454	1418	5057	9575	1139	3913	5124	1527	7308
4	2021/7/24	2155	2354	3951	7850	1411	5043	9588	1136	3892	5120	1530	7178
5	2021/7/25	2155	2358	3948	7832	1407	5044	9586	1136	3850	5132	1530	7178
6	2021/7/26	2176	2363	2430	379	1407	1925	9546	1136	3957		1533	
7	2021/7/27	2177	2368	2444	379	1410	1931	9550	1142	3979		1538	
8	2021/7/28	2177	2371	2444	379	1422	1935	9559	1136	3952		1532	
9	2021/7/29	2184	2375	2447	380	1426	1934	9556	1130	3943		1533	

按厂家统计

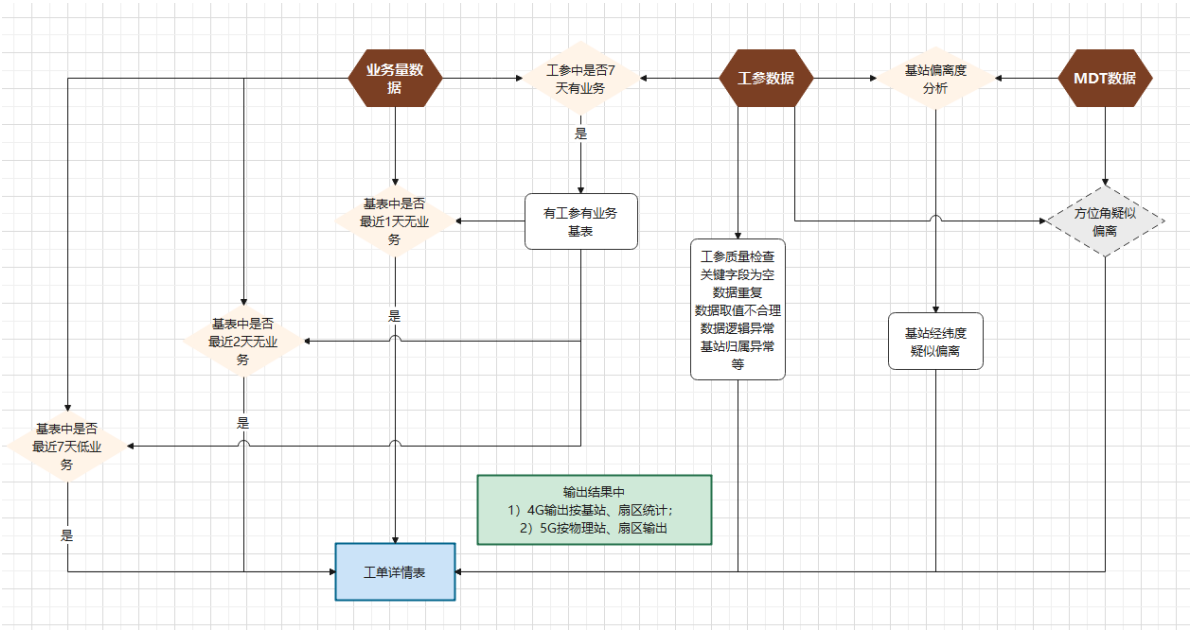
```
SELECT * FROM
(
SELECT sdate,vender,COUNT(*) COUNT_NUM from sdr_flow_cell_throughput
where sdate>sysdate-20
GROUP BY sdate,vender
)
PIVOT (SUM(COUNT_NUM) total FOR vender IN ('华为', '中兴', '爱立信'))
ORDER BY sdate
```

	SDATE	'华为'_TOTAL	'中兴'_TOTAL	'爱立信'_TOTAL
1	2021/7/21	22848	24206	4630
2	2021/7/22	22847	24206	4633
3	2021/7/23	23088	24257	4645
4	2021/7/24	23052	23519	4637
5	2021/7/25	23003	23516	4637
6	2021/7/26	23120	3732	
7	2021/7/27	23184	3734	
8	2021/7/28	23173	3734	
9	2021/7/29	23174	3734	

3、业务逻辑设计

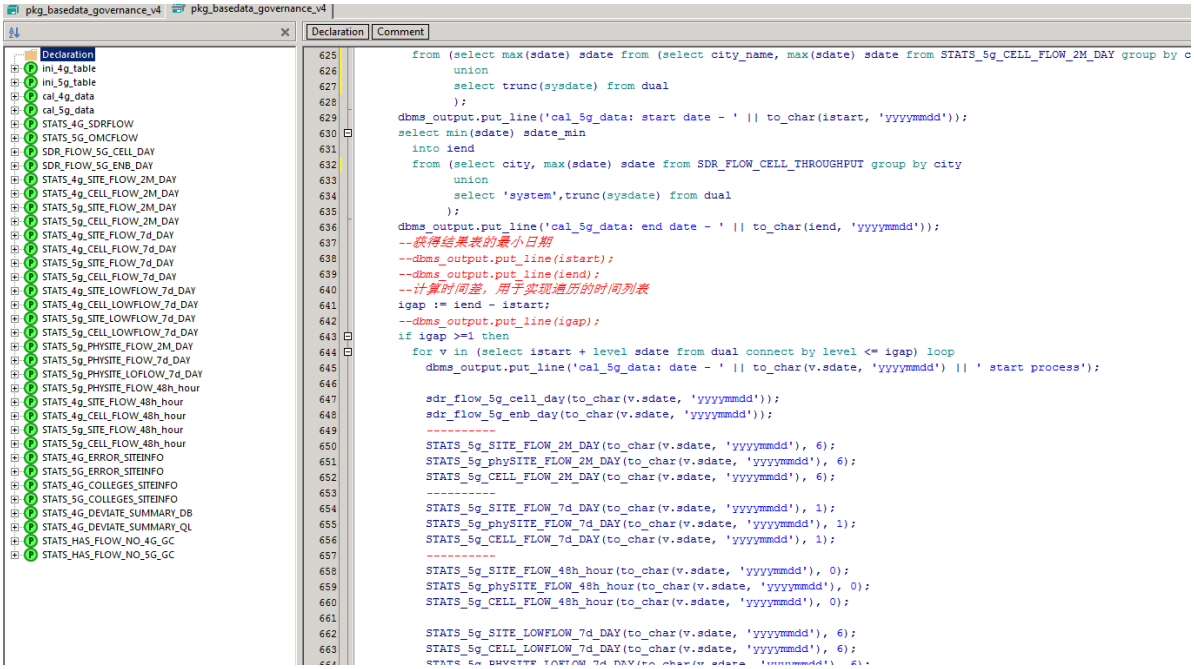
目前基本完成了工参资源大屏需求的业务开发。业务需求包括7天有业务4/5G基站/扇区统计、2天无业务4/5G基站/扇区统计、1天无业务4/5G基站/扇区统计、基站/扇区工参质量检查（包含工参不全不准、经纬度偏离）、统计报表等。

3.1 业务分析流程



3.2 业务分析程序

详细代码见7章节



3.3 逻辑说明

```
# 1）统计的目标是工参中有业务的扇区或基站，在工参中但无业务的扇区不进行质量维护；
# 2）在工参中有is_alive字段标记工参中扇区在本地业务统计周期是否有业务量；
# 3）是否有业务目前有客户确定，目前是最近7天总业务量大于0即算有业务（时间门限可能会变更）；
# 4）对于2天无业务、1天无业务的统计，也是在最近7天有业务的基础上统计的最新的2天或1天无业务，类似于新断站的统计；
# 5）对于4g，基站是按照city，enbid进行统计，扇区是按照city，enbid，cell_id进行统计；4g不按照物理站的角度统计基站数量；
# 6）对于5g，基站是按照city，gnbid，physite进行统计，扇区是按照city，gnbid，physite，cell_id进行统计；因为5g的gnbid通常下挂基站较多，因此由客户确定为使用物理站统计基站数量。
# 7）对于低业务统计仍是按照7天周期统计，扇区业务量门限是 0<limit<2GByte，基站业务量门限是0<limit<7Gbyte
# 8）由业务无工参统计，用于下发地市对工参数据进行完善；
```

3.4 表功能说明

在c##fast491数据库中，工参资源治理的表分别是SDR、SEQ、STATS开头。其中，SDR为业务量数据表；SEQ为工参配置表；STATS为汇聚结果表或统计报表。

表名	表用途说明
SEQ_CFG_COLLEGES_GEO	保存校园基站信息
SEQ_5G_SITEINFO_BACKUP	5G工参的备份数据
SEQ_4G_SITEINFO_BACKUP	4G工参的备份数据
SEQ_MDT_GRID50_DBSCAN	DBSCAN聚类后的栅格数据
SEQ_MDT_GRID50_CELL	原始MDT栅格数据
SEQ_5G_SITEINFO_GEO	5G工参的数据含GEO
SEQ_5G_SITEINFO	5G工参的数据
SEQ_4G_SITEINFO_GEO	4G工参的数据含GEO
SEQ_4G_SITEINFO	4G工参的数据
SDR_FLOW_REGIN_CONF	区县编码配置表，用于SDR业务量表关联区县
SDR_FLOW_CITY_CONF	地市编码配置表，用于SDR业务量表关联地市
SDR_FLOW_CELL_THROUGHPUT	5G网管收集的业务量数据表-原始表
SDR_FLOW_5G_ENB_HOUR	5G基站小时业务量表-暂不使用
SDR_FLOW_5G_ENB_DAY	5G基站天业务量表-暂不使用
SDR_FLOW_5G_CELL_DAY	5G扇区天业务量表-暂不使用
SDR_FLOW_4G_ENB_HOUR	4G基站小时业务量表-暂不使用
SDR_FLOW_4G_ENB_DAY	4G基站小时业务量表-暂不使用
SDR_FLOW_4G_CELL_HOUR	4G基站小时业务量表-暂不使用
SDR_FLOW_4G_CELL_DAY	4G基站小时业务量表-暂不使用
STATS_HAS_FLOW_NO_5G_GC	输出报表：4G有业务无工参
STATS_HAS_FLOW_NO_4G_GC	输出报表：5G有业务无工参
STATS_5G_SITE_LOWFLOW_7D_DAY	5G基站7天低业务表-暂未使用
STATS_5G_SITE_FLOW_7D_DAY	5G基站2天无业务表-暂未使用
STATS_5G_SITE_FLOW_48H_HOUR	5G基站1天无业务表-暂未使用
STATS_5G_SITE_FLOW_2M_DAY	5G基站1周有业务表-暂未使用
STATS_5G_PHYSITE_LOFLOW_7D_DAY	5G物理基站7天低业务表
STATS_5G_PHYSITE_FLOW_7D_DAY	5G物理基站2天无业务表
STATS_5G_PHYSITE_FLOW_48H_HOUR	5G物理基站1天无业务表
STATS_5G_PHYSITE_FLOW_2M_DAY	5G物理基站1周有业务表
STATS_5G_ERR_FLAG	5G扇区问题类型统计表

表名	表用途说明
STATS_5G_ERROR_SITEINFO	5G扇区工参质量问题详表
STATS_5G_COLLEGES_SITEINFO	5G校园扇区表
STATS_5G_CELL_LOWFLOW_7D_DAY	5G扇区7天低业务表
STATS_5G_CELL_FLOW_7D_DAY	5G扇区2天无业务表
STATS_5G_CELL_FLOW_48H_HOUR	5G扇区1天无业务表
STATS_5G_CELL_FLOW_2M_DAY	5G扇区1周有业务表
STATS_4G_SITE_LOWFLOW_7D_DAY	4G基站7天低业务表
STATS_4G_SITE_FLOW_7D_DAY	4G基站2天无业务表
STATS_4G_SITE_FLOW_48H_HOUR	4G基站1天无业务表
STATS_4G_SITE_FLOW_2M_DAY	4G基站1周有业务表
STATS_4G_SITEINFO_DEVIATE	4G基站经纬度偏离统计报表-暂未使用
STATS_4G_SITEINFO_CELL_DEVIATE	4G扇区经纬度偏离统计报表
STATS_4G_ERR_FLAG	4G扇区问题类型统计表
STATS_4G_ERROR_SITEINFO	4G扇区工参质量问题详表
STATS_4G_DEVIATE_SUMMARY_QL	4G基站经纬度偏离统计报表-全量计算
STATS_4G_DEVIATE_SUMMARY_DB	4G基站经纬度偏离统计报表-聚类分析
STATS_4G_COLLEGES_SITEINFO	4G校园扇区表
STATS_4G_CELL_LOWFLOW_7D_DAY	4G扇区7天低业务表
STATS_4G_CELL_FLOW_7D_DAY	4G扇区2天无业务表
STATS_4G_CELL_FLOW_48H_HOUR	4G扇区1天无业务表
STATS_4G_CELL_FLOW_2M_DAY	4G扇区1周有业务表

3.5、基站经纬度偏离

3.5.1 提取数据

```
# 1）基站经纬度偏离目前每月计算一次。
# 基站经纬度偏离使用的是原始MDT数据汇聚而来；从fastdo_lte库的tdlte_mro_locate_hour表二次汇聚而来。该操作通常由郝建帮忙完成。
create table hb_mdt_grid_cell_manual as
select
day sdate,
city,
floor(s_cell_id/256) enbid,
s_cell_id eci,
floor(orig_lon/0.00045) gridx,
floor(orig_lat/0.00045) gridy,
sum(case when (s_rsrp<-20 and s_rsrp>-140) then 1 else 0 end) rsrp_samples
```

```
from fastdo_lte.tdlte_mro_locate_hour
where orig_lon between 113 and 120
and orig_lat between 36 and 43
and day = '20210512'
and s_cell_id >0
group by day,city,floor(s_cell_id/256) ,s_cell_id,floor(orig_lon/0.00045)
,floor(orig_lat/0.00045) ;
```

3.5.2 汇聚导出

```
# 2）汇聚结果导出
# 采用beeline可以直接导出为csv方式，下面分别提供单独导出与批量导出脚本。

# 单地市导出
beeline -u "jdbc:hive2://10.168.26.1:10000/fastdo_lte;" -n zhihui030 -p
mAGR_H7b2u_Gbi7N --outputformat=csv2 --showHeader=true -e "select * from
fastdo_lte.hb_mdt_grid_cell_manual_zl where ct='BAODING' and rsrp_samples>5
and ds='20210726'" > BAODING.csv

# 所有地市批量导出
for city in
{BAODING,CANGZHOU,CHENGDE,HANDAN,HENGSHUI,LANGFANG,QINHUANGDAO,SHIJIAZHUANG,
TANGSHAN,XINGTAI,XIONGAN,ZHANGJIAKOU}; do echo
beeline -u "jdbc:hive2://10.168.26.1:10000/fastdo_lte;" -n zhihui030 -p
mAGR_H7b2u_Gbi7N --outputformat=csv2 --showHeader=true -e "select * from
fastdo_lte.hb_mdt_grid_cell_manual_zl where ct='$city' and rsrp_samples>5
and  ds='20210726'" > $city'.csv';
done
```

3.5.3 聚类处理

```
# python脚本路径在133.96.92.137
C:\Users\zhanglei\PycharmProjects\untitled2\dbscan.py，内容如下

# -*- coding: utf-8 -*-
"""
-------------------------------------------------
   File Name:   deviate_hebei
   Author:      ZhangLei
   date:        2021/05/08
-------------------------------------------------
"""

import sklearn.cluster as skc
import pandas as pd
from pandas import DataFrame
import os
import multiprocessing as mp
import logging


# 主处理函数
def process(icity):
    logging.basicConfig(level=logging.DEBUG,
                        filename=icity + '.log',
```

```
                        filemode='a',  # 模式，有w和a，w就是写模式，每次都会重新写日志，覆盖之前的日志
                        # a是追加模式，默认如果不写的话，就是追加模式
                        format='%(asctime)s - [%(filename)s-%(funcName)s-line:%(lineno)d] - %(levelname)s: %(message)s'
                        # 日志格式
                        )

    name = mp.current_process().name
    logging.debug("进程:" + name + "启动!")
    print("进程:" + name + "启动!")
    os.chdir(r'./MDT_GRID')

    # 后期需自定义修改文件名,用于适配
    infile = os.path.join(os.getcwd(), icity + '.csv')
    logging.debug('文件地址:' + infile)
    # 打印输入的日志文件
    logging.debug('输入文件:' + infile)

    # 注意：需要根据文件的分隔符进行修改
    # data = pd.read_csv(infile,
    #                    usecols=['sdate', 'city', 'enbid', 'eci', 'gridx', 'gridy', 'rsrp_samples'],sep='\t')
    data = pd.read_csv(infile,
                       usecols=['sdate', 'city', 'enbid', 'eci', 'gridx', 'gridy', 'rsrp_samples'], sep=',')

    enodebs = data['enbid'].drop_duplicates()  # 遍历enodeb

    logging.debug('待处理基站数量:' + str(len(enodebs)))

    result_circle = []

    for enbid in enodebs:
        logging.debug("###处理基站编号：" + str(enbid))
        temp = data[data['enbid'].isin([enbid])]
        X = temp[['gridx', 'gridy']].drop_duplicates()
        for min_sample in range(41, 1, -2):
            db = skc.DBSCAN(eps=4, min_samples=min_sample).fit(X)  # DBSCAN聚类方法 还有参数，matric = ""距离计算方法
            labels = db.labels_  # 和X同一个维度，labels对应索引序号的值 为她所在簇的序号。若簇编号为-1，表示为噪声
            raito = len(labels[labels[:] == -1]) / len(labels)  # 计算噪声点个数占总数的比例
            if raito < 0.2:
                sample_select = min_sample
                logging.debug('满足噪声比例的MIN_SAMPLE值：' + str(sample_select) + '，噪声比为：' + str(raito))
                break

        # 获取分簇的数目
        # labels结果如[ 0  1  0  2  0  1  1 -1  0  2  2]，一般-1表示噪声点
        n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)

        logging.debug('分簇的数目: %d' % n_clusters_)

        # print("轮廓系数: %0.3f" % metrics.silhouette_score(X, labels)) #轮廓系数评价聚类的好坏
```

```
        cluster_grid = X[labels != -1]

        # 一个小区可能聚类多个簇，找出最大的簇：cluster_select就是选择最大的簇的grid列表
        for index, row in cluster_grid.iterrows():
            result_circle.append([enbid, row["gridx"], row["gridy"]])

    outputfile = os.path.join(os.getcwd(), icity + '_DBSCAN_SITE.csv')
    logging.debug('输出文件:' + outputfile)
    data_yuanxin = DataFrame(result_circle)
    data_yuanxin.columns = ['enbid', 'gridx', 'gridy']
    data_yuanxin.to_csv(outputfile, index=False, encoding='gbk')


if __name__ == '__main__':
    # 单进程运行
    # process('HENGSHUI')
    # exit(0)

    # 多线程运行
    citys = ['BAODING', 'CANGZHOU', 'CHENGDE', 'HANDAN', 'HENGSHUI', 'LANGFANG', 'QINHUANGDAO',
             'SHIJIAZHUANG', 'TANGSHAN', 'XINGTAI', 'XIONGAN', 'ZHANGJIAKOU']
    for v in citys:
        p = mp.Process(name='子线程_' + v, target=process, args=(v,))
        p.start()
    p.join()

    print('All subprocesses done.')
```

python脚本执行效果及输出：

```
C:\Users\zhanglei\anaconda3\python.exe C:/Users/zhanglei/PycharmProjects/untitled2/dbscan.py
进程:子线程_XIONGAN启动!
进程:子线程_CANGZHOU启动!
进程:子线程_QINHUANGDAO启动!
进程:子线程_TANGSHAN启动!
进程:子线程_LANGFANG启动!
进程:子线程_HENGSHUI启动!
进程:子线程_ZHANGJIAKOU启动!
进程:子线程_HANDAN启动!
进程:子线程_SHIJIAZHUANG启动!
进程:子线程_BAODING启动!
进程:子线程_CHENGDE启动!
进程:子线程_XINGTAI启动!
All subprocesses done.
```

dbscan_site.ctl	2021/8/6 12:36	CTL 文件	2 KB
mdt_site.ctl	2021/8/6 13:22	CTL 文件	2 KB
BAODING.csv	2021/8/6 10:19	Microsoft Exce...	99,559 KB
BAODING_DBSCAN_SITE.csv	2021/8/6 11:23	Microsoft Exce...	29,521 KB
CANGZHOU.csv	2021/8/6 10:19	Microsoft Exce...	77,267 KB
CANGZHOU_DBSCAN_SITE.csv	2021/8/6 11:20	Microsoft Exce...	22,578 KB
CHENGDE.csv	2021/8/6 10:22	Microsoft Exce...	34,890 KB
CHENGDE_DBSCAN_SITE.csv	2021/8/6 11:17	Microsoft Exce...	10,111 KB
HANDAN.csv	2021/8/6 10:22	Microsoft Exce...	87,011 KB
HANDAN_DBSCAN_SITE.csv	2021/8/6 11:20	Microsoft Exce...	23,558 KB
HENGSHUI.csv	2021/8/6 10:22	Microsoft Exce...	37,169 KB
HENGSHUI_DBSCAN_SITE.csv	2021/8/6 11:16	Microsoft Exce...	10,055 KB
LANGFANG.csv	2021/8/6 10:22	Microsoft Exce...	83,452 KB
LANGFANG_DBSCAN_SITE.csv	2021/8/6 11:18	Microsoft Exce...	20,968 KB
QINHUANGDAO.csv	2021/8/6 10:21	Microsoft Exce...	36,955 KB
QINHUANGDAO_DBSCAN_SITE.csv	2021/8/6 11:15	Microsoft Exce...	9,060 KB
SHIJIAZHUANG.csv	2021/8/6 10:20	Microsoft Exce...	102,066 KB
SHIJIAZHUANG_DBSCAN_SITE.csv	2021/8/6 11:22	Microsoft Exce...	26,017 KB
TANGSHAN.csv	2021/8/6 10:21	Microsoft Exce...	117,978 KB
TANGSHAN_DBSCAN_SITE.csv	2021/8/6 11:21	Microsoft Exce...	31,166 KB
XINGTAI.csv	2021/8/6 10:21	Microsoft Exce...	90,605 KB
XINGTAI_DBSCAN_SITE.csv	2021/8/6 11:19	Microsoft Exce...	23,494 KB
XIONGAN.csv	2021/8/6 10:20	Microsoft Exce...	8,649 KB
XIONGAN_DBSCAN_SITE.csv	2021/8/6 11:13	Microsoft Exce...	2,113 KB
ZHANGJIAKOU.csv	2021/8/6 10:20	Microsoft Exce...	41,360 KB
ZHANGJIAKOU_DBSCAN_SITE.csv	2021/8/6 11:18	Microsoft Exce...	11,285 KB
dbscan_site.log	2021/8/6 12:39	文本文档	6 KB
mdt_site.log	2021/8/6 13:28	文本文档	6 KB
sqlldr.txt	2021/8/11 15:51	文本文档	1 KB

3.5.4 结果入库

```
# 检查并修改dbscan_site.ctl、mdt_site.ctl文件内容，默认不需要更新。
# 在当前目录的命令行窗口，逐行执行sqlldr.txt中的语句，即：
sqlldr c##fast491/\"F@st491*321\"@192.168.0.64:1521/fast
control=C:\Users\zhanglei\PycharmProjects\untitled2\MDT_GRID\mdt_site.ctl

sqlldr c##fast491/\"F@st491*321\"@192.168.0.64:1521/fast
control=C:\Users\zhanglei\PycharmProjects\untitled2\MDT_GRID\dbscan_site.ctl


```

3.5.5 偏离分析

```
# 5）入库后，请将本次数据备份到当前目录下的out_202107文件夹（文件夹名按数据处理时间修改）。
# 执行oracle中的偏离度数据分析过程（需要在工参业务量数据更新完成后执行）：

call pkg_basedata_governance_v4.STATS_4G_DEVIATE_SUMMARY_QL(null);--参数为null，表示采用最新MDT聚类数据进行计算，不为空参数按指定日期计算
call pkg_basedata_governance_v4.STATS_4G_DEVIATE_SUMMARY_DB(null);--参数为null，表示采用最新MDT栅格数据进行计算，不为空参数按指定日期计算
```

5、周期输出与报表统计

5.1 数据处理

```
--1、初始化表（清空所有结果表），按需执行
--call pkg_basedata_governance_v4.ini_4g_table();--会清空所有4g结果表，谨慎
--call pkg_basedata_governance_v4.ini_5g_table();--会清空所有5g结果表，谨慎

--2、批量执行计算，目前暂不使用。
call pkg_basedata_governance_v4.cal_4g_data();
call pkg_basedata_governance_v4.cal_5g_data();

--3、按指定日期手动计算，目前的使用方式
# 业务量分析处理
call pkg_basedata_governance_v4.STATS_4G_SDRFLOW(null);              --参数为null，表示采用最新SEQ数据进行计算，不为空参数按指定日期计算
call pkg_basedata_governance_v4.STATS_5G_OMCFLOW(null);              --参数为null，表示采用最新OMC数据进行计算，不为空参数按指定日期计算

# 工参质量检查
call pkg_basedata_governance_v4.STATS_4G_ERROR_SITEINFO(null);      --参数为null，表示采用最新工参数据进行计算，不为空参数按指定日期计算
call pkg_basedata_governance_v4.STATS_5G_ERROR_SITEINFO(null);      --参数为null，表示采用最新工参数据进行计算，不为空参数按指定日期计算

# 基站偏离度检查
call pkg_basedata_governance_v4.STATS_4G_DEVIATE_SUMMARY_QL(null);--参数为null，表示采用最新MDT聚类数据进行计算，不为空参数按指定日期计算
call pkg_basedata_governance_v4.STATS_4G_DEVIATE_SUMMARY_DB(null);--参数为null，表示采用最新MDT栅格数据进行计算，不为空参数按指定日期计算
# 逻辑步骤说明
1）fdd_MDT_cell_position：统计每个小区eci（enb、cellid）纬度的加权主覆盖中心经纬度；
2）fdd_MDT_ENB_position：以fdd_MDT_cell_position、seq_4g_siteinfo，计算物理站的虚拟经纬度；
3）fdd_MDT_cell_position_distance：以fdd_MDT_cell_position、seq_4g_siteinfo，计算小区与覆盖中心的距离；
4）stats_4G_SITEINFO_DEVIATE：fdd_MDT_cell_position_distance、fdd_MDT_ENB_position，计算物理站虚拟位置与工参基站位置的偏离距离（多个小区取最大作为基站的偏离距离）；
5）按需根据偏离距离过滤并进行投诉转置


--4、统计有业务无工参的小区，需手动调用
# 有业务无工参的报表统计
call pkg_basedata_governance_v4.STATS_HAS_FLOW_NO_4G_GC(null);      --参数为null，表示采用最新SEQ数据进行计算，不为空参数按指定日期计算
```

```
call pkg_basedata_governance_v4.STATS_HAS_FLOW_NO_5G_GC(null);     --参数为null，表示采用最新OMC数据进行计算，不为空参数按指定日期计算
```

5.2 统计报表

```
# 下面其中1、2、5、6需要将数据填到Excel模板中，客户前期经常关注分析结果。
# 4接口用到。

# 1）v1.4 版本报表-省级粒度-excel展示汇报
select DATA_TYPE "业务名称", '-' "行政区名称",-1 "行政区编码",CELL_TOTAL "小区数量",CELL_NUM "单位数量",RATE "百分占比" from v_STATS_BIGSCREEN_SUMMARY_22  --20210707 update

# 2）v1.4 版本报表-市级粒度-excel展示汇报
SELECT * FROM v_STATS_BIGSCREEN_WORKORDER_21 --20210707 update

# 3）v1.4 版本报表-市级粒度-excel展示汇报--暂未使用
SELECT * FROM v_STATS_BIGSCREEN_WORKORDER_32 --20210707 update

# 4）v1.4 版本报表-增加行政区维度-目前用于接口00号文件推送，东方国信大屏使用
select DATA_TYPE "业务名称", district_name "行政区名称",district_code "行政区编码",site_num "小区数量",DATA_NUM "单位数量",rate "百分占比" from v_STATS_BIGSCREEN_SUMMARY_32 --20210707 update


# 5）v1.4 版本报表-新增按照运营商进行统计,excel展示汇报
SELECT * FROM V_STATS_BIGSCREEN_SUMMARY_22_Y --20210707 update

# 6）v1.4 版本报表-统计透视,按承建方透视统计,excel展示汇报
SELECT *
  FROM (SELECT * FROM V_STATS_BIGSCREEN_SUMMARY_22_Y)

PIVOT(SUM(CELL_TOTAL) CELL_TOTAL, sum(CELL_NUM) CELL_NUM, sum(RATE) RATE
   FOR CONSTRUCTION IN('联通扇区', '电信扇区'))
 ORDER BY DATA_TYPE
```

5.3 推送结果到ESB接口机

```
# 在数据检查无误后，需要将数据上传接口机。
# 1）双击接口程序即可一键上传，可以通过观察同目录下的日志查看上传结果。
# D:\gongcanzhili\02_gczl\10接口程序源码\oracle_2_esb\gooracle.exe

# 日志信息如下：
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:119: 查询结果导出csv成功：./20210817/gcToyxgx_22_4g_site_deviate1km_20210817.csv，行数：2119
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:119: 查询结果导出csv成功：./20210817/gcToyxgx_23_4g_site_deviate2km_20210817.csv，行数：367
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1315: Ftp连接成功
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1322: Ftp登录成功
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1329: 当前工作目录:/data
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1336: 切换目录成功:/data/bigscreen
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1357: 创建目录:20210817 成功
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1365: 切换目录成功
```

```
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1371: 打印当前目录:20210817
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1381: 文件读取完成,准备上传...
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1388: 文件上传成功:/data/bigscreen20210817/gcToyxgx_00_bigscreen_20210817.csv
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1381: 文件读取完成,准备上传...
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1388: 文件上传成功:/data/bigscreen20210817/gcToyxgx_01_4g_site_youyewu_7d_20210817.csv
Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1381: 文件读取完成,准备上传...
Test- 2021/08/17 09:03:53 query_data_from_oracle_2_csv.go:1388: 文件上传成功:/data/bigscreen20210817/gcToyxgx_02_4g_cell_youyewu_7d_20210817.csv

# 2）检查文件上传结果：登录192.168.0.41服务器
[do@hbunicom-nifi-41 20210817]$ pwd
/data/bigscreen/20210817
[do@hbunicom-nifi-41 20210817]$ ll
总用量 176268
-rw-r--r-- 1 do do   377585 8月  17 09:01 gcToyxgx_00_bigscreen_20210817.csv
-rw-r--r-- 1 do do 12100785 8月  17 09:01 gcToyxgx_01_4g_site_youyewu_7d_20210817.csv
-rw-r--r-- 1 do do 41556169 8月  17 09:01 gcToyxgx_02_4g_cell_youyewu_7d_20210817.csv
-rw-r--r-- 1 do do  1893669 8月  17 09:01 gcToyxgx_03_5g_site_youyewu_7d_20210817.csv
-rw-r--r-- 1 do do  7883155 8月  17 09:01 gcToyxgx_04_5g_cell_youyewu_7d_20210817.csv
-rw-r--r-- 1 do do    45842 8月  17 09:01 gcToyxgx_05_4g_site_wuyewu_2d_20210817.csv
-rw-r--r-- 1 do do   497571 8月  17 09:01 gcToyxgx_06_4g_cell_wuyewu_2d_20210817.csv
-rw-r--r-- 1 do do  2592161 8月  17 09:01 gcToyxgx_07_4g_site_diyewu_2d_20210817.csv
-rw-r--r-- 1 do do 14921970 8月  17 09:01 gcToyxgx_08_4g_cell_diyewu_2d_20210817.csv
-rw-r--r-- 1 do do    11789 8月  17 09:01 gcToyxgx_09_5g_site_wuyewu_2d_20210817.csv
-rw-r--r-- 1 do do    70762 8月  17 09:01 gcToyxgx_10_5g_cell_wuyewu_2d_20210817.csv
-rw-r--r-- 1 do do    73151 8月  17 09:01 gcToyxgx_11_5g_site_diyewu_2d_20210817.csv
-rw-r--r-- 1 do do   303511 8月  17 09:01 gcToyxgx_12_5g_cell_diyewu_2d_20210817.csv
-rw-r--r-- 1 do do    64798 8月  17 09:01 gcToyxgx_13_4g_site_wuyewu_1d_20210817.csv
-rw-r--r-- 1 do do   742229 8月  17 09:01 gcToyxgx_14_4g_cell_wuyewu_1d_20210817.csv
-rw-r--r-- 1 do do    18369 8月  17 09:01 gcToyxgx_15_5g_site_wuyewu_1d_20210817.csv
-rw-r--r-- 1 do do   107904 8月  17 09:01 gcToyxgx_16_5g_cell_wuyewu_1d_20210817.csv
-rw-r--r-- 1 do do     5168 8月  17 09:01 gcToyxgx_17_4g_cell_errors_20210817.csv
-rw-r--r-- 1 do do     4040 8月  17 09:01 gcToyxgx_18_5g_cell_errors_20210817.csv
```

```
-rw-r--r-- 1 do do    70428 8月  17 09:01 gcToyxgx_19_4g_site_deviate_20210817.csv
-rw-r--r-- 1 do do 83563954 8月  17 09:01 gcToyxgx_20_4g_seq_siteinfo_20210817.csv
-rw-r--r-- 1 do do 12876151 8月  17 09:01 gcToyxgx_21_5g_seq_siteinfo_20210817.csv
-rw-r--r-- 1 do do   567631 8月  17 09:01 gcToyxgx_22_4g_site_deviate1km_20210817.csv
-rw-r--r-- 1 do do    93183 8月  17 09:01 gcToyxgx_23_4g_site_deviate2km_20210817.csv
-rw-rw-r-- 1 do do       64 8月  17 09:04 note.txt
[do@hbunicom-nifi-41 20210817]$

# 3）其它一些注意事项
数据中大数字（长度超过6位）会采用科学记数法的形式写入csv中，导致使用不变，可以通过修改查询脚本，将数字转为文本即可。
```

6、数据共享

6.1 系统接口规范

```
# 1）目前工参大屏分析共对接了东方国信（大屏展示呈现-秦亚伟，掌沃建-安全）、亚信（工单流转|定制化报告），其中与掌沃建系统对接之前已经完成，由郝建每月定时通过大数据接口向安全共享地理化栅格覆盖数据；而本次接口规范则描述工参资源课题与东方国信（大屏展示呈现）、国信（工单流转|定制化报告）的互通规范。

# 2）当有新的业务需求或需求变动时，一般有客户牵头联系相关方沟通变化细节，固化约定，由我方制定接口规范，统一按规范进行数据交互。注意：如果是在原有接口中增加字段，需要把字段添加在原有接口末尾。

# 3）接口文件地址：基础工参治理\接口规范\工参治理第二阶段文件接口规范 v11 20210720.xlsx
```

日期	记录	修改人
2021/4/24	本接口主要定义上海诺基亚贝尔对接亚信科技、东方国信的工单数据与大屏数据。	张磊
2021/5/14	新增低业务的45G基站小区接口;增加工参数据判断为空的条件字段说明。	张磊
2021/5/17	增加样本，增加建议索引字段，增加描述字段等说明。基站偏离可视化分析。 接口文件中gcToyxgx_00_bigscreen_XX.csv中，其中XX为日期（数据分析日期），00表示文件接口编号，文件内容及样本在00的sheet中，其它类似。 数据源侧，接口文件按日期目录存储，如/data/bigscreen/20210517/gcToyxgx_00_bigscreen_20180517.csv	张磊
2021/5/20	根据亚信、国信建议在文件接口中，增加地市编码、行政区编码；根据亚信建议，修改程序输出数据为月粒度。为00号文件接口增加注释。	张磊
2021/5/21	按照国信要求，在00号文件中，增加地市编码，并进行对接程序文件更新。增加城市编码表。	张磊
2021/5/25	删除00号文件中，22-5g-基站-7天低业务物理站，5g基站级统一采用物理站；调整文件接口显示为科学记数法的问题（亚信反馈）。	张磊
2021/5/26	00号文件，将地市修改为行政区，地市编码修改为行政区编码；增加2个文件接口，4/5G两个全量工参数据，分别对应文件接口的20、21号文件；	张磊
2021/5/27	增加2个文件接口，4/5G两个全量工参数据，分别对应文件接口的20、21号文件；	张磊
2021/6/1	根据亚信建议，1月有业务的基站扇区、7天低业务或无业务基站扇区、48小时低业务或无业务基站扇区，增加业务量字段（对应01-16号文件）。目前仅更新规范，接口文件更新6.10前完成；将接口中，所有"小区"字符修改为"扇区"，避免市场部与住址小区混淆。修改接口文件19的名称由"4g经纬度偏移>3000扇区数"到"4g经纬度偏移>3000基站数",修改01-16号文件示例模板，增加业务量列字段。	张磊
2021/7/5	原始基础工参数据增加了4个字段【承建方,是否共站址,对应共站址基站ID,共站址运营商】，故17、18、20、21接口需要新增4个字段。	张磊
2021/7/13	按照20210713会议沟通结果，修改"接口详情"sheet中"接口文件内容描述"列，修改内容为：1月->7天，7天->2天，48小时->1天，"4g经纬度偏移>3000基站数"->"4g基站经纬度疑似偏离"，同步修改00号文件中的业务名称列。	张磊
2021/7/20	根据客户建议，需要在大屏展示中区分联通扇区与电信扇区，因此需要在原有文件接口中增加承建方字段，影响的文件接口有00至16，根据约定，新增字段增加在文件接口末尾。根据业务内容修改"sheet接口详情"的接口文件名，使接口文件名更易读，增加一列"新接口文件名称（XX为日期，0720修改更新）"，7.20之后的推送使用新接口文件名。	张磊

6.2 已对接接口文件脚本

```
# 00 --大屏报表数据
    select DATA_TYPE     "业务名称",
           district_name "行政区名称",
           district_code "行政区编码",
           site_num      "小区数量",
           DATA_NUM      "单位数量",
           rate          "百分占比",
           CONSTRUCTION  "承建方"
      from v_STATS_BIGSCREEN_SUMMARY_32

```

```
# 01 1月有业务的4g基站数
    select a.CITY_NAME "地市名称",
           a.DISTRICT_NAME "行政区名称",
           a.CITY_CODE "地市编码",
           a.DISTRICT_CODE "行政区编码",
           a.PHYSTATION_ADDRESS "物理站名称",
           a.BBU_NAME "BBU名称",
           to_char(a.ENBID) "基站enbid",
           a.vender "厂家",
           round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --THROUGHPUT 单位是byte，需要除以1024*1024*1024
           ,
           CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            vender,
                            first_value(construction) over(partition by district_name, district_code, enbid) construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, THROUGHPUT
              from STATS_4g_SITE_FLOW_2M_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_SITE_FLOW_2M_DAY)) b
     where /*a.city_name=b.city_name and*/
     a.enbid = b.enbid

# 02 1月有业务的4g小区数
    select distinct a.CITY_NAME "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME "BBU名称",
                    to_char(a.ENBID) "基站enbid",
                    a.cell_id "小区id",
                    a.lon "小区经度",
                    a.lat "小区纬度",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --THROUGHPUT 单位是byte，需要除以1024*1024*1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            cell_id,
                            lon,
```

```
                            lat,
                            vender,
                            construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)) a,
           (select distinct city_name, enbid, cell_id, THROUGHPUT
              from STATS_4g_CELL_FLOW_2M_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_CELL_FLOW_2M_DAY)) b
     where /*a.city_name=b.city_name and*/
     a.enbid = b.enbid
     and a.cell_id = b.cell_id

# 03 1月有业务的5g基站数
    select distinct a.city_name "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME "基站名称",
                    to_char(a.gnbid) "基站GNBID",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位是Mbyte，需要除以1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            vender,
                            first_value(construction) over(partition by district_name, district_code, gnbid) construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, PHYSTATION_ADDRESS, gnbid, THROUGHPUT
              from STATS_5g_PHYSITE_FLOW_2M_DAY
             where sdate =
                   (select max(sdate) from STATS_5g_PHYSITE_FLOW_2M_DAY)) b
     where /* a.city_name=b.city_name and*/
     a.gnbid = b.gnbid
     and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS

# 04 1月有业务的5g小区数
    select distinct a.city_name "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME "基站名称",
                    to_char(a.gnbid) "基站GNBID",
                    a.cell_name "小区名称",
                    a.cell_id "小区id",
```

```
                    a.lon "小区经度",
                    a.lat "小区纬度",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位是Mbyte，需要除以1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            cell_name,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, cell_id, gnbid, THROUGHPUT
              from STATS_5g_CELL_FLOW_2M_DAY
             where sdate =
                   (select max(sdate) from STATS_5g_CELL_FLOW_2M_DAY)) b
     where /*a.city_name=b.city_name and */
     a.gnbid = b.gnbid
     and a.cell_id = b.cell_id

# 05 7天无业务4g基站数
    select distinct a.CITY_NAME "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME "BBU名称",
                    to_char(a.ENBID) "基站enbid",
                    a.vender "厂家",
                    0 "业务量",
                    construction "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            vender,
                            first_value(construction) over(partition by district_name, district_code, enbid) construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid
              from STATS_4g_SITE_FLOW_7d_DAY
             where sdate =
```

```
                  (select max(sdate) from STATS_4g_SITE_FLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid

# 06 7天无业务4g小区数
    select distinct a.CITY_NAME          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME           "BBU名称",
                    to_char(a.ENBID)             "基站enbid",
                    a.cell_id            "小区id",
                    a.lon                "小区经度",
                    a.lat                "小区纬度",
                    a.vender             "厂家",
                    0 "业务量",
                    construction "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, cell_id
              from STATS_4g_CELL_FLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_CELL_FLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid
     and a.cell_id = b.cell_id

# 07 7天低业务4g基站数
    select distinct a.CITY_NAME "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME "BBU名称",
                    to_char(a.ENBID) "基站enbid",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte，需要除以1024*1024*1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
```

```
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            vender,
                            first_value(construction) over(partition by
district_name, district_code, enbid) construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, THROUGHPUT
              from STATS_4g_SITE_lowFLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_SITE_lowFLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid

# 08 7天低业务4g小区数
    select distinct a.CITY_NAME "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME "BBU名称",
                    to_char(a.ENBID) "基站enbid",
                    a.cell_id "小区id",
                    a.lon "小区经度",
                    a.lat "小区纬度",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte，需要除以1024*1024*1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, cell_id, THROUGHPUT
              from STATS_4g_CELL_lowFLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_CELL_lowFLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid
     and a.cell_id = b.cell_id

# 09 7天无业务5g基站数 4492
    select distinct a.city_name          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
```

```
                    a.CITY_CODE           "地市编码",
                    a.DISTRICT_CODE       "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME        "基站名称",
                    to_char(a.gnbid)              "基站GNBID",
                    a.vender              "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            vender,
                            first_value(construction) over(partition by
district_name, district_code, gnbid) construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, PHYSTATION_ADDRESS, GNBID
              from STATS_5g_phySITE_FLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_5g_phySITE_FLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
     and a.GNBID = b.GNBID

# 10 7天无业务5g小区数
    select distinct a.city_name           "地市名称",
                    a.DISTRICT_NAME       "行政区名称",
                    a.CITY_CODE           "地市编码",
                    a.DISTRICT_CODE       "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME        "基站名称",
                    to_char(a.gnbid)              "基站GNBID",
                    a.cell_name           "小区名称",
                    a.cell_id             "小区id",
                    a.lon                 "小区经度",
                    a.lat                 "小区纬度",
                    a.vender              "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            cell_name,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_5g_siteinfo
```

```
            where sdate = (select max(sdate) from seq_5g_siteinfo)
              and is_alive = 1) a,
          (select distinct city_name, GNBID, cell_id
             from STATS_5g_CELL_FLOW_7d_DAY
            where sdate =
                  (select max(sdate) from STATS_5g_CELL_FLOW_7d_DAY)) b
    where /*a.city_name= b.city_name and*/
    a.GNBID = b.GNBID
    and a.cell_id = b.cell_id

# 11 7天低业务5g基站数
   select distinct a.city_name "地市名称",
                   a.DISTRICT_NAME "行政区名称",
                   a.CITY_CODE "地市编码",
                   a.DISTRICT_CODE "行政区编码",
                   a.PHYSTATION_ADDRESS "物理站名称",
                   a.STATION_NAME "基站名称",
                   to_char(a.gnbid) "基站GNBID",
                   a.vender "厂家",
                   round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位是Mbyte，需要除以1024,
                   ,
                   CONSTRUCTION "承建方"
     from (select distinct city_name,
                           DISTRICT_NAME,
                           CITY_CODE,
                           DISTRICT_CODE,
                           PHYSTATION_ADDRESS,
                           STATION_NAME,
                           gnbid,
                           vender,
                           first_value(construction) over(partition by district_name, district_code, gnbid) construction
             from seq_5g_siteinfo
            where sdate = (select max(sdate) from seq_5g_siteinfo)
              and is_alive = 1) a,
          (select distinct city_name, PHYSTATION_ADDRESS, GNBID, THROUGHPUT
             from STATS_5g_PHYSITE_LOFLOW_7d_DAY
            where sdate =
                  (select max(sdate) from STATS_5g_PHYSITE_LOFLOW_7d_DAY)) b
    where /*a.city_name= b.city_name and*/
    a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
    and a.GNBID = b.GNBID

# 12 7天低业务5g小区数
   select distinct a.city_name "地市名称",
                   a.DISTRICT_NAME "行政区名称",
                   a.CITY_CODE "地市编码",
                   a.DISTRICT_CODE "行政区编码",
                   a.PHYSTATION_ADDRESS "物理站名称",
                   a.STATION_NAME "基站名称",
                   to_char(a.gnbid) "基站GNBID",
                   a.cell_name "小区名称",
                   a.cell_id "小区id",
                   a.lon "小区经度",
                   a.lat "小区纬度",
```

```
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位是Mbyte，需要除以1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            cell_name,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, GNBID, cell_id, THROUGHPUT
              from STATS_5g_CELL_LOWFLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_5g_CELL_LOWFLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.GNBID = b.GNBID
     and a.cell_id = b.cell_id

# 13 48小时无业务4g基站数
    select distinct a.CITY_NAME          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME           "BBU名称",
                    to_char(a.ENBID)              "基站enbid",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            vender,
                            first_value(construction) over(partition by district_name, district_code, enbid) construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid
              from STATS_4g_SITE_FLOW_48h_hour
             where sdate =
                   (select max(sdate) from STATS_4g_SITE_FLOW_48h_hour)) b
     where /*a.city_name= b.city_name and*/
```

```
      a.enbid = b.enbid

# 14 48小时无业务4g小区数
    select distinct a.CITY_NAME          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME           "BBU名称",
                    to_char(a.ENBID)             "基站enbid",
                    a.cell_id            "小区id",
                    a.lon                "小区经度",
                    a.lat                "小区纬度",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, cell_id
              from STATS_4g_CELL_FLOW_48h_hour
             where sdate =
                   (select max(sdate) from STATS_4g_CELL_FLOW_48h_hour)) b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid
     and a.cell_id = b.cell_id

# 15 48小时无业务5g基站数
    select distinct a.city_name          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME       "基站名称",
                    to_char(a.gnbid)             "基站GNBID",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            vender,
```

```
                            first_value(construction) over(partition by
district_name, district_code, gnbid) construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, PHYSTATION_ADDRESS, GNBID
              from STATS_5g_phySITE_FLOW_48h_hour
             where sdate =
                   (select max(sdate) from STATS_5g_phySITE_FLOW_48h_hour))
b
     where /*a.city_name= b.city_name and*/
     a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
     and a.GNBID = b.GNBID

# 16 48小时无业务5g小区数
    select distinct a.city_name          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME       "基站名称",
                    to_char(a.gnbid)             "基站GNBID",
                    a.cell_name          "小区名称",
                    a.cell_id            "小区id",
                    a.lon                "小区经度",
                    a.lat                "小区纬度",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            cell_name,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, GNBID, cell_id
              from STATS_5g_CELL_FLOW_48h_hour
             where sdate =
                   (select max(sdate) from STATS_5g_CELL_FLOW_48h_hour)) b
     where /*a.city_name = b.city_name
       and*/ a.GNBID = b.GNBID
       and a.cell_id = b.cell_id

# 17 4g小区错误或字段不全
    select distinct ERROR_CLASS        "异常类型",
                    CITY_NAME          "城市名称",
                    DISTRICT_NAME      "行政区名称",
                    CITY_CODE          "城市编码",
```

```
                    DISTRICT_CODE      "行政区编码",
                    NETWORK_NAME       "网络类型",
                    PHYSTATION_ADDRESS "物理站名称",
                    BBU_NAME           "BBU名称",
                    to_char(ENBID)              "基站ENBID",
                    CELL_NAME          "小区名称",
                    CELL_ID            "小区id",
                    LON                "小区经度",
                    LAT                "小区纬度",
                    DIRECTION          "小区方位角",
                    HEIGHT             "天线挂高",
                    M_DOWNTILT         "机械下倾",
                    E_DOWNTILT         "电子下倾",
                    STATION_TYPE       "基站类型",
                    ISDIGITALINDOOR    "是否电子室分",
                    DOWN_FREQ          "下行频点",
                    VENDER             "厂家",
                    OWN_SCHOOLYARD     "是否校园站",
                    TOWERADDRESS_CODE  "铁塔站址编号",
                    PROPERTY           "产权归属",
                    SCENE              "场景",
                    IS_SCENESITE       "是否场景站",
                    MARKETING_NETWORK  "销售网络",
                    TERMINALAMOUNT_5G  "终端数量5G",
                    SECTOR_INCOMING    "扇区收入",
                    IS_BUSY            "是否超忙",
                    CONSTRUCTION       "承建方",
                    IS_SAME_ADDRESS    "是否共站址",
                    SAME_ADDRESS_SITES "对应共站址基站ID",
                    IS_SAME_ADDRESS_CT "共站址运营商"
      from STATS_4G_ERROR_SITEINFO
     where sdate = (select max(sdate) from STATS_4G_ERROR_SITEINFO)
       and is_alive = 1

# 18 5g小区错误或字段不全
    select distinct ERROR_CLASS "异常类型",
                    CITY_NAME "城市名称",
                    DISTRICT_NAME "行政区名称",
                    CITY_CODE "城市编码",
                    DISTRICT_CODE "行政区编码",
                    NETWORK_NAME "网络类型",
                    PHYSTATION_ADDRESS "物理站名称",
                    STATION_NAME "基站名称",
                    to_char(GNBID) "基站GNBID",
                    CELL_NAME "小区名称",
                    CELL_ID "小区id",
                    LON "小区经度",
                    LAT "小区纬度",
                    DIRECTION "方位角",
                    HEIGHT "天线挂高",
                    M_DOWNTILT "机械下倾",
                    E_DOWNTILT "电子下倾",
                    STATION_TYPE "基站类型",
                    ISDIGITALINDOOR "是否数字化室分",
                    DOWN_FREQ "下行频点",
                    VENDER "厂家",
                    OWN_SCHOOLYARD "是否校园站",
                    TOWERADDRESS_CODE "铁塔地址编码",
```

```
                    PROPERTY "产权归属",
                    SCENE "场景",
                    IS_SCENESITE "是否场景站",
                    MARKETING_NETWORK "销售网格",
                    TERMINALAMOUNT_5G "终端数量5G",
                    SECTOR_INCOMING "扇区收入",
                    IS_BUSY "是否超忙",
                    CONSTRUCTION "承建方",
                    IS_SAME_ADDRESS "是否共站址",
                    SAME_ADDRESS_SITES "对应共站址基站ID",
                    IS_SAME_ADDRESS_CT "共站址运营商"
      from STATS_5G_ERROR_SITEINFO
     where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
       and is_alive = 1

# 19 4g经纬度偏移>3000小区数
    select CITY_NAME           "城市名称",
           DISTRICT_NAME       "行政区名称",
           CITY_CODE           "城市编码",
           DISTRICT_CODE       "行政区编码",
           PHYSTATION_ADDRESS "物理站址",
           BBU_NAME            "BBU名称",
           to_char(ENBID)              "基站enbid",
           STATION_TYPE        "基站类型",
           VENDER              "设备厂家",
           SCENE               "覆盖厂家",
           DEVIATE_DISTANCE    "基站偏离距离",
           C1_ECI              "小区1_eci",
           C1_CELL_NAME        "小区1_小区名称",
           C1_DISTANCE         "小区1_偏离距离",
           C2_ECI              "小区2_eci",
           C2_CELL_NAME        "小区2_小区名称",
           C2_DISTANCE         "小区2_偏离距离",
           C3_ECI              "小区3_eci",
           C3_CELL_NAME        "小区3_小区名称",
           C3_DISTANCE         "小区3_偏离距离",
           C4_ECI              "小区4_eci",
           C4_CELL_NAME        "小区4_小区名称",
           C4_DISTANCE         "小区4_偏离距离",
           C5_ECI              "小区5_eci",
           C5_CELL_NAME        "小区5_小区名称",
           C5_DISTANCE         "小区5_偏离距离",
           C6_ECI              "小区6_eci",
           C6_CELL_NAME        "小区6_小区名称",
           C6_DISTANCE         "小区6_偏离距离",
           C7_ECI              "小区7_eci",
           C7_CELL_NAME        "小区7_小区名称",
           C7_DISTANCE         "小区7_偏离距离",
           C8_ECI              "小区8_eci",
           C8_CELL_NAME        "小区8_小区名称",
           C8_DISTANCE         "小区8_偏离距离",
           C9_ECI              "小区9_eci",
           C9_CELL_NAME        "小区9_小区名称",
           C9_DISTANCE         "小区9_偏离距离",
           CONSTRUCTION        "承建方"
      from stats_4G_DEVIATE_SUMMARY_DB
      where DEVIATE_DISTANCE>3000

```

```
# 20 导出4g工参数据
    SELECT SDATE "数据日期",
           CITY_CODE "城市编码",
           CITY_NAME "城市名称",
           DISTRICT_CODE "行政区编码",
           DISTRICT_NAME "行政区名称",
           NETWORK_NAME "网络类型",
           PHYSTATION_ADDRESS "物理站名称",
           BBU_NAME "BBU名称",
           to_char(ENBID) "基站ENBID",
           CELL_NAME "小区名称",
           CELL_ID "小区id",
           to_char('46001' || enbid * 256 + cell_id) "小区全球唯一标识",
           LON "小区经度",
           LAT "小区纬度",
           DIRECTION "小区方位角",
           HEIGHT "天线挂高",
           M_DOWNTILT "机械下倾",
           E_DOWNTILT "电子下倾",
           STATION_TYPE "基站类型",
           ISDIGITALINDOOR "是否电子室分",
           DOWN_FREQ "下行频点",
           VENDER "厂家",
           OWN_SCHOOLYARD "是否校园站",
           TOWERADDRESS_CODE "铁塔站址编号",
           PROPERTY "产权归属",
           SCENE "场景",
           IS_SCENESITE "是否场景站",
           MARKETING_NETWORK "销售网络",
           TERMINALAMOUNT_5G "终端数量5G",
           SECTOR_INCOMING "扇区收入",
           IS_BUSY "是否超忙",
           IS_ALIVE "是否有业务",
           IS_ALIVE_UPDATE_TIME "业务更新时间",
           CONSTRUCTION "承建方",
           IS_SAME_ADDRESS "是否共站址",
           SAME_ADDRESS_SITES "对应共站址基站ID",
           IS_SAME_ADDRESS_CT "共站址运营商"
      FROM seq_4g_siteinfo
     where sdate = (select max(sdate) from seq_4g_siteinfo)
       and is_alive = 1


# 21 导出5g工参数据
    SELECT SDATE "数据日期",
           CITY_CODE "城市编码",
           CITY_NAME "城市名称",
           DISTRICT_CODE "行政区编码",
           DISTRICT_NAME "行政区名称",
           NETWORK_NAME "网络类型",
           PHYSTATION_ADDRESS "物理站名称",
           STATION_NAME "基站名称",
           to_char(GNBID) "基站GNBID",
           CELL_NAME "小区名称",
           CELL_ID "小区id",
           to_char('46001' || gnbid * 4096 + cell_id) "小区全球唯一识别码",
           LON "小区经度",
           LAT "小区纬度",
```

```
            DIRECTION "方位角",
            HEIGHT "天线挂高",
            M_DOWNTILT "机械下倾",
            E_DOWNTILT "电子下倾",
            STATION_TYPE "基站类型",
            ISDIGITALINDOOR "是否数字化室分",
            DOWN_FREQ "下行频点",
            VENDER "厂家",
            OWN_SCHOOLYARD "是否校园站",
            TOWERADDRESS_CODE "铁塔地址编码",
            PROPERTY "产权归属",
            SCENE "场景",
            IS_SCENESITE "是否场景站",
            MARKETING_NETWORK "销售网格",
            TERMINALAMOUNT_5G "终端数量5G",
            SECTOR_INCOMING "扇区收入",
            IS_BUSY "是否超忙",
            IS_ALIVE "是否有业务",
            IS_ALIVE_UPDATE_TIME "业务更新时间",
            CONSTRUCTION "承建方",
            IS_SAME_ADDRESS "是否共站址",
            SAME_ADDRESS_SITES "对应共站址基站ID",
            IS_SAME_ADDRESS_CT "共站址运营商"
       FROM seq_5g_siteinfo
      where sdate = (select max(sdate) from seq_5g_siteinfo)
        and is_alive = 1
```

6.3 接口程序及源码

```
# 开发语言：接口对接程序采用go语言编写；
# 程序功能：实现接口规范中21个接口文件的一键式导出与上传接口数据到ESB接口服务器。当有新需求或接口文件变更时，需要重新修改接口程序，编译后更新升级。

# 程序执行后的后续三方流程：数据上传ESB接口服务器(192.168.0.41)-->ESB根据文件名扫描文件变化采集数据-->推送东方国信/亚信。说明：ESB仅根据文件名进行同步，所以当文件名没有变化而仅是内容变化，ESB并不会同步数据。

# 程序使用方法：将编译好的可执行程序放到133.96.92.137服务器任意目录，双击执行即可；执行中及完成可观察当前目录日志文件判断是否正常；可登录ESB接口服务器(192.168.0.41)判断数据是否都正常。

# 注意事项：go语言在将数字写入csv文件时，如果数字大于6位数，会以科学记数法形式写入，解决办法是在数据查询脚本使用to_char()函数将数字转为文本即可。
```

```
package main

import (
    "database/sql"
    "encoding/csv"
    "fmt"
    "io"
    "log"
    "os"
    "path"
    "time"

```

```
    "github.com/jlaffaye/ftp"
    _ "github.com/mattn/go-oci8"
)

var logger *log.Logger
var file *os.File
var err error

func init() {
    file, err = os.OpenFile("test.log",
os.O_APPEND|os.O_CREATE|os.O_WRONLY, 666)
    if err != nil {
        log.Fatal(err)
    }

    logger = log.New(file, "", log.LstdFlags)
    logger.SetPrefix("Test- ") // 设置日志前缀
    logger.SetFlags(log.Ldate | log.Ltime | log.Lshortfile)
    /*
        const (
            // 字位共同控制输出日志信息的细节。不能控制输出的顺序和格式。
            // 在所有项目后会有一个冒号：2009/01/23 01:23:23.123123
/a/b/c/d.go:23: message
            Ldate         = 1 << iota     // 日期：2009/01/23
            Ltime                         // 时间：01:23:23
            Lmicroseconds                 // 微秒分辨率：01:23:23.123123（用
于增强Ltime位）
            Llongfile                     // 文件全路径名+行号：
/a/b/c/d.go:23
            Lshortfile                    // 文件无路径名+行号：d.go:23（会覆
盖掉Llongfile）
            LstdFlags     = Ldate | Ltime // 标准logger的初始值
        )
    */
}

func sqlExec(db *sql.DB, sqlStmt string) error {
    res, err := db.Exec(sqlStmt)
    if err != nil {
        logger.Fatal("sqlExec执行失败:" + err.Error())
    }

    num, err := res.RowsAffected()
    if err != nil {
        logger.Fatal("获取查询影响行数失败:" + err.Error())
    }

    logger.Printf("SQL Execute success rows affected %d\n", num)
    return nil
}

func sqlQuery2Csv(db *sql.DB, sqlStmt string, newFileName string) error {
    rows, err := db.Query(sqlStmt)
    if err != nil {
        logger.Fatal("sqlQuery执行失败,err: %s,err sql:%s", err, sqlStmt)
    }
    defer rows.Close()
    //返回所有列
```

```
    cols, _ := rows.Columns()
    //这里表示一行所有列的值，用[]byte表示
    vals := make([][]byte, len(cols))
    //这里表示一行填充数据
    scans := make([]interface{}, len(cols))
    //这里scans引用vals，把数据填充到[]byte里
    for k, _ := range vals {
        scans[k] = &vals[k]
    }
    // 统计行数
    i := 0
    //result := make(map[int]map[string]string)
    //这样可以追加写
    nfs, err := os.OpenFile(newFileName, os.O_RDWR|os.O_CREATE|os.O_TRUNC,
0666)
    if err != nil {
        logger.Fatalf("can not create file, err is %+v", err)
    }
    defer nfs.Close()

    nfs.Seek(0, io.SeekEnd)

    w := csv.NewWriter(nfs)

    //设置属性
    w.Comma = ','
    w.UseCRLF = true
    //row := []string{"1", "2", "3", "4", "5,6"}
    // 写入文件
    err = w.Write(cols)

    for rows.Next() {
        //填充数据
        rows.Scan(scans...)
        //每行数据
        var row []string
        //把vals中的数据复制到row中
        for _, v := range vals {
            // key := cols[k]
            //这里把[]byte数据转成string
            row = append(row, string(v))
        }
        //放入结果集
        //result[i] = row;
        err = w.Write(row)
        i++
    }
    w.Flush()

    if err != nil {
        logger.Fatal("写入csv失败", err)
    }

    // logger.Printf("SQL Query success rows queried %d\n", i)
    logger.Printf("查询结果导出csv成功："+newFileName+"，行数：%d\n", i)
    return nil
}

```

```go
//判断文件或文件夹是否存在
func isExist(path string) bool {
    _, err := os.Stat(path)
    if err != nil {
        if os.IsExist(err) {
            return true
        }
        if os.IsNotExist(err) {
            return false
        }
        return false
    }
    return true
}

func main() {
    // 20210520 修改文件夹及文件名日期为月粒度
    dateDir := time.Now().Format("20060102")
    // dateDir := time.Now().Format("200601")
    fileDir := "./" + dateDir
    if !isExist(fileDir) {
        os.Mkdir(fileDir, 0766)
    }
    db, err := sql.Open("oci8", fmt.Sprintf("%s/%s@%s", "c##fast491",
"F@st491*321", "192.168.0.64:1521/fast"))
    if err != nil {
        logger.Fatal("oracle登录失败:" + err.Error())
    }
    defer db.Close()
    logger.Printf("数据库已登录")

    err = db.Ping()
    if err != nil {
        logger.Fatal("oracle链接不可达:" + err.Error())
    }

    waitFiles := make([]string, 0, 5)

    logger.Printf("查询结果导出csv")

    // 导出文件-0 大屏展示数据
    fileName := "gcToyxgx_00_bigscreen_" + dateDir + ".csv"
    filepath := fileDir + "/" + fileName
    exportSql := `
    select DATA_TYPE     "业务名称",
           district_name "行政区名称",
           district_code "行政区编码",
           site_num      "小区数量",
           DATA_NUM      "单位数量",
           rate          "百分占比",
           CONSTRUCTION  "承建方"
      from v_STATS_BIGSCREEN_SUMMARY_32
    `
    // sql末尾不能有分号
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
```

```
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-1 7天有业务的4g基站数
    fileName = "gcToyxgx_01_4g_site_youyewu_7d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select a.CITY_NAME "地市名称",
           a.DISTRICT_NAME "行政区名称",
           a.CITY_CODE "地市编码",
           a.DISTRICT_CODE "行政区编码",
           a.PHYSTATION_ADDRESS "物理站名称",
           a.BBU_NAME "BBU名称",
           to_char(a.ENBID) "基站enbid",
           a.vender "厂家",
           round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte，需要除以1024*1024*1024
           ,
           CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            vender,
                            first_value(construction) over(partition by
district_name, district_code, enbid) construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, THROUGHPUT
              from STATS_4g_SITE_FLOW_2M_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_SITE_FLOW_2M_DAY)) b
     where /*a.city_name=b.city_name and*/
     a.enbid = b.enbid
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }
    // 导出文件-2 7天有业务的4g小区数
    fileName = "gcToyxgx_02_4g_cell_youyewu_7d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.CITY_NAME "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
```

```
                    a.BBU_NAME "BBU名称",
                    to_char(a.ENBID) "基站enbid",
                    a.cell_id "小区id",
                    a.lon "小区经度",
                    a.lat "小区纬度",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --THROUGHPUT 单位是byte，需要除以1024*1024*1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)) a,
           (select distinct city_name, enbid, cell_id, THROUGHPUT
              from STATS_4g_CELL_FLOW_2M_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_CELL_FLOW_2M_DAY)) b
     where /*a.city_name=b.city_name and*/
     a.enbid = b.enbid
     and a.cell_id = b.cell_id
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }
    // 导出文件-3 7天有业务的5g基站数
    fileName = "gcToyxgx_03_5g_site_youyewu_7d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.city_name "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME "基站名称",
                    to_char(a.gnbid) "基站GNBID",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位是Mbyte，需要除以1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
```

```
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            vender,
                            first_value(construction) over(partition by
district_name, district_code, gnbid) construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, PHYSTATION_ADDRESS, gnbid,
THROUGHPUT
              from STATS_5g_PHYSITE_FLOW_2M_DAY
             where sdate =
                   (select max(sdate) from STATS_5g_PHYSITE_FLOW_2M_DAY))
b
     where /* a.city_name=b.city_name and*/
     a.gnbid = b.gnbid
     and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }
    // 导出文件-4 7天有业务的5g小区数
    fileName = "gcToyxgx_04_5g_cell_youyewu_7d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.city_name "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME "基站名称",
                    to_char(a.gnbid) "基站GNBID",
                    a.cell_name "小区名称",
                    a.cell_id "小区id",
                    a.lon "小区经度",
                    a.lat "小区纬度",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单
位是Mbyte，需要除以1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            cell_name,
                            cell_id,
```

```
                            lon,
                            lat,
                            vender,
                            construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, cell_id, gnbid, THROUGHPUT
              from STATS_5g_CELL_FLOW_2M_DAY
             where sdate =
                   (select max(sdate) from STATS_5g_CELL_FLOW_2M_DAY)) b
     where /*a.city_name=b.city_name and */
     a.gnbid = b.gnbid
     and a.cell_id = b.cell_id
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }
    // 导出文件-5 7天无业务4g基站数
    fileName = "gcToyxgx_05_4g_site_wuyewu_2d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.CITY_NAME "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME "BBU名称",
                    to_char(a.ENBID) "基站enbid",
                    a.vender "厂家",
                    0 "业务量",
                    construction "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            vender,
                            first_value(construction) over(partition by
district_name, district_code, enbid) construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid
              from STATS_4g_SITE_FLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_SITE_FLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
```

```
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }
    // 导出文件-6 7天无业务4g小区数
    fileName = "gcToyxgx_06_4g_cell_wuyewu_2d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.CITY_NAME          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME           "BBU名称",
                    to_char(a.ENBID)             "基站enbid",
                    a.cell_id            "小区id",
                    a.lon                "小区经度",
                    a.lat                "小区纬度",
                    a.vender             "厂家",
                    0 "业务量",
                    construction "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, cell_id
              from STATS_4g_CELL_FLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_CELL_FLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid
     and a.cell_id = b.cell_id
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }
    // 导出文件-7 7天低业务4g基站数
    fileName = "gcToyxgx_07_4g_site_diyewu_2d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
```

```
    exportSql = `
    select distinct a.CITY_NAME "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME "BBU名称",
                    to_char(a.ENBID) "基站enbid",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte，需要除以1024*1024*1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            vender,
                            first_value(construction) over(partition by
district_name, district_code, enbid) construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, THROUGHPUT
              from STATS_4g_SITE_lowFLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_SITE_lowFLOW_7d_DAY))
b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-8 7天低业务4g小区数
    fileName = "gcToyxgx_08_4g_cell_diyewu_2d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.CITY_NAME "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME "BBU名称",
                    to_char(a.ENBID) "基站enbid",
                    a.cell_id "小区id",
                    a.lon "小区经度",
                    a.lat "小区纬度",
                    a.vender "厂家",
```

```
                    round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte，需要除以1024*1024*1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, cell_id, THROUGHPUT
              from STATS_4g_CELL_lowFLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_4g_CELL_lowFLOW_7d_DAY))
b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid
     and a.cell_id = b.cell_id
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-9 7天无业务5g基站数
    fileName = "gcToyxgx_09_5g_site_wuyewu_2d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.city_name          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME       "基站名称",
                    to_char(a.gnbid)             "基站GNBID",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
```

```
                            vender,
                            first_value(construction) over(partition by district_name, district_code, gnbid) construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, PHYSTATION_ADDRESS, GNBID
              from STATS_5g_phySITE_FLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_5g_phySITE_FLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
     and a.GNBID = b.GNBID
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-10 7天无业务5g小区数
    fileName = "gcToyxgx_10_5g_cell_wuyewu_2d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.city_name          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME       "基站名称",
                    to_char(a.gnbid)              "基站GNBID",
                    a.cell_name          "小区名称",
                    a.cell_id            "小区id",
                    a.lon                "小区经度",
                    a.lat                "小区纬度",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            cell_name,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
```

```
628             (select distinct city_name, GNBID, cell_id
629                from STATS_5g_CELL_FLOW_7d_DAY
630               where sdate =
631                     (select max(sdate) from STATS_5g_CELL_FLOW_7d_DAY)) b
632      where /*a.city_name= b.city_name and*/
633      a.GNBID = b.GNBID
634      and a.cell_id = b.cell_id
635     `
636     err = sqlQuery2Csv(db, exportSql, filepath)
637     if err != nil {
638         logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
639     } else {
640         if isExist(filepath) {
641             waitFiles = append(waitFiles, filepath)
642         }
643     }
644
645     // 导出文件-11 7天低业务5g物理基站数
646     fileName = "gcToyxgx_11_5g_site_diyewu_2d_" + dateDir + ".csv"
647     filepath = fileDir + "/" + fileName
648     exportSql = `
649     select distinct a.city_name "地市名称",
650                     a.DISTRICT_NAME "行政区名称",
651                     a.CITY_CODE "地市编码",
652                     a.DISTRICT_CODE "行政区编码",
653                     a.PHYSTATION_ADDRESS "物理站名称",
654                     a.STATION_NAME "基站名称",
655                     to_char(a.gnbid) "基站GNBID",
656                     a.vender "厂家",
657                     round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单
位是Mbyte，需要除以1024,
658                     ,
659                     CONSTRUCTION "承建方"
660       from (select distinct city_name,
661                             DISTRICT_NAME,
662                             CITY_CODE,
663                             DISTRICT_CODE,
664                             PHYSTATION_ADDRESS,
665                             STATION_NAME,
666                             gnbid,
667                             vender,
668                             first_value(construction) over(partition by
district_name, district_code, gnbid) construction
669               from seq_5g_siteinfo
670              where sdate = (select max(sdate) from seq_5g_siteinfo)
671                and is_alive = 1) a,
672            (select distinct city_name, PHYSTATION_ADDRESS, GNBID,
THROUGHPUT
673               from STATS_5g_PHYSITE_LOFLOW_7d_DAY
674              where sdate =
675                    (select max(sdate) from
STATS_5g_PHYSITE_LOFLOW_7d_DAY)) b
676      where /*a.city_name= b.city_name and*/
677      a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
678      and a.GNBID = b.GNBID
679     `
680     err = sqlQuery2Csv(db, exportSql, filepath)
681     if err != nil {
```

```
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-12 7天低业务5g小区数
    fileName = "gcToyxgx_12_5g_cell_diyewu_2d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.city_name "地市名称",
                    a.DISTRICT_NAME "行政区名称",
                    a.CITY_CODE "地市编码",
                    a.DISTRICT_CODE "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME "基站名称",
                    to_char(a.gnbid) "基站GNBID",
                    a.cell_name "小区名称",
                    a.cell_id "小区id",
                    a.lon "小区经度",
                    a.lat "小区纬度",
                    a.vender "厂家",
                    round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位是Mbyte，需要除以1024
                    ,
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            cell_name,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, GNBID, cell_id, THROUGHPUT
              from STATS_5g_CELL_LOWFLOW_7d_DAY
             where sdate =
                   (select max(sdate) from STATS_5g_CELL_LOWFLOW_7d_DAY)) b
     where /*a.city_name= b.city_name and*/
     a.GNBID = b.GNBID
     and a.cell_id = b.cell_id
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
```

```
        }
    }

    // 导出文件-13 48小时无业务4g基站数
    fileName = "gcToyxgx_13_4g_site_wuyewu_1d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.CITY_NAME          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME           "BBU名称",
                    to_char(a.ENBID)              "基站enbid",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            vender,
                            first_value(construction) over(partition by
district_name, district_code, enbid) construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid
              from STATS_4g_SITE_FLOW_48h_hour
             where sdate =
                   (select max(sdate) from STATS_4g_SITE_FLOW_48h_hour)) b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-14 48小时无业务4g小区数
    fileName = "gcToyxgx_14_4g_cell_wuyewu_1d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.CITY_NAME          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.BBU_NAME           "BBU名称",
                    to_char(a.ENBID)              "基站enbid",
                    a.cell_id            "小区id",
```

```
                    a.lon                "小区经度",
                    a.lat                "小区纬度",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct CITY_NAME,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            BBU_NAME,
                            ENBID,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_4g_siteinfo
             where sdate = (select max(sdate) from seq_4g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, enbid, cell_id
              from STATS_4g_CELL_FLOW_48h_hour
             where sdate =
                   (select max(sdate) from STATS_4g_CELL_FLOW_48h_hour)) b
     where /*a.city_name= b.city_name and*/
     a.enbid = b.enbid
     and a.cell_id = b.cell_id
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-15 48小时无业务5g基站数
    fileName = "gcToyxgx_15_5g_site_wuyewu_1d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.city_name          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME       "基站名称",
                    to_char(a.gnbid)              "基站GNBID",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
```

```
                            vender,
                            first_value(construction) over(partition by district_name, district_code, gnbid) construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
           (select distinct city_name, PHYSTATION_ADDRESS, GNBID
              from STATS_5g_phySITE_FLOW_48h_hour
             where sdate =
                   (select max(sdate) from STATS_5g_phySITE_FLOW_48h_hour)) b
     where /*a.city_name= b.city_name and*/
     a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
     and a.GNBID = b.GNBID
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-16 48小时无业务5g小区数
    fileName = "gcToyxgx_16_5g_cell_wuyewu_1d_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct a.city_name          "地市名称",
                    a.DISTRICT_NAME      "行政区名称",
                    a.CITY_CODE          "地市编码",
                    a.DISTRICT_CODE      "行政区编码",
                    a.PHYSTATION_ADDRESS "物理站名称",
                    a.STATION_NAME       "基站名称",
                    to_char(a.gnbid)              "基站GNBID",
                    a.cell_name          "小区名称",
                    a.cell_id            "小区id",
                    a.lon                "小区经度",
                    a.lat                "小区纬度",
                    a.vender             "厂家",
                    0 "业务量",
                    CONSTRUCTION "承建方"
      from (select distinct city_name,
                            DISTRICT_NAME,
                            CITY_CODE,
                            DISTRICT_CODE,
                            PHYSTATION_ADDRESS,
                            STATION_NAME,
                            gnbid,
                            cell_name,
                            cell_id,
                            lon,
                            lat,
                            vender,
                            construction
              from seq_5g_siteinfo
             where sdate = (select max(sdate) from seq_5g_siteinfo)
               and is_alive = 1) a,
```

```
            (select distinct city_name, GNBID, cell_id
               from STATS_5g_CELL_FLOW_48h_hour
              where sdate =
                    (select max(sdate) from STATS_5g_CELL_FLOW_48h_hour)) b
     where /*a.city_name = b.city_name
       and*/ a.GNBID = b.GNBID
       and a.cell_id = b.cell_id
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-17 4g小区错误或字段不全
    fileName = "gcToyxgx_17_4g_cell_errors_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct ERROR_CLASS         "异常类型",
                    CITY_NAME           "城市名称",
                    DISTRICT_NAME       "行政区名称",
                    CITY_CODE           "城市编码",
                    DISTRICT_CODE       "行政区编码",
                    NETWORK_NAME        "网络类型",
                    PHYSTATION_ADDRESS "物理站名称",
                    BBU_NAME            "BBU名称",
                    to_char(ENBID)              "基站ENBID",
                    CELL_NAME           "小区名称",
                    CELL_ID             "小区id",
                    LON                 "小区经度",
                    LAT                 "小区纬度",
                    DIRECTION           "小区方位角",
                    HEIGHT              "天线挂高",
                    M_DOWNTILT          "机械下倾",
                    E_DOWNTILT          "电子下倾",
                    STATION_TYPE        "基站类型",
                    ISDIGITALINDOOR     "是否电子室分",
                    DOWN_FREQ           "下行频点",
                    VENDER              "厂家",
                    OWN_SCHOOLYARD      "是否校园站",
                    TOWERADDRESS_CODE   "铁塔站址编号",
                    PROPERTY            "产权归属",
                    SCENE               "场景",
                    IS_SCENESITE        "是否场景站",
                    MARKETING_NETWORK   "销售网络",
                    TERMINALAMOUNT_5G   "终端数量5G",
                    SECTOR_INCOMING     "扇区收入",
                    IS_BUSY             "是否超忙",
                    CONSTRUCTION        "承建方",
                    IS_SAME_ADDRESS     "是否共站址",
                    SAME_ADDRESS_SITES "对应共站址基站ID",
                    IS_SAME_ADDRESS_CT "共站址运营商"
      from STATS_4G_ERROR_SITEINFO
     where sdate = (select max(sdate) from STATS_4G_ERROR_SITEINFO)
       and is_alive = 1
```

```
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-18 5g小区错误或字段不全
    fileName = "gcToyxgx_18_5g_cell_errors_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select distinct ERROR_CLASS "异常类型",
                    CITY_NAME "城市名称",
                    DISTRICT_NAME "行政区名称",
                    CITY_CODE "城市编码",
                    DISTRICT_CODE "行政区编码",
                    NETWORK_NAME "网络类型",
                    PHYSTATION_ADDRESS "物理站名称",
                    STATION_NAME "基站名称",
                    to_char(GNBID) "基站GNBID",
                    CELL_NAME "小区名称",
                    CELL_ID "小区id",
                    LON "小区经度",
                    LAT "小区纬度",
                    DIRECTION "方位角",
                    HEIGHT "天线挂高",
                    M_DOWNTILT "机械下倾",
                    E_DOWNTILT "电子下倾",
                    STATION_TYPE "基站类型",
                    ISDIGITALINDOOR "是否数字化室分",
                    DOWN_FREQ "下行频点",
                    VENDER "厂家",
                    OWN_SCHOOLYARD "是否校园站",
                    TOWERADDRESS_CODE "铁塔地址编码",
                    PROPERTY "产权归属",
                    SCENE "场景",
                    IS_SCENESITE "是否场景站",
                    MARKETING_NETWORK "销售网格",
                    TERMINALAMOUNT_5G "终端数量5G",
                    SECTOR_INCOMING "扇区收入",
                    IS_BUSY "是否超忙",
                    CONSTRUCTION "承建方",
                    IS_SAME_ADDRESS "是否共站址",
                    SAME_ADDRESS_SITES "对应共站址基站ID",
                    IS_SAME_ADDRESS_CT "共站址运营商"
      from STATS_5G_ERROR_SITEINFO
     where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
       and is_alive = 1

    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
```

```
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-19 4g经纬度偏移>3000小区数
    fileName = "gcToyxgx_19_4g_site_deviate_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
    select CITY_NAME          "城市名称",
           DISTRICT_NAME      "行政区名称",
           CITY_CODE          "城市编码",
           DISTRICT_CODE      "行政区编码",
           PHYSTATION_ADDRESS "物理站址",
           BBU_NAME           "BBU名称",
           ENBID              "基站enbid",
           STATION_TYPE       "基站类型",
           VENDER             "设备厂家",
           SCENE              "覆盖厂家",
           DEVIATE_DISTANCE   "基站偏离距离",
           to_char(C1_ECI)    "小区1_eci",
           C1_CELL_NAME       "小区1_小区名称",
           C1_DISTANCE        "小区1_偏离距离",
           to_char(C2_ECI)    "小区2_eci",
           C2_CELL_NAME       "小区2_小区名称",
           C2_DISTANCE        "小区2_偏离距离",
           to_char(C3_ECI)    "小区3_eci",
           C3_CELL_NAME       "小区3_小区名称",
           C3_DISTANCE        "小区3_偏离距离",
           to_char(C4_ECI)    "小区4_eci",
           C4_CELL_NAME       "小区4_小区名称",
           C4_DISTANCE        "小区4_偏离距离",
           to_char(C5_ECI)    "小区5_eci",
           C5_CELL_NAME       "小区5_小区名称",
           C5_DISTANCE        "小区5_偏离距离",
           to_char(C6_ECI)    "小区6_eci",
           C6_CELL_NAME       "小区6_小区名称",
           C6_DISTANCE        "小区6_偏离距离",
           to_char(C7_ECI)    "小区7_eci",
           C7_CELL_NAME       "小区7_小区名称",
           C7_DISTANCE        "小区7_偏离距离",
           to_char(C8_ECI)    "小区8_eci",
           C8_CELL_NAME       "小区8_小区名称",
           C8_DISTANCE        "小区8_偏离距离",
           to_char(C9_ECI)    "小区9_eci",
           C9_CELL_NAME       "小区9_小区名称",
           C9_DISTANCE        "小区9_偏离距离"
      from stats_4G_DEVIATE_SUMMARY_DB
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-20 4g全量工参数据
```

```
    fileName = "gcToyxgx_20_4g_seq_siteinfo_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
     SELECT SDATE "数据日期",
            CITY_CODE "城市编码",
            CITY_NAME "城市名称",
            DISTRICT_CODE "行政区编码",
            DISTRICT_NAME "行政区名称",
            NETWORK_NAME "网络类型",
            PHYSTATION_ADDRESS "物理站名称",
            BBU_NAME "BBU名称",
            to_char(ENBID) "基站ENBID",
            CELL_NAME "小区名称",
            CELL_ID "小区id",
            to_char('46001' || enbid * 256 + cell_id) "小区全球唯一标识",
            LON "小区经度",
            LAT "小区纬度",
            DIRECTION "小区方位角",
            HEIGHT "天线挂高",
            M_DOWNTILT "机械下倾",
            E_DOWNTILT "电子下倾",
            STATION_TYPE "基站类型",
            ISDIGITALINDOOR "是否电子室分",
            DOWN_FREQ "下行频点",
            VENDER "厂家",
            OWN_SCHOOLYARD "是否校园站",
            TOWERADDRESS_CODE "铁塔站址编号",
            PROPERTY "产权归属",
            SCENE "场景",
            IS_SCENESITE "是否场景站",
            MARKETING_NETWORK "销售网络",
            TERMINALAMOUNT_5G "终端数量5G",
            SECTOR_INCOMING "扇区收入",
            IS_BUSY "是否超忙",
            IS_ALIVE "是否有业务",
            IS_ALIVE_UPDATE_TIME "业务更新时间",
            CONSTRUCTION "承建方",
            IS_SAME_ADDRESS "是否共站址",
            SAME_ADDRESS_SITES "对应共站址基站ID",
            IS_SAME_ADDRESS_CT "共站址运营商"
       FROM seq_4g_siteinfo
      where sdate = (select max(sdate) from seq_4g_siteinfo)
        and is_alive = 1
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败", err)
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // 导出文件-21 5g全量工参数据
    fileName = "gcToyxgx_21_5g_seq_siteinfo_" + dateDir + ".csv"
    filepath = fileDir + "/" + fileName
    exportSql = `
     SELECT SDATE "数据日期",
```

```
            CITY_CODE "城市编码",
            CITY_NAME "城市名称",
            DISTRICT_CODE "行政区编码",
            DISTRICT_NAME "行政区名称",
            NETWORK_NAME "网络类型",
            PHYSTATION_ADDRESS "物理站名称",
            STATION_NAME "基站名称",
            to_char(GNBID) "基站GNBID",
            CELL_NAME "小区名称",
            CELL_ID "小区id",
            to_char('46001' || gnbid * 4096 + cell_id) "小区全球唯一识别码",
            LON "小区经度",
            LAT "小区纬度",
            DIRECTION "方位角",
            HEIGHT "天线挂高",
            M_DOWNTILT "机械下倾",
            E_DOWNTILT "电子下倾",
            STATION_TYPE "基站类型",
            ISDIGITALINDOOR "是否数字化室分",
            DOWN_FREQ "下行频点",
            VENDER "厂家",
            OWN_SCHOOLYARD "是否校园站",
            TOWERADDRESS_CODE "铁塔地址编码",
            PROPERTY "产权归属",
            SCENE "场景",
            IS_SCENESITE "是否场景站",
            MARKETING_NETWORK "销售网格",
            TERMINALAMOUNT_5G "终端数量5G",
            SECTOR_INCOMING "扇区收入",
            IS_BUSY "是否超忙",
            IS_ALIVE "是否有业务",
            IS_ALIVE_UPDATE_TIME "业务更新时间",
            CONSTRUCTION "承建方",
            IS_SAME_ADDRESS "是否共站址",
            SAME_ADDRESS_SITES "对应共站址基站ID",
            IS_SAME_ADDRESS_CT "共站址运营商"
       FROM seq_5g_siteinfo
      where sdate = (select max(sdate) from seq_5g_siteinfo)
        and is_alive = 1
    `
    err = sqlQuery2Csv(db, exportSql, filepath)
    if err != nil {
        logger.Fatal("从数据库导出数据到csv失败", err)
    } else {
        if isExist(filepath) {
            waitFiles = append(waitFiles, filepath)
        }
    }

    // logFileName := "C:/Users/ZhangLei/Desktop/Tableau Desktop Pro v2019.4.1 CRACK.rar"

    ftpUploadFile("192.168.0.41:21", "do", "RichrOOt", waitFiles, "/data/bigscreen", dateDir)
    // for _, value := range waitFiles {

    //  ftpUploadFile("127.0.0.1:21", "user", "user", filepath, "./", dateDir)
```

```
    // }

}

// ftpserver：服务器地址+端口；ftpuser：ftp账号；pw：ftp密码；localFile：上传的文件；remotePath：远程地址；saveFileName：上传后的远程文件名字；dayDir：远程地址要创建的目录
func ftpUploadFile(ftpserver string, ftpuser string, pw string, localFiles []string, remotePath string, dayDir string) {
    conn, err := ftp.Connect(ftpserver)
    if err != nil {
        logger.Fatal("Ftp连接失败:" + err.Error())
    } else {
        logger.Printf("Ftp连接成功")
    }

    err = conn.Login(ftpuser, pw)
    if err != nil {
        logger.Fatal("Ftp登录失败:" + err.Error())
    } else {
        logger.Printf("Ftp登录成功")
    }

    dir, err := conn.CurrentDir()
    if err != nil {
        logger.Fatal("打印当前工作目录失败:" + err.Error())
    } else {
        logger.Printf("当前工作目录:" + dir)
    }
    //切换到指定远端目录
    err = conn.ChangeDir(remotePath)
    if err != nil {
        logger.Printf("切换目录错误:" + err.Error())
    } else {
        logger.Printf("切换目录成功:" + remotePath)
    }

    entries, err := conn.List(remotePath)
    if err != nil {
        logger.Printf(err.Error())
    }
    folderIsExists := false
    for _, entry := range entries {
        if entry.Type == ftp.EntryTypeFolder {
            if entry.Name == dayDir {
                folderIsExists = true //说明目录存在
                break
            }
        }
    }
    if !folderIsExists {
        err = conn.MakeDir(dayDir)
        if err != nil {
            logger.Printf("创建目录:" + dayDir + ";错误:" + err.Error())
        } else {
            logger.Printf("创建目录:" + dayDir + " 成功")
        }
    }
```

```
1251
1252        err = conn.ChangeDir(dayDir)
1253        if err != nil {
1254            logger.Printf("切换目录错误:" + err.Error())
1255        } else {
1256            logger.Printf("切换目录成功")
1257        }
1258        dir, err = conn.CurrentDir()
1259        if err != nil {
1260            logger.Printf("获取当前目录错误:" + err.Error())
1261        } else {
1262            logger.Printf("打印当前目录:" + dayDir)
1263        }
1264
1265        for _, value := range localFiles {
1266            file, err := os.Open(value)
1267            saveFileName := path.Base(value)
1268
1269            if err != nil {
1270                logger.Printf("打开上传文件错误:" + err.Error())
1271            } else {
1272                logger.Printf("文件读取完成,准备上传...")
1273            }
1274            defer file.Close()
1275            err = conn.Stor(saveFileName, file)
1276            if err != nil {
1277                logger.Printf("文件上传错误:" + err.Error())
1278            } else {
1279                logger.Printf("文件上传成功:" + remotePath + dayDir + "/" +
saveFileName)
1280            }
1281        }
1282
1283        conn.Logout()
1284        conn.Quit()
1285
1286    }
```

6.4 有业务无工参报表源码

```
1  # 该工作其实是工参治理形成闭环的一项工作。
2  # 目前非自动环境：每周按需求输出有业务无工参详表给刘子龙，用于下发地市填报数据更新，并将更新完的工参数据导入我们的分析系统结合实时业务量再次分析检查，从而实现工参治理的闭环。自动化环境初步规划有亚信工单流程触发。
```

6.4.1 报表脚本

```
1  # 目前通报的问题有6类。oracle_2_excel.exe工具可实现自动导出下面6类数据到Excel表。
2
3  # 1）4G有业务无工参小区详情
4  SELECT * FROM STATS_HAS_FLOW_NO_4G_GC
5
6
7  # 2）4G有业务无工参小区详情
8  SELECT * FROM STATS_HAS_FLOW_NO_5G_GC
9
```

```

# 3）工参问题+基站经纬度偏离
--create table  STATS_4G_ERR_FLAG AS
with a_2d as
 (select distinct ERROR_CLASS, a.city_name, a.ENBID, a.cell_id
    from STATS_4G_ERROR_SITEINFO a
   where a.is_alive = 1),
b_err as
 (select distinct '基站经纬度偏离大于3000米' ERROR_CLASS,
                  a.city_name,
                  a.ENBID,
                  a.cell_id
    from seq_4g_siteinfo a, STATS_4G_DEVIATE_SUMMARY_DB b
   where a.sdate = (select max(sdate) from seq_4g_siteinfo)
     and a.is_alive = 1
     and a.city_name = b.city_name
     and a.enbid = b.enbid)
select city_name,
       ENBID,
       cell_id,
       listagg(ERROR_CLASS, ',') within group(order by ENBID) as ERROR_CLASS
  from (select * from a_2d union all select * from b_err)
 group by city_name, ENBID, cell_id;


# 4）5g工参不全不准问题
SELECT distinct city_name, gnbid, CELL_ID, ERROR_CLASS
  FROM STATS_5G_ERROR_SITEINFO
 where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
   and is_alive = 1;


# 5）4g2天无业务
select distinct a.city_name,
                a.ENBID,
                a.cell_id,
                '4G小区2天无业务' ERROR_CLASS
  from seq_4g_siteinfo a, STATS_4g_CELL_FLOW_7d_DAY b
 where a.sdate = (select max(sdate) from seq_4g_siteinfo)
   and a.is_alive = 1
   and b.sdate = (select max(sdate) from STATS_4g_CELL_FLOW_7d_DAY)
   and a.enbid = b.enbid
   and a.cell_id = b.cell_id;


# 6）5g2天无业务
select distinct a.city_name,
                a.gnbid,
                a.cell_id,
                '5G小区2天无业务' ERROR_CLASS
  from seq_5g_siteinfo a, STATS_5g_CELL_FLOW_7d_DAY b
 where a.sdate = (select max(sdate) from seq_5g_siteinfo)
   and a.is_alive = 1
   and b.sdate = (select max(sdate) from STATS_5g_CELL_FLOW_7d_DAY)
   and a.gnbid = b.gnbid
   and a.cell_id = b.cell_id;
```

6.4.2 工具源码

```
//oracle_2_excel.exe工具的源码：
package main

import (
    "database/sql"
    "fmt"
    "log"
    "os"
    "strconv"
    "strings"
    "time"

    _ "github.com/mattn/go-oci8"
    "github.com/xuri/excelize"
)

var logger *log.Logger
var file *os.File
var err error

func init() {
    file, err = os.OpenFile("logger.log",
os.O_APPEND|os.O_CREATE|os.O_WRONLY, 666)
    if err != nil {
        log.Fatal(err)
    }

    logger = log.New(file, "", log.LstdFlags)
    logger.SetPrefix("Test- ") // 设置日志前缀
    logger.SetFlags(log.Ldate | log.Ltime | log.Lshortfile)
    /*
        const (
            // 字位共同控制输出日志信息的细节。不能控制输出的顺序和格式。
            // 在所有项目后会有一个冒号：2009/01/23 01:23:23.123123
/a/b/c/d.go:23: message
            Ldate         = 1 << iota     // 日期：2009/01/23
            Ltime                         // 时间：01:23:23
            Lmicroseconds                 // 微秒分辨率：01:23:23.123123（用于
增强Ltime位）
            Llongfile                     // 文件全路径名+行号：
/a/b/c/d.go:23
            Lshortfile                    // 文件无路径名+行号：d.go:23（会覆盖
掉Llongfile）
            LstdFlags     = Ldate | Ltime // 标准logger的初始值
        )
    */
}

func sqlQuery2Csv(db *sql.DB, sqlStmt string) (map[int][][]byte, []string,
error) {
    rows, err := db.Query(sqlStmt)
    if err != nil {
        logger.Fatal("sqlQuery执行失败,err: %s,err sql:%s", err, sqlStmt)
    }
    defer rows.Close()
```

```
    //返回所有列
    cols, _ := rows.Columns()
    //这里表示一行所有列的值，用[]byte表示
    result := make(map[int][][]byte, 10)

    //这里表示一行填充数据
    scans := make([]interface{}, len(cols))
    //这里scans引用vals，把数据填充到[]byte里

    // 统计行数
    i := 0
    //result := make(map[int]map[string]string)

    for rows.Next() {

        //填充数据
        vals := make([][]byte, len(cols))
        for k, _ := range vals {
            scans[k] = &vals[k]
        }
        rows.Scan(scans...)
        //每行数据
        //把vals中的数据复制到row中
        // for _, v := range vals {
        //  // key := cols[k]
        //  //这里把[]byte数据转成string
        //  row = append(row, string(v))
        // }
        result[i] = vals
        i++
    }

    return result, cols, nil
}

func isExist(slice []string, val string) (string, bool) {
    for _, item := range slice {
        if strings.ToUpper(item) == strings.ToUpper(val) {
            return item, true
        }
    }
    return "", false
}

func IsNum(s string) bool {
    _, err := strconv.ParseFloat(s, 64)
    return err == nil
}

func main() {
    // 数据库部分
    dateDir := time.Now().Format("20060102") // dateDir :=
time.Now().Format("200601")
    db, err := sql.Open("oci8", fmt.Sprintf("%s/%s@%s", "c##fast491",
"F@st491*321", "192.168.0.64:1521/fast"))
    if err != nil {
        logger.Fatal("oracle登录失败:" + err.Error())
    }
```

```
    defer db.Close()
    logger.Printf("数据库已登录")

    err = db.Ping()
    if err != nil {
        logger.Fatal("oracle链接不可达:" + err.Error())
    }

    // 创建Excel工作簿
    file := excelize.NewFile()
    styleID, err := file.NewStyle(`{"font":{"color":"#777777"}}`)
    if err != nil {
        logger.Println(err)
    }

    // 编号001
    sheets := file.GetSheetList()

    sname, existFlag := isExist(sheets, "sheet1")

    if existFlag {
        file.SetSheetName(sname, "4G工参不准不准+经纬度偏离")
    } else {
        file.NewSheet("4G工参不准不准+经纬度偏离")
    }

    streamWriter, err := file.NewStreamWriter("4G工参不准不准+经纬度偏离")
    if err != nil {
        logger.Println(err)
    }
    if err := streamWriter.SetRow("A1", []interface{}{
        excelize.Cell{StyleID: styleID, Value: "Data"}}); err != nil {
        logger.Println(err)
    }

    exportSql := `
    with a_2d as
     (select distinct ERROR_CLASS, a.city_name, a.ENBID, a.cell_id
        from STATS_4G_ERROR_SITEINFO a
       where a.is_alive = 1),
    b_err as
     (select distinct '基站经纬度偏离大于3000米' ERROR_CLASS,
                      a.city_name,
                      a.ENBID,
                      a.cell_id
        from seq_4g_siteinfo a, STATS_4G_DEVIATE_SUMMARY_DB b
       where a.sdate = (select max(sdate) from seq_4g_siteinfo)
         and a.is_alive = 1
         and a.city_name = b.city_name
         and a.enbid = b.enbid)
    select city_name,
           ENBID,
           cell_id,
           listagg(ERROR_CLASS, ',') within group(order by ENBID) as
ERROR_CLASS
      from (select * from a_2d union all select * from b_err)
     group by city_name, ENBID, cell_id
    `
```

```
    // sql末尾不能有分号

    data, coln, err := sqlQuery2Csv(db, exportSql)
    //由于streamWriter.SetRow参数需要是[]interface{}格式，[]string不能直接转换，因此这里需要重新定义一个新的[]interface{}类型做下转换。
    colname := make([]interface{}, len(coln))
    colLen := len(coln)
    for i := 0; i < colLen; i++ {
        colname[i] = coln[i]
    }

    // logger.Println(data)
    cell, _ := excelize.CoordinatesToCellName(1, 1)
    if err := streamWriter.SetRow(cell, colname); err != nil {
        logger.Println(err)
    }

    rowLen := len(data)
    for row := 0; row < rowLen; row++ {
        v := data[row]
        //rowLen索引从0开始，第一行是表头，因此只能从第二行写。
        cell, _ := excelize.CoordinatesToCellName(1, row+2)
        strV := make([]interface{}, len(v))
        for k1, v1 := range v {
            cellV := string(v1)
            i1, err := strconv.Atoi(cellV)
            if err != nil {
                strV[k1] = cellV
            } else {
                strV[k1] = i1
            }
        }

        // logger.Println(strV)

        if err := streamWriter.SetRow(cell, strV); err != nil {
            logger.Println(err)
        }
    }

    if err := streamWriter.Flush(); err != nil {
        logger.Println(err)
    }
    logger.Println("4G工参不准不准+经纬度偏离  写入成功")

    // 编号002
    sheets = file.GetSheetList()

    sname, existFlag = isExist(sheets, strings.ToUpper("sheet2"))

    if existFlag {
        file.SetSheetName(sname, "5G工参不全不准")
    } else {
        file.NewSheet("5G工参不全不准")
    }

    streamWriter, err = file.NewStreamWriter("5G工参不全不准")
    if err != nil {
```

```
        logger.Println(err)
    }
    // if err := streamWriter.SetRow("A1", []interface{}{
    //  excelize.Cell{StyleID: styleID, Value: "Data"}}); err != nil {
    //  fmt.Println(err)
    // }

    exportSql = `
    SELECT distinct city_name, to_char(gnbid) gnbid, CELL_ID, ERROR_CLASS
      FROM STATS_5G_ERROR_SITEINFO
     where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
       and is_alive = 1
    `
    // sql末尾不能有分号

    data, coln, err = sqlQuery2Csv(db, exportSql)
    //由于streamWriter.SetRow参数需要是[]interface{}格式，[]string不能直接转换，因此这里需要重新定义一个新的[]interface{}类型做下转换。
    colname = make([]interface{}, len(coln))
    colLen = len(coln)
    for i := 0; i < colLen; i++ {
        colname[i] = coln[i]
    }

    // logger.Println(data)
    cell, _ = excelize.CoordinatesToCellName(1, 1)
    if err := streamWriter.SetRow(cell, colname); err != nil {
        logger.Println(err)
    }

    rowLen = len(data)
    for row := 0; row < rowLen; row++ {
        v := data[row]
        //rowLen索引从0开始，第一行是表头，因此只能从第二行写。
        cell, _ := excelize.CoordinatesToCellName(1, row+2)
        strV := make([]interface{}, len(v))
        for k1, v1 := range v {
            cellV := string(v1)
            i1, err := strconv.Atoi(cellV)
            if err != nil {
                strV[k1] = cellV
            } else {
                strV[k1] = i1
            }
        }

        // logger.Println(strV)

        if err := streamWriter.SetRow(cell, strV); err != nil {
            logger.Println(err)
        }
    }
    // 执行flush将数据写入sheet
    if err := streamWriter.Flush(); err != nil {
        logger.Println(err)
    }

    logger.Println("5G工参不全不准  写入成功")
```

```
277
278     // 编号003
279     sheets = file.GetSheetList()
280
281     sname, existFlag = isExist(sheets, strings.ToUpper("sheet3"))
282
283     if existFlag {
284         file.SetSheetName(sname, "4G小区2天无业务")
285     } else {
286         file.NewSheet("4G小区2天无业务")
287     }
288
289     streamWriter, err = file.NewStreamWriter("4G小区2天无业务")
290     if err != nil {
291         logger.Println(err)
292     }
293     if err := streamWriter.SetRow("A1", []interface{}{
294         excelize.Cell{StyleID: styleID, Value: "Data"}}); err != nil {
295         logger.Println(err)
296     }
297
298     exportSql = `
299     select distinct a.city_name,
300                     a.ENBID,
301                     a.cell_id,
302                     '4G小区2天无业务' ERROR_CLASS
303       from seq_4g_siteinfo a, STATS_4g_CELL_FLOW_7d_DAY b
304      where a.sdate = (select max(sdate) from seq_4g_siteinfo)
305        and a.is_alive = 1
306        and b.sdate = (select max(sdate) from STATS_4g_CELL_FLOW_7d_DAY)
307        and a.enbid = b.enbid
308        and a.cell_id = b.cell_id
309     `
310     // sql末尾不能有分号
311
312     data, coln, err = sqlQuery2Csv(db, exportSql)
313     //由于streamWriter.SetRow参数需要是[]interface{}格式，[]string不能直接转换，
    因此这里需要重新定义一个新的[]interface{}类型做下转换。
314     colname = make([]interface{}, len(coln))
315     colLen = len(coln)
316     for i := 0; i < colLen; i++ {
317         colname[i] = coln[i]
318     }
319
320     // logger.Println(data)
321     cell, _ = excelize.CoordinatesToCellName(1, 1)
322     if err := streamWriter.SetRow(cell, colname); err != nil {
323         logger.Println(err)
324     }
325
326     rowLen = len(data)
327     for row := 0; row < rowLen; row++ {
328         v := data[row]
329         //rowLen索引从0开始，第一行是表头，因此只能从第二行写。
330         cell, _ := excelize.CoordinatesToCellName(1, row+2)
331         strV := make([]interface{}, len(v))
332         for k1, v1 := range v {
333             cellV := string(v1)
```

```
            i1, err := strconv.Atoi(cellV)
            if err != nil {
                strV[k1] = cellV
            } else {
                strV[k1] = i1
            }
        }

        // logger.Println(strV)

        if err := streamWriter.SetRow(cell, strV); err != nil {
            logger.Println(err)
        }
    }
    // 执行flush将数据写入sheet
    if err := streamWriter.Flush(); err != nil {
        logger.Println(err)
    }

    logger.Println("4G小区2天无业务 写入成功")

    // 编号004
    sheets = file.GetSheetList()

    sname, existFlag = isExist(sheets, strings.ToUpper("sheet4"))

    if existFlag {
        file.SetSheetName(sname, "5G小区2天无业务")
    } else {
        file.NewSheet("5G小区2天无业务")
    }

    streamWriter, err = file.NewStreamWriter("5G小区2天无业务")
    if err != nil {
        logger.Println(err)
    }
    if err := streamWriter.SetRow("A1", []interface{}{
        excelize.Cell{StyleID: styleID, Value: "Data"}}); err != nil {
        logger.Println(err)
    }

    exportSql = `
    select distinct a.city_name,
                    to_char(a.gnbid) gnbid,
                    a.cell_id,
                    '5G小区2天无业务' ERROR_CLASS
      from seq_5g_siteinfo a, STATS_5g_CELL_FLOW_7d_DAY b
     where a.sdate = (select max(sdate) from seq_5g_siteinfo)
       and a.is_alive = 1
       and b.sdate = (select max(sdate) from STATS_5g_CELL_FLOW_7d_DAY)
       and a.gnbid = b.gnbid
       and a.cell_id = b.cell_id
    `
    // sql末尾不能有分号

    data, coln, err = sqlQuery2Csv(db, exportSql)
    //由于streamWriter.SetRow参数需要是[]interface{}格式，[]string不能直接转换，因此这里需要重新定义一个新的[]interface{}类型做下转换。
```

```
    colname = make([]interface{}, len(coln))
    colLen = len(coln)
    for i := 0; i < colLen; i++ {
        colname[i] = coln[i]
    }

    // logger.Println(data)
    cell, _ = excelize.CoordinatesToCellName(1, 1)
    if err := streamWriter.SetRow(cell, colname); err != nil {
        logger.Println(err)
    }

    rowLen = len(data)
    for row := 0; row < rowLen; row++ {
        v := data[row]
        //rowLen索引从0开始，第一行是表头，因此只能从第二行写。
        cell, _ := excelize.CoordinatesToCellName(1, row+2)
        strV := make([]interface{}, len(v))
        for k1, v1 := range v {
            cellV := string(v1)
            i1, err := strconv.Atoi(cellV)
            if err != nil {
                strV[k1] = cellV
            } else {
                strV[k1] = i1
            }
        }

        // logger.Println(strV)

        if err := streamWriter.SetRow(cell, strV); err != nil {
            logger.Println(err)
        }
    }
    // 执行flush将数据写入sheet
    if err := streamWriter.Flush(); err != nil {
        logger.Println(err)
    }

    logger.Println("5G小区2天无业务 写入成功")

    // 编号005
    sheets = file.GetSheetList()

    sname, existFlag = isExist(sheets, strings.ToUpper("sheet5"))

    if existFlag {
        file.SetSheetName(sname, "4G有业务无工参")
    } else {
        file.NewSheet("4G有业务无工参")
    }

    streamWriter, err = file.NewStreamWriter("4G有业务无工参")
    if err != nil {
        logger.Println(err)
    }
    if err := streamWriter.SetRow("A1", []interface{}{
        excelize.Cell{StyleID: styleID, Value: "Data"}}); err != nil {
```

```
        logger.Println(err)
    }

    exportSql = `SELECT * FROM STATS_HAS_FLOW_NO_4G_GC
    `
    // sql末尾不能有分号

    data, coln, err = sqlQuery2Csv(db, exportSql)
    //由于streamWriter.SetRow参数需要是[]interface{}格式，[]string不能直接转换，因此这里需要重新定义一个新的[]interface{}类型做下转换。
    colname = make([]interface{}, len(coln))
    colLen = len(coln)
    for i := 0; i < colLen; i++ {
        colname[i] = coln[i]
    }

    // logger.Println(data)
    cell, _ = excelize.CoordinatesToCellName(1, 1)
    if err := streamWriter.SetRow(cell, colname); err != nil {
        logger.Println(err)
    }

    rowLen = len(data)
    for row := 0; row < rowLen; row++ {
        v := data[row]
        //rowLen索引从0开始，第一行是表头，因此只能从第二行写。
        cell, _ := excelize.CoordinatesToCellName(1, row+2)
        strV := make([]interface{}, len(v))
        for k1, v1 := range v {
            cellV := string(v1)
            i1, err := strconv.Atoi(cellV)
            if err != nil {
                strV[k1] = cellV
            } else {
                strV[k1] = i1
            }
        }

        // logger.Println(strV)

        if err := streamWriter.SetRow(cell, strV); err != nil {
            logger.Println(err)
        }
    }
    // 执行flush将数据写入sheet
    if err := streamWriter.Flush(); err != nil {
        logger.Println(err)
    }

    logger.Println("4G有业务无工参 写入成功")

    // 编号005
    sheets = file.GetSheetList()

    sname, existFlag = isExist(sheets, strings.ToUpper("sheet6"))

    if existFlag {
        file.SetSheetName(sname, "5G有业务无工参")
```

```
    } else {
        file.NewSheet("5G有业务无工参")
    }

    streamWriter, err = file.NewStreamWriter("5G有业务无工参")
    if err != nil {
        logger.Println(err)
    }
    if err := streamWriter.SetRow("A1", []interface{}{
        excelize.Cell{StyleID: styleID, Value: "Data"}}); err != nil {
        logger.Println(err)
    }

    exportSql = `SELECT * FROM STATS_HAS_FLOW_NO_5G_GC
    `
    // sql末尾不能有分号

    data, coln, err = sqlQuery2Csv(db, exportSql)
    //由于streamWriter.SetRow参数需要是[]interface{}格式，[]string不能直接转换，因此这里需要重新定义一个新的[]interface{}类型做下转换。
    colname = make([]interface{}, len(coln))
    colLen = len(coln)
    for i := 0; i < colLen; i++ {
        colname[i] = coln[i]
    }

    // logger.Println(data)
    cell, _ = excelize.CoordinatesToCellName(1, 1)
    if err := streamWriter.SetRow(cell, colname); err != nil {
        logger.Println(err)
    }

    rowLen = len(data)
    for row := 0; row < rowLen; row++ {
        v := data[row]
        //rowLen索引从0开始，第一行是表头，因此只能从第二行写。
        cell, _ := excelize.CoordinatesToCellName(1, row+2)
        strV := make([]interface{}, len(v))
        for k1, v1 := range v {
            cellV := string(v1)
            i1, err := strconv.Atoi(cellV)
            if err != nil {
                strV[k1] = cellV
            } else {
                strV[k1] = i1
            }
        }

        // logger.Println(strV)

        if err := streamWriter.SetRow(cell, strV); err != nil {
            logger.Println(err)
        }
    }
    // 执行flush将数据写入sheet
    if err := streamWriter.Flush(); err != nil {
        logger.Println(err)
    }
```

```
563
564     logger.Println("5G有业务无工参 写入成功")
565
566     if err := file.SaveAs("45G工参质量检查_" + dateDir + ".xlsx"); err != nil
    {
567         logger.Println(err)
568     }
569
570     logger.Println("45G工参质量检查_" + dateDir + ".xlsx 保存成功~")
571
572 }
```

7、业务包全部代码

目前工参治理大屏后端功能模块部署在c##fast491@192.168.0.64:1521/fast中，包体名为pkg_basedata_governance_v4，下面为截至20210817日的最新源码信息：

```
1  # pkg_basedata_governance_v4 完整代码
2  CREATE OR REPLACE PACKAGE pkg_basedata_governance_v4 IS
3      --20210512 张磊 新增7天低业务量需求，基站7GB，扇区2GB
4  /*    procedure sdr_flow_4g_cell_hour(i_sdate varchar2);
5      procedure sdr_flow_4g_cell_day(i_sdate varchar2);
6      procedure sdr_flow_4g_enb_hour(i_sdate varchar2);
7      procedure sdr_flow_4g_enb_day(i_sdate varchar2);
8      procedure sdr_flow_5g_enb_hour(i_sdate varchar2);
9      procedure sdr_flow_5g_enb_day(i_sdate varchar2);*/
10     procedure ini_4g_table;
11     procedure ini_5g_table;
12     procedure cal_4g_data;
13     procedure cal_5g_data;
14     procedure STATS_4G_SDRFLOW(i_sdate varchar2 DEFAULT NULL);
15     procedure STATS_5G_OMCFLOW(i_sdate varchar2 DEFAULT NULL);
16     procedure sdr_flow_5g_cell_day(i_sdate varchar2);
17     procedure sdr_flow_5g_enb_day(i_sdate varchar2);
18     -- 一月有业务基站小区统计
19     procedure STATS_4g_SITE_FLOW_2M_DAY(i_sdate varchar2,idays number);
20     procedure STATS_4g_CELL_FLOW_2M_DAY(i_sdate varchar2,idays number);
21     procedure STATS_5g_SITE_FLOW_2M_DAY(i_sdate varchar2,idays number);
22     procedure STATS_5g_CELL_FLOW_2M_DAY(i_sdate varchar2,idays number);
23     procedure STATS_5g_PHYSITE_FLOW_2M_DAY(i_sdate varchar2,idays number);
24     -- 7天无业务基站小区统计
25     procedure STATS_4g_SITE_FLOW_7d_DAY(i_sdate varchar2,idays number);
26     procedure STATS_4g_CELL_FLOW_7d_DAY(i_sdate varchar2,idays number);
27     procedure STATS_5g_SITE_FLOW_7d_DAY(i_sdate varchar2,idays number);
28     procedure STATS_5g_phySITE_FLOW_7d_DAY(i_sdate varchar2,idays number);
29     procedure STATS_5g_CELL_FLOW_7d_DAY(i_sdate varchar2,idays number);
30     -- 7天低业务基站小区统计
31     procedure STATS_4g_SITE_LOWFLOW_7d_DAY(i_sdate varchar2,idays number);
32     procedure STATS_4g_CELL_LOWFLOW_7d_DAY(i_sdate varchar2,idays number);
33     procedure STATS_5g_SITE_LOWFLOW_7d_DAY(i_sdate varchar2,idays number);
34     procedure STATS_5g_PHYSITE_LOFLOW_7d_DAY(i_sdate varchar2,idays number);
35     procedure STATS_5g_CELL_LOWFLOW_7d_DAY(i_sdate varchar2,idays number);
36     -- 2天无业务基站小区统计
37     procedure STATS_4g_SITE_FLOW_48h_hour(i_sdate varchar2,idays number);
38     procedure STATS_4g_CELL_FLOW_48h_hour(i_sdate varchar2,idays number);
39     procedure STATS_5g_SITE_FLOW_48h_hour(i_sdate varchar2,idays number);
40     procedure STATS_5g_phySITE_FLOW_48h_hour(i_sdate varchar2,idays number);
```

```
    procedure STATS_5g_CELL_FLOW_48h_hour(i_sdate varchar2,idays number);
    -- 工参不准或错误
    procedure STATS_4G_ERROR_SITEINFO(i_sdate varchar2 default null);
    procedure STATS_5G_ERROR_SITEINFO(i_sdate varchar2 default null);
    -- 校园45G站点计算
    procedure STATS_4G_COLLEGES_SITEINFO(i_sdate varchar2,outdoor_range number,indoor_range number);
    procedure STATS_5G_COLLEGES_SITEINFO(i_sdate varchar2,outdoor_range number,indoor_range number);
    -- 4G基站偏离度计算
    procedure STATS_4G_DEVIATE_SUMMARY_DB(i_sdate varchar2 default null);
    procedure STATS_4G_DEVIATE_SUMMARY_QL(i_sdate varchar2 default null);
    procedure STATS_HAS_FLOW_NO_4G_GC(i_sdate varchar2 default null);
    procedure STATS_HAS_FLOW_NO_5G_GC(i_sdate varchar2 default null);
end;

```

```
CREATE OR REPLACE PACKAGE BODY pkg_basedata_governance_v4 AS

    --5g没有小区级小时粒度数据。

    --1 4g 小区级小时粒度流量
    /*procedure sdr_flow_4g_cell_hour(i_sdate varchar2) is

    begin
      delete from sdr_flow_4g_cell_hour where trunc(sdate) = to_date(i_sdate,'yyyymmdd');
      insert into sdr_flow_4g_cell_hour
      select sdate,
             enbid,
             cell_id,
             net,
             LAYER2ID,
             REGION_NAME,
             LAYER3ID,
             COUNTY_NAME,
             sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
             sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
             sum(THROUGHPUT) THROUGHPUT
        from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
                                       TO_DATE('1970-01-01 08:00:00',
                                               'YYYY-MM-DD HH24:MI:SS'),
                                       'yyyy-mm-dd hh24'),
                               'yyyy-mm-dd hh24')*\ sdate,
                     t.ran_ne_id,
                     gnbid,
                     sitename,
                     cgisai,
                     --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
                     --case when length(cgisai)=12 then substr(cgisai,6,5) else substr(cgisai,6,4) end enbid,
                     --case when length(cgisai)=12 then substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
                     to_number((case
                                  when length(cgisai) = 12 then
```

```
                                  substr(cgisai, 6, 5)
                                 else
                                  substr(cgisai, 6, 4)
                               end),
                               'XXXXXXXX') enbid,
                     to_number((case
                                 when length(cgisai) = 12 then
                                  substr(cgisai, -2)
                                 else
                                  substr(cgisai, -4)
                               end),
                               'XXXXXXXX') cell_id,
                     \*

IF(LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))

IF(LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
                     rat,
                     CASE
                       WHEN rat = 9 THEN
                        '5G'
                       WHEN RAT = 6 THEN
                        '4G'
                     END NET,
                     l4_dw_throughput,
                     l4_ul_throughput,
                     (l4_dw_throughput + l4_ul_throughput) throughput,
                     layer2id,
                     d.region_name,
                     layer3id,
                     h.county_name
                FROM sdr_flow_cell_source_csv t
                LEFT JOIN sdr_flow_region_code d
                  ON t.layer2id = d.region_id
                LEFT JOIN sdr_flow_mapping_ran_ne_id g
                  ON t.ran_ne_id = g.ran_ne_id
                LEFT JOIN SDR_FLOW_county_code h
                  ON t.layer3id = h.county_id
               WHERE T.RAT = 6
                 and cgisai is not null
                 and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
       group by sdate,
                enbid,
                cell_id,
                net,
                LAYER2ID,
                REGION_NAME,
                LAYER3ID,
                COUNTY_NAME;

    end;*/


    --2 4g 小区级天粒度流量
    /*procedure sdr_flow_4g_cell_day(i_sdate varchar2) is

    begin
```

```
      delete from sdr_flow_4g_cell_day where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
      insert into sdr_flow_4g_cell_day
      select trunc(sdate) sdate,
             enbid,
             cell_id,
             net,
             LAYER2ID,
             REGION_NAME,
             LAYER3ID,
             COUNTY_NAME,

             sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
             sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
             sum(THROUGHPUT) THROUGHPUT
        from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
                                       TO_DATE('1970-01-01 08:00:00',
                                               'YYYY-MM-DD HH24:MI:SS'),
                                       'yyyy-mm-dd hh24'),
                               'yyyy-mm-dd hh24')*\ sdate,
                     t.ran_ne_id,
                     gnbid,
                     sitename,
                     cgisai,
                     --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
                     --case when length(cgisai)=12 then substr(cgisai,6,5)
else substr(cgisai,6,4) end enbid,
                     --case when length(cgisai)=12 then
substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
                     to_number((case
                                 when length(cgisai) = 12 then
                                  substr(cgisai, 6, 5)
                                 else
                                  substr(cgisai, 6, 4)
                               end),
                               'XXXXXXXX') enbid,
                     to_number((case
                                 when length(cgisai) = 12 then
                                  substr(cgisai, -2)
                                 else
                                  substr(cgisai, -4)
                               end),
                               'XXXXXXXX') cell_id,

                     \*

IF(LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))

IF(LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
                     rat,
                     CASE
                       WHEN rat = 9 THEN
                        '5G'
                       WHEN RAT = 6 THEN
                        '4G'
                     END NET,
                     l4_dw_throughput,
                     l4_ul_throughput,
```

```
                    (l4_dw_throughput + l4_ul_throughput) throughput,
                    layer2id,
                    d.region_name,
                    layer3id,
                    h.county_name
               FROM sdr_flow_cell_source_csv t
               LEFT JOIN sdr_flow_region_code d
                 ON t.layer2id = d.region_id
               LEFT JOIN sdr_flow_mapping_ran_ne_id g
                 ON t.ran_ne_id = g.ran_ne_id
               LEFT JOIN SDR_FLOW_county_code h
                 ON t.layer3id = h.county_id
              WHERE T.RAT = 6
                and cgisai is not null
                and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
       group by trunc(sdate),
                enbid,
                cell_id,
                net,
                LAYER2ID,
                REGION_NAME,
                LAYER3ID,
                COUNTY_NAME;

    end;*/



    --3 4g 基站级小时粒度流量
    /*procedure sdr_flow_4g_enb_hour(i_sdate varchar2) is

    begin
      delete from sdr_flow_4g_enb_hour where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
      insert into sdr_flow_4g_enb_hour
      select sdate,
             enbid,
             net,
             LAYER2ID,
             REGION_NAME,
             LAYER3ID,
             COUNTY_NAME,

             sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
             sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
             sum(THROUGHPUT) THROUGHPUT
        from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
                                       TO_DATE('1970-01-01 08:00:00',
                                               'YYYY-MM-DD HH24:MI:SS'),
                                       'yyyy-mm-dd hh24'),
                              'yyyy-mm-dd hh24')*\ sdate,
                     t.ran_ne_id,
                     gnbid,
                     sitename,
                     cgisai,
                     --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
                     --case when length(cgisai)=12 then substr(cgisai,6,5)
else substr(cgisai,6,4) end enbid,
```

```
                    --case when length(cgisai)=12 then
substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
                    to_number((case
                                when length(cgisai) = 12 then
                                 substr(cgisai, 6, 5)
                                else
                                 substr(cgisai, 6, 4)
                              end),
                              'XXXXXXXX') enbid,
                    to_number((case
                                when length(cgisai) = 12 then
                                 substr(cgisai, -2)
                                else
                                 substr(cgisai, -4)
                              end),
                              'XXXXXXXX') cell_id,

                    \*

IF(LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))

IF(LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
                    rat,
                    CASE
                      WHEN rat = 9 THEN
                       '5G'
                      WHEN RAT = 6 THEN
                       '4G'
                    END NET,
                    l4_dw_throughput,
                    l4_ul_throughput,
                    (l4_dw_throughput + l4_ul_throughput) throughput,
                    layer2id,
                    d.region_name,
                    layer3id,
                    h.county_name
               FROM sdr_flow_cell_source_csv t
               LEFT JOIN sdr_flow_region_code d
                 ON t.layer2id = d.region_id
               LEFT JOIN sdr_flow_mapping_ran_ne_id g
                 ON t.ran_ne_id = g.ran_ne_id
               LEFT JOIN SDR_FLOW_county_code h
                 ON t.layer3id = h.county_id
              WHERE T.RAT = 6
                and cgisai is not null
                and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
       group by sdate,
                enbid,
                net,
                LAYER2ID,
                REGION_NAME,
                LAYER3ID,
                COUNTY_NAME;

    end;*/
```

```
    --4 4g 基站级天粒度流量
    /*procedure sdr_flow_4g_enb_day(i_sdate varchar2) is

    begin
      delete from sdr_flow_4g_enb_day where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
      insert into sdr_flow_4g_enb_day
      select trunc(sdate) sdate,
             enbid,
             net,
             LAYER2ID,
             REGION_NAME,
             LAYER3ID,
             COUNTY_NAME,

             sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
             sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
             sum(THROUGHPUT) THROUGHPUT
        from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
                                       TO_DATE('1970-01-01 08:00:00',
                                               'YYYY-MM-DD HH24:MI:SS'),
                                       'yyyy-mm-dd hh24'),
                               'yyyy-mm-dd hh24')*\ sdate,
                     t.ran_ne_id,
                     gnbid,
                     sitename,
                     cgisai,
                     --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
                     --case when length(cgisai)=12 then substr(cgisai,6,5)
else substr(cgisai,6,4) end enbid,
                     --case when length(cgisai)=12 then
substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
                     to_number((case
                                 when length(cgisai) = 12 then
                                  substr(cgisai, 6, 5)
                                 else
                                  substr(cgisai, 6, 4)
                               end),
                               'XXXXXXXX') enbid,
                     to_number((case
                                 when length(cgisai) = 12 then
                                  substr(cgisai, -2)
                                 else
                                  substr(cgisai, -4)
                               end),
                               'XXXXXXXX') cell_id,

                     \*
IF(LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))
IF(LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
                     rat,
                     CASE
                       WHEN rat = 9 THEN
                        '5G'
                       WHEN RAT = 6 THEN
                        '4G'
```

```
                    END NET,
                    l4_dw_throughput,
                    l4_ul_throughput,
                    (l4_dw_throughput + l4_ul_throughput) throughput,
                    layer2id,
                    d.region_name,
                    layer3id,
                    h.county_name
               FROM sdr_flow_cell_source_csv t
               LEFT JOIN sdr_flow_region_code d
                 ON t.layer2id = d.region_id
               LEFT JOIN sdr_flow_mapping_ran_ne_id g
                 ON t.ran_ne_id = g.ran_ne_id
               LEFT JOIN SDR_FLOW_county_code h
                 ON t.layer3id = h.county_id
              WHERE T.RAT = 6
                and cgisai is not null
                and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
       group by trunc(sdate),
                enbid,
                net,
                LAYER2ID,
                REGION_NAME,
                LAYER3ID,
                COUNTY_NAME;

    end;*/


    --5 5g 基站级小时粒度流量
    /*procedure sdr_flow_5g_enb_hour(i_sdate varchar2) is

    begin
      delete from sdr_flow_5g_enb_hour where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
      insert into sdr_flow_5g_enb_hour
      select sdate,
             RAN_NE_ID,
             GNBID,
             SITENAME,
             net,
             LAYER2ID,
             REGION_NAME,
             LAYER3ID,
             COUNTY_NAME,

             sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
             sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
             sum(THROUGHPUT) THROUGHPUT
        from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
                                       TO_DATE('1970-01-01 08:00:00',
                                               'YYYY-MM-DD HH24:MI:SS'),
                                       'yyyy-mm-dd hh24'),
                               'yyyy-mm-dd hh24')*\ sdate,
                    t.ran_ne_id,
                    gnbid,
                    sitename,
                    cgisai,
```

```
                    --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
                    --case when length(cgisai)=12 then substr(cgisai,6,5)
else substr(cgisai,6,4) end enbid,
                    --case when length(cgisai)=12 then
substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
                    to_number((case
                                when length(cgisai) = 12 then
                                 substr(cgisai, 6, 5)
                                else
                                 substr(cgisai, 6, 4)
                              end),
                              'XXXXXXXX') enbid,
                    to_number((case
                                when length(cgisai) = 12 then
                                 substr(cgisai, -2)
                                else
                                 substr(cgisai, -4)
                              end),
                              'XXXXXXXX') cell_id,

                    \*
IF(LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))
IF(LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
                    rat,
                    CASE
                      WHEN rat = 9 THEN
                       '5G'
                      WHEN RAT = 6 THEN
                       '4G'
                    END NET,
                    l4_dw_throughput,
                    l4_ul_throughput,
                    (l4_dw_throughput + l4_ul_throughput) throughput,
                    layer2id,
                    d.region_name,
                    layer3id,
                    h.county_name
               FROM sdr_flow_cell_source_csv t
               LEFT JOIN sdr_flow_region_code d
                 ON t.layer2id = d.region_id
               LEFT JOIN sdr_flow_mapping_ran_ne_id g
                 ON t.ran_ne_id = g.ran_ne_id
               LEFT JOIN SDR_FLOW_county_code h
                 ON t.layer3id = h.county_id
              WHERE T.RAT = 9
              and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
      group by sdate,
               RAN_NE_ID,
               GNBID,
               SITENAME,
               net,
               LAYER2ID,
               REGION_NAME,
               LAYER3ID,
               COUNTY_NAME;

```

```
419     end;*/
420
421
422     --6 5g 基站级天粒度流量
423     /*procedure sdr_flow_5g_enb_day(i_sdate varchar2) is
424
425     begin
426       delete from sdr_flow_5g_enb_day where trunc(sdate) =
    to_date(i_sdate,'yyyymmdd');
427       insert into sdr_flow_5g_enb_day
428       select trunc(sdate) sdate,
429              RAN_NE_ID,
430              GNBID,
431              SITENAME,
432              net,
433              LAYER2ID,
434              REGION_NAME,
435              LAYER3ID,
436              COUNTY_NAME,
437
438              sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
439              sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
440              sum(THROUGHPUT) THROUGHPUT
441         from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
442                                       TO_DATE('1970-01-01 08:00:00',
443                                               'YYYY-MM-DD HH24:MI:SS'),
444                                       'yyyy-mm-dd hh24'),
445                               'yyyy-mm-dd hh24')*\ sdate,
446                      t.ran_ne_id,
447                      gnbid,
448                      sitename,
449                      cgisai,
450                      --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
451                      --case when length(cgisai)=12 then substr(cgisai,6,5)
    else substr(cgisai,6,4) end enbid,
452                      --case when length(cgisai)=12 then
    substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
453                      to_number((case
454                                  when length(cgisai) = 12 then
455                                   substr(cgisai, 6, 5)
456                                  else
457                                   substr(cgisai, 6, 4)
458                                end),
459                                'XXXXXXXX') enbid,
460                      to_number((case
461                                  when length(cgisai) = 12 then
462                                   substr(cgisai, -2)
463                                  else
464                                   substr(cgisai, -4)
465                                end),
466                                'XXXXXXXX') cell_id,
467
468                      \*
469
    IF(LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))
470
    IF(LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
471                      rat,
```

```
                    CASE
                      WHEN rat = 9 THEN
                       '5G'
                      WHEN RAT = 6 THEN
                       '4G'
                    END NET,
                    l4_dw_throughput,
                    l4_ul_throughput,
                    (l4_dw_throughput + l4_ul_throughput) throughput,
                    layer2id,
                    d.region_name,
                    layer3id,
                    h.county_name
               FROM sdr_flow_cell_source_csv t
               LEFT JOIN sdr_flow_region_code d
                 ON t.layer2id = d.region_id
               LEFT JOIN sdr_flow_mapping_ran_ne_id g
                 ON t.ran_ne_id = g.ran_ne_id
               LEFT JOIN SDR_FLOW_county_code h
                 ON t.layer3id = h.county_id
              WHERE T.RAT = 9
              and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
       group by trunc(sdate),
                RAN_NE_ID,
                GNBID,
                SITENAME,
                net,
                LAYER2ID,
                REGION_NAME,
                LAYER3ID,
                COUNTY_NAME;

    end;*/
    procedure ini_4g_table is
    begin

      dbms_output.put_line('start reset table ...');

      execute immediate 'truncate table stats_4g_cell_flow_2m_day';
      execute immediate 'truncate table stats_4g_cell_flow_48h_hour';
      execute immediate 'truncate table stats_4g_cell_flow_7d_day';
      execute immediate 'truncate table stats_4g_colleges_siteinfo';
      execute immediate 'truncate table stats_4g_error_siteinfo';
      execute immediate 'truncate table stats_4g_siteinfo_deviate';
      execute immediate 'truncate table stats_4g_site_flow_2m_day';
      execute immediate 'truncate table stats_4g_site_flow_48h_hour';
      execute immediate 'truncate table stats_4g_site_flow_7d_day';
      execute immediate 'truncate table STATS_4g_SITE_LOWFLOW_7d_DAY';
      execute immediate 'truncate table STATS_4g_CELL_LOWFLOW_7d_DAY';

      dbms_output.put_line('reset table end...');

    end;

    procedure ini_5g_table is
    begin

      dbms_output.put_line('start reset table ...');
```

```
530
531       execute immediate 'truncate table sdr_flow_5g_cell_day';
532       execute immediate 'truncate table sdr_flow_5g_enb_day';
533       execute immediate 'truncate table stats_5g_cell_flow_2m_day';
534       execute immediate 'truncate table stats_5g_cell_flow_48h_hour';
535       execute immediate 'truncate table stats_5g_cell_flow_7d_day';
536       execute immediate 'truncate table stats_5g_colleges_siteinfo';
537       execute immediate 'truncate table stats_5g_error_siteinfo';
538       execute immediate 'truncate table stats_5g_physite_flow_2m_day';
539       execute immediate 'truncate table stats_5g_physite_flow_48h_hour';
540       execute immediate 'truncate table stats_5g_physite_flow_7d_day';
541       execute immediate 'truncate table stats_5g_site_flow_2m_day';
542       execute immediate 'truncate table stats_5g_site_flow_48h_hour';
543       execute immediate 'truncate table stats_5g_site_flow_7d_day';
544       execute immediate 'truncate table STATS_5g_SITE_LOWFLOW_7d_DAY';
545       execute immediate 'truncate table STATS_5g_CELL_LOWFLOW_7d_DAY';
546       execute immediate 'truncate table STATS_5g_PHYSITE_LOFLOW_7d_DAY';
547
548       dbms_output.put_line('reset table end...');
549
550     end;
551
552     procedure seq_4g_siteinfo_backup is
553     begin
554
555       dbms_output.put_line('backup seq_4g_siteinfo ...');
556
557       insert into seq_4g_siteinfo_backup
558       select * from seq_4g_siteinfo;
559       commit;
560
561       execute immediate 'truncate table seq_4g_siteinfo';
562
563       dbms_output.put_line('backup seq_4g_siteinfo end...');
564
565     end;
566
567     procedure seq_5g_siteinfo_backup is
568     begin
569
570       dbms_output.put_line('backup seq_5g_siteinfo ...');
571
572       insert into seq_5g_siteinfo_backup
573       select * from seq_5g_siteinfo;
574       commit;
575
576       execute immediate 'truncate table seq_5g_siteinfo';
577
578       dbms_output.put_line('backup seq_5g_siteinfo end...');
579
580     end;
581
582     procedure cal_4g_data is
583       istart date;
584       iend   date;
585       igap   number;
586       sdate varchar2(8):='';
587     begin
```

```
      --获得小区流量源表中所有地市的最小日期
      select min(sdate) sdate_min
        into istart
        from (select  max(sdate) sdate from stats_4g_cell_flow_2m_day
              union
              select to_date(max(sdate),'yyyymmdd') sdate from
sdr_flow_4g_enb_day
              union
              select trunc(sysdate) from dual
              );
      dbms_output.put_line('cal_4g_data: start date - ' || to_char(istart,
'yyyymmdd'));
      --获得结果表的最小日期
      select min(to_date(sdate, 'yyyymmdd')) sdate_max
        into iend
        from (select max(sdate) sdate from sdr_flow_4g_cell_day
              union
              select max(sdate) sdate from sdr_flow_4g_enb_day
              union
              select to_char(sysdate,'yyyymmdd') from dual
             );
      dbms_output.put_line('cal_4g_data: end date - ' || to_char(iend,
'yyyymmdd'));
      --dbms_output.put_line(istart);
      --dbms_output.put_line(iend);
      --计算时间差，用于实现遍历的时间列表
      igap := iend - istart;
      --dbms_output.put_line(igap);
      if igap >=1 then
        for v in (select istart + level sdate from dual connect by level
<= igap) loop

          dbms_output.put_line('cal_4g_data: date - ' || to_char(v.sdate,
'yyyymmdd') || ' start process');

          STATS_4g_SITE_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
          STATS_4g_CELL_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);

          STATS_4g_SITE_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);
          STATS_4g_CELL_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);

          STATS_4g_SITE_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);
          STATS_4g_CELL_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);

          STATS_4g_SITE_LOWFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
          STATS_4g_CELL_LOWFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
        end loop;
      end if;
      -- 执行工参异常检测
      dbms_output.put_line('check_45site: 0 -
STATS_4G_ERROR_SITEINFO/STATS_4G_COLLEGES_SITEINFO - ' || sdate || ' start
process');
      select to_char(max(sdate), 'yyyymmdd') into sdate from
seq_4g_siteinfo;
      STATS_4G_ERROR_SITEINFO(sdate);
      STATS_4G_COLLEGES_SITEINFO(sdate,50,50);
      -- 执行工参基站偏离距离计算
```

```
637       select to_char(max(sdate), 'yyyymmdd') into sdate from
SEQ_MDT_GRID50_CELL;
638       STATS_4G_DEVIATE_SUMMARY_DB(sdate);
639       STATS_4G_DEVIATE_SUMMARY_QL(sdate);
640
641     end;
642
643
644     procedure cal_5g_data is
645       istart date;
646       iend   date;
647       igap   number;
648       sdate varchar2(8):='';
649     begin
650       --获得小区流量源表中所有地市的最小日期
651       select min(sdate) sdate_max
652         into istart
653         from (select max(sdate) sdate from (select city_name, max(sdate)
sdate from STATS_5g_CELL_FLOW_2M_DAY group by city_name)
654               union
655               select trunc(sysdate) from dual
656               );
657       dbms_output.put_line('cal_5g_data: start date - ' || to_char(istart,
'yyyymmdd'));
658       select min(sdate) sdate_min
659         into iend
660         from (select city, max(sdate) sdate from SDR_FLOW_CELL_THROUGHPUT
group by city
661               union
662               select 'system',trunc(sysdate) from dual
663              );
664       dbms_output.put_line('cal_5g_data: end date - ' || to_char(iend,
'yyyymmdd'));
665       --获得结果表的最小日期
666       --dbms_output.put_line(istart);
667       --dbms_output.put_line(iend);
668       --计算时间差，用于实现遍历的时间列表
669       igap := iend - istart;
670       --dbms_output.put_line(igap);
671       if igap >=1 then
672         for v in (select istart + level sdate from dual connect by level
<= igap) loop
673           dbms_output.put_line('cal_5g_data: date - ' || to_char(v.sdate,
'yyyymmdd') || ' start process');
674
675           sdr_flow_5g_cell_day(to_char(v.sdate, 'yyyymmdd'));
676           sdr_flow_5g_enb_day(to_char(v.sdate, 'yyyymmdd'));
677           ----------
678           STATS_5g_SITE_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
679           STATS_5g_phySITE_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
680           STATS_5g_CELL_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
681           ----------
682           STATS_5g_SITE_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);
683           STATS_5g_phySITE_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);
684           STATS_5g_CELL_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);
685           ----------
686           STATS_5g_SITE_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);
687           STATS_5g_phySITE_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);
```

```
688           STATS_5g_CELL_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);
689
690           STATS_5g_SITE_LOWFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
691           STATS_5g_CELL_LOWFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
692           STATS_5g_PHYSITE_LOFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
693
694         end loop;
695       end if;
696       -- 执行工参异常检测
697       select to_char(max(sdate), 'yyyymmdd') into sdate from
    seq_5g_siteinfo;
698       dbms_output.put_line('check_45site: 3 -
    STATS_5G_ERROR_SITEINFO/STATS_5G_COLLEGES_SITEINFO - ' || sdate || ' start
    process');
699       STATS_5G_ERROR_SITEINFO(sdate);
700       STATS_5G_COLLEGES_SITEINFO(sdate,50,50);
701     end;
702
703     procedure STATS_4G_SDRFLOW(i_sdate     varchar2 default null) is--mdt
    数据中的日期
704                                   istart varchar2(8);
705     begin
706       istart:='20210720';
707
708       if i_sdate is null then
709         select min(sdate) into istart
710         from (select max(sdate) sdate
711                 from sdr_flow_4g_cell_day
712               union
713               select max(sdate) sdate
714                 from sdr_flow_4g_enb_day
715               union
716               select to_char(sysdate, 'yyyymmdd')
717                 from dual);
718       else
719         istart := i_sdate;
720       end if;
721
722       dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
    istart=' || istart);
723       dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
    start');
724
725       STATS_4g_SITE_FLOW_2M_DAY(istart, 6);
726       STATS_4g_CELL_FLOW_2M_DAY(istart, 6);
727       STATS_4g_SITE_FLOW_7d_DAY(istart, 1);
728       STATS_4g_CELL_FLOW_7d_DAY(istart, 1);
729       STATS_4g_SITE_FLOW_48h_hour(istart, 0);
730       STATS_4g_CELL_FLOW_48h_hour(istart, 0);
731       STATS_4g_SITE_LOWFLOW_7d_DAY(istart, 6);
732       STATS_4g_CELL_LOWFLOW_7d_DAY(istart, 6);
733
734       dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
    end');
735
736     end;
737
```

```
    procedure STATS_5G_OMCFLOW(i_sdate    varchar2 default null) is --5G工参中的最新日期
                                    istart varchar2(8);
    begin
      istart:='20210720';

      if i_sdate is null then
      select to_char(min(sdate), 'yyyymmdd')
        into istart
        from (select city, max(sdate) sdate from SDR_FLOW_CELL_THROUGHPUT group by city
              union
              select 'system', trunc(sysdate) from dual);
      else
        istart := i_sdate;
      end if;

      dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' : istart=' || istart);
      dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' : start');

      sdr_flow_5g_cell_day(istart);
      sdr_flow_5g_enb_day(istart);
      ----------
      STATS_5g_SITE_FLOW_2M_DAY(istart, 6);
      STATS_5g_phySITE_FLOW_2M_DAY(istart, 6);
      STATS_5g_CELL_FLOW_2M_DAY(istart, 6);
      ----------
      STATS_5g_SITE_FLOW_7d_DAY(istart, 1);
      STATS_5g_phySITE_FLOW_7d_DAY(istart, 1);
      STATS_5g_CELL_FLOW_7d_DAY(istart, 1);
      ----------
      STATS_5g_SITE_FLOW_48h_hour(istart, 0);
      STATS_5g_phySITE_FLOW_48h_hour(istart, 0);
      STATS_5g_CELL_FLOW_48h_hour(istart, 0);

      STATS_5g_SITE_LOWFLOW_7d_DAY(istart, 6);
      STATS_5g_CELL_LOWFLOW_7d_DAY(istart, 6);
      STATS_5g_PHYSITE_LOFLOW_7d_DAY(istart, 6);
      dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' : end');

    end;

    --7 5g 小区级天粒度流量
    -- 20210726 不在使用
    procedure SDR_FLOW_5G_CELL_DAY(i_sdate varchar2) is

    begin
      delete from sdr_flow_5g_cell_day where sdate =i_sdate;
      insert into sdr_flow_5g_cell_day
      select to_char(SDATE,'yyyymmdd') SDATE,
             GNBID,
             CELL_ID,
             CITY,
             DISTRICT,
             VENDER,
```

```
791             sum(UL_THROUGHPUT_M) UL_THROUGHPUT_M,
792             sum(DL_THROUGHPUT_M) DL_THROUGHPUT_M,
793             sum(UL_THROUGHPUT_M + DL_THROUGHPUT_M) THROUGHPUT
794        from SDR_FLOW_CELL_THROUGHPUT t
795       where SDATE = to_date(i_sdate,'yyyymmdd')
796       group by SDATE, GNBID, CELL_ID, CITY, DISTRICT, VENDER;
797       commit;
798
799    end;
800
801    -- 20210726 不在使用
802    procedure SDR_FLOW_5G_ENB_DAY(i_sdate varchar2) is
803
804    begin
805      delete from sdr_flow_5g_enb_day where sdate = i_sdate;
806      insert into sdr_flow_5g_enb_day
807      select to_char(SDATE,'yyyymmdd') SDATE,
808             GNBID,
809             CITY,
810             DISTRICT,
811             VENDER,
812             sum(UL_THROUGHPUT_M) UL_THROUGHPUT_M,
813             sum(DL_THROUGHPUT_M) DL_THROUGHPUT_M,
814             sum(UL_THROUGHPUT_M + DL_THROUGHPUT_M) THROUGHPUT
815        from SDR_FLOW_CELL_THROUGHPUT t
816       where SDATE = to_date(i_sdate,'yyyymmdd')
817       group by SDATE, GNBID, CITY, DISTRICT, VENDER;
818       commit;
819
820    end;
821
822
823    procedure STATS_4g_SITE_FLOW_2M_DAY(i_sdate varchar2,idays number) is
824    --idays =29 表示一个月
825    --ilimits，idays+1指定的天数中，累积多少天有流量才认为在指定天数中有流量
826    begin
827      delete from STATS_4g_SITE_FLOW_2M_DAY where trunc(sdate) =
    to_date(i_sdate,'yyyymmdd');
828
829      insert into STATS_4g_SITE_FLOW_2M_DAY
830        select to_date(i_sdate,'yyyymmdd')  sdate,
831               null CITY_CODE,
832               null CITY_NAME,
833               enbid,
834               sum(nvl(THROUGHPUT,0)) THROUGHPUT
835          FROM sdr_flow_4g_enb_day
836         where to_date(sdate,'yyyymmdd') between to_date(i_sdate,
    'yyyymmdd') - idays and to_date(i_sdate, 'yyyymmdd') --29 时间参数
837          group by enbid
838          having sum(nvl(THROUGHPUT,0)) > 0;
839
840        commit;
841    end;
842
843
844
845    procedure STATS_4g_CELL_FLOW_2M_DAY(i_sdate varchar2,idays   number)
    is
```

```
846
847     begin
848       delete from STATS_4g_CELL_FLOW_2M_DAY where trunc(sdate) =
      to_date(i_sdate, 'yyyymmdd');
849
850       insert into STATS_4g_CELL_FLOW_2M_DAY
851         select to_date(i_sdate, 'yyyymmdd') sdate,
852                null CITY_CODE,
853                null CITY_NAME,
854                enbid,
855                cell_id,
856                sum(nvl(THROUGHPUT,0)) THROUGHPUT
857           FROM sdr_flow_4g_cell_day
858          where to_date(sdate, 'yyyymmdd') between to_date(i_sdate,
      'yyyymmdd') - idays and to_date(i_sdate, 'yyyymmdd') --29 时间参数
859           group by enbid, cell_id
860           having sum(nvl(THROUGHPUT,0)) > 0;
861
862       commit;
863       -- 为工参基站中的最新数据打上有话务标签
864       -- 由于部分地市上报地市名称不统一，导致统计出错，因此去除地市的关联条件
      20210420
865       update seq_4g_siteinfo a
866          set a.is_alive = 0, a.is_alive_update_time = null
867        where sdate = (select max(sdate) from seq_4g_siteinfo);
868       commit;
869
870       merge into (select *
871                     from seq_4g_siteinfo
872                    where sdate = (select max(sdate) from seq_4g_siteinfo)
873                    --20210721 新增，地市名称不合法的不输出，否则亚新侧无法正常按照
      报表汇总
874                    and city_name in ('承德市','邯郸市','廊坊市','石家庄市','秦
      皇岛市','张家口市','邢台市','保定市','沧州市','衡水市','唐山市','雄安新区')
875                    ) a
876       using (select distinct city_name, enbid, cell_id
877                 from STATS_4g_CELL_FLOW_2M_DAY
878                where sdate =
879                      (select max(sdate) from STATS_4g_CELL_FLOW_2M_DAY)) b
880       on (/*a.city_name = b.city_name and*/ a.enbid = b.enbid and
      a.cell_id = b.cell_id)
881       when matched then
882         update set a.is_alive = 1, a.is_alive_update_time = sysdate;
883       commit;
884
885     end;
886
887
888
889
890     procedure STATS_5g_SITE_FLOW_2M_DAY(i_sdate varchar2,idays number ) is
891
892     begin
893       delete from STATS_5g_SITE_FLOW_2M_DAY where trunc(sdate) =
      to_date(i_sdate,'yyyymmdd');
894
895       insert into STATS_5g_SITE_FLOW_2M_DAY
896       select to_date(i_sdate, 'yyyymmdd') sdate,
```

```
897             null CITY_CODE,
898             null CITY_NAME,
899             gnbid,
900             sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,0)) THROUGHPUT
901        FROM SDR_FLOW_CELL_THROUGHPUT
902      where sdate between to_date(i_sdate, 'yyyymmdd') - idays and
to_date(i_sdate, 'yyyymmdd') --29 时间参数
903        group by gnbid
904        having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,0))>0;
905      commit;
906
907    end;
908
909
910
911    procedure STATS_5g_CELL_FLOW_2M_DAY(i_sdate varchar2,
912                                        idays   number) is
913
914    begin
915      delete from STATS_5g_CELL_FLOW_2M_DAY where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
916
917      insert into STATS_5g_CELL_FLOW_2M_DAY
918        select to_date(i_sdate, 'yyyymmdd') sdate,
919               null CITY_CODE,
920               null CITY_NAME,
921               GNBID,
922               cell_id,
923               sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,0)) THROUGHPUT
924          FROM SDR_FLOW_CELL_THROUGHPUT
925            where sdate between to_date(i_sdate, 'yyyymmdd') - idays and
to_date(i_sdate, 'yyyymmdd') --29 时间参数
926            group by GNBID, CELL_ID
927            having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,0)) > 0;
928      commit;
929
930      -- 为工参基站中的最新数据打上有话务标签
931      -- 由于部分地市上报地市名称不统一，导致统计出错，因此去除地市的关联条件
20210420
932      update seq_5g_siteinfo a
933         set a.is_alive = 0, a.is_alive_update_time = null
934       where sdate = (select max(sdate) from seq_5g_siteinfo);
935      commit;
936      merge into (select *
937                    from seq_5g_siteinfo
938                   where sdate = (select max(sdate) from seq_5g_siteinfo)
939                   --20210721 新增，地市名称不合法的不输出，否则亚新侧无法正常按照
报表汇总
940                   and city_name in ('承德市','邯郸市','廊坊市','石家庄市','秦
皇岛市','张家口市','邢台市','保定市','沧州市','衡水市','唐山市','雄安新区')
941                   ) a
942      using (select distinct gnbid, cell_id
943               from STATS_5g_CELL_FLOW_2M_DAY
944              where sdate =
945                    (select max(sdate) from STATS_5g_CELL_FLOW_2M_DAY)) b
946      on (/*a.city_name = b.city_name and */a.gnbid = b.gnbid and
a.cell_id = b.cell_id)
947      when matched then
```

```
948         update set a.is_alive = 1, a.is_alive_update_time = sysdate;
949       commit;
950     end;
951
952     procedure STATS_4g_SITE_FLOW_7d_DAY(i_sdate varchar2,idays number) is
953
954     begin
955       delete from STATS_4g_SITE_FLOW_7d_DAY where trunc(sdate) =
    to_date(i_sdate,'yyyymmdd');
956       insert into STATS_4g_SITE_FLOW_7d_DAY
957       select SDATE, CITY_CODE, CITY_NAME, ENBID --, BENBID
958        from (select to_date(i_sdate, 'yyyymmdd') sdate,
959                     a.CITY_CODE,
960                     a.CITY_NAME,
961                     a.enbid,
962                     b.enbid benbid
963                from (SELECT distinct CITY_CODE, CITY_NAME, ENBID
964                        FROM seq_4g_siteinfo  where sdate = (select
    max(sdate) from seq_4g_siteinfo)) a
965                left join (SELECT enbid
966                             FROM sdr_flow_4g_enb_day
967                            where to_date(sdate,'yyyymmdd') between
968                                  to_date(i_sdate, 'yyyymmdd') - idays and
969                                  to_date(i_sdate, 'yyyymmdd') --29 时间参数
970                            group by enbid
971                           having sum(nvl(THROUGHPUT,0)) > 0
972                            ) b
973                  on a.enbid = b.enbid)
974       where benbid is null;
975       commit;
976
977
978     end;
979
980
981
982
983     procedure STATS_4g_CELL_FLOW_7d_DAY(i_sdate varchar2,idays number) is
984
985     begin
986       delete from STATS_4g_CELL_FLOW_7d_DAY where trunc(sdate) =
    to_date(i_sdate,'yyyymmdd');
987       insert into STATS_4g_CELL_FLOW_7d_DAY
988       select SDATE, CITY_CODE, CITY_NAME, ENBID, CELL_ID--, BENBID,
    BCELL_ID
989        from (select to_date(i_sdate, 'yyyymmdd') sdate,
990                     a.CITY_CODE,
991                     a.CITY_NAME,
992                     a.enbid,
993                     a.cell_id,
994                     b.enbid benbid,
995                     b.cell_id bcell_id
996                from (SELECT distinct CITY_CODE, CITY_NAME, ENBID, cell_id
997                        FROM seq_4g_siteinfo where sdate = (select
    max(sdate) from seq_4g_siteinfo)) a
998                left join (SELECT enbid, cell_id
999                             FROM sdr_flow_4g_cell_day
1000                           where to_date(sdate,'yyyymmdd') between
```

```
                                to_date(i_sdate, 'yyyymmdd') - idays and
                                to_date(i_sdate, 'yyyymmdd') --29 时间参数
                          group by enbid, cell_id
                         having sum(nvl(THROUGHPUT,0)) > 0
                         ) b
                  on a.enbid = b.enbid
                 and a.cell_id = b.cell_id)
       where benbid is null;
       commit;


    end;




    procedure STATS_5g_SITE_FLOW_7d_DAY(i_sdate varchar2,idays number) is

    begin
       delete from STATS_5g_SITE_FLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
       insert into STATS_5g_SITE_FLOW_7d_DAY
       select SDATE, CITY_CODE, CITY_NAME, GNBID --, BGNBID
        from (select to_date(i_sdate, 'yyyymmdd') sdate,
                     a.CITY_CODE,
                     a.CITY_NAME,
                     a.gnbid,
                     b.gnbid bgnbid
                from (SELECT distinct CITY_CODE, CITY_NAME, GNBID
                        FROM seq_5g_siteinfo  where sdate = (select
max(sdate) from seq_5g_siteinfo)) a
                left join (SELECT gnbid
                            FROM SDR_FLOW_CELL_THROUGHPUT
                           where sdate between
                                to_date(i_sdate, 'yyyymmdd') - idays and
                                to_date(i_sdate, 'yyyymmdd') --29 时间参数
                          group by gnbid
                         having
sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,0)) > 0
                         ) b
                  on a.gnbid = b.gnbid)
       where bgnbid is null;
       commit;

    end;

    procedure STATS_5g_CELL_FLOW_7d_DAY(i_sdate varchar2,idays number) is

    begin
       delete from STATS_5g_CELL_FLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
       insert into STATS_5g_CELL_FLOW_7d_DAY
       select distinct SDATE, CITY_CODE, CITY_NAME, GNBID, CELL_ID--,
BGNBID, BCELL_ID
        from (select to_date(i_sdate, 'yyyymmdd') sdate,
                     a.CITY_CODE,
                     a.CITY_NAME,
                     a.GNBID,
```

```
                    a.cell_id,
                    b.GNBID bgnbid,
                    b.cell_id bcell_id
               from (SELECT distinct CITY_CODE, CITY_NAME, GNBID, CELL_ID
                       FROM seq_5g_siteinfo  where sdate = (select
max(sdate) from seq_5g_siteinfo)) a
               left join (SELECT GNBID, CELL_ID
                           FROM SDR_FLOW_CELL_THROUGHPUT
                          where sdate between
                                to_date(i_sdate, 'yyyymmdd') - idays and
                                to_date(i_sdate, 'yyyymmdd') --29 时间参数
                          group by GNBID, CELL_ID
                         having
sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,0)) > 0
                         ) b
                 on a.GNBID = b.GNBID
                and a.cell_id = b.cell_id)
      where bgnbid is null;
      commit;

    end;


    procedure STATS_4g_SITE_LOWFLOW_7d_DAY(i_sdate varchar2,idays number)
is

    begin
      delete from STATS_4g_SITE_LOWFLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
      insert into STATS_4g_SITE_LOWFLOW_7d_DAY
        select to_date(i_sdate, 'yyyymmdd') sdate,
               a.CITY_CODE,
               a.CITY_NAME,
               a.enbid,
               b.THROUGHPUT
          from (SELECT distinct CITY_CODE, CITY_NAME, ENBID
                  FROM seq_4g_siteinfo
                 where sdate = (select max(sdate) from seq_4g_siteinfo)) a
          inner join (SELECT enbid, sum(nvl(THROUGHPUT, 0)) THROUGHPUT
                       FROM sdr_flow_4g_enb_day
                      where --nvl(THROUGHPUT,0) > 0 and nvl(THROUGHPUT,0)
<7516192768 -- 低业务量：基站按<7GB计算，sdr单位为byte，因此需除以3个1024
                     /* and */
                      to_date(sdate, 'yyyymmdd') between
                      to_date(i_sdate, 'yyyymmdd') - idays and
                      to_date(i_sdate, 'yyyymmdd') --29 时间参数
                      group by enbid
                     having sum(nvl(THROUGHPUT, 0)) > 0 and
sum(nvl(THROUGHPUT, 0)) < 7516192768) b
            on a.enbid = b.enbid;
      commit;


    end;
```

```
1106      procedure STATS_4g_CELL_LOWFLOW_7d_DAY(i_sdate varchar2,idays number)
is
1107
1108      begin
1109        delete from STATS_4g_CELL_LOWFLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
1110        insert into STATS_4g_CELL_LOWFLOW_7d_DAY
1111          select to_date(i_sdate, 'yyyymmdd') sdate,
1112                 a.CITY_CODE,
1113                 a.CITY_NAME,
1114                 a.enbid,
1115                 a.cell_id,
1116                 b.THROUGHPUT
1117            from (SELECT distinct CITY_CODE, CITY_NAME, ENBID, cell_id
1118                    FROM seq_4g_siteinfo
1119                   where sdate = (select max(sdate) from seq_4g_siteinfo)) a
1120            inner join (SELECT enbid,
1121                               cell_id,
1122                               sum(nvl(THROUGHPUT, 0)) THROUGHPUT
1123                          FROM sdr_flow_4g_cell_day
1124                         where --nvl(THROUGHPUT,0) > 0 and nvl(THROUGHPUT,0)
<2147483648 -- 低业务量：扇区按<2GB计算，sdr单位为byte，因此需除以3个1024
1125                        /* and*/
1126                          to_date(sdate, 'yyyymmdd') between
1127                          to_date(i_sdate, 'yyyymmdd') - idays and
1128                          to_date(i_sdate, 'yyyymmdd') --29 时间参数
1129                          group by enbid, cell_id
1130                        having sum(nvl(THROUGHPUT, 0)) > 0 and
sum(nvl(THROUGHPUT, 0)) < 2147483648) b
1131              on a.enbid = b.enbid
1132             and a.cell_id = b.cell_id;
1133        commit;
1134
1135      end;
1136
1137      -- 5G 业务量单位为MByte，因此换单单位>7GB，为>7*1024
1138      procedure STATS_5g_SITE_LOWFLOW_7d_DAY(i_sdate varchar2,idays number)
is
1139
1140      begin
1141        delete from STATS_5g_SITE_LOWFLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
1142        insert into STATS_5g_SITE_LOWFLOW_7d_DAY
1143          select to_date(i_sdate, 'yyyymmdd') sdate,
1144                 a.CITY_CODE,
1145                 a.CITY_NAME,
1146                 a.gnbid,
1147                 b.THROUGHPUT
1148            from (SELECT distinct CITY_CODE, CITY_NAME, GNBID
1149                    FROM seq_5g_siteinfo
1150                   where sdate = (select max(sdate) from seq_5g_siteinfo)) a
1151           inner join (SELECT gnbid,
sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) THROUGHPUT
1152                        FROM SDR_FLOW_CELL_THROUGHPUT
1153                       where sdate between
1154                             to_date(i_sdate, 'yyyymmdd') - idays and
1155                             to_date(i_sdate, 'yyyymmdd') --29 时间参数
1156                       group by gnbid
```

```
                    having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) >
0 and sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) < 7168) b
            on a.gnbid = b.gnbid;
      commit;

    end;

    procedure STATS_5g_CELL_LOWFLOW_7d_DAY(i_sdate varchar2,idays number)
is

    begin
      delete from STATS_5g_CELL_LOWFLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
      insert into STATS_5g_CELL_LOWFLOW_7d_DAY
        select to_date(i_sdate, 'yyyymmdd') sdate,
               a.CITY_CODE,
               a.CITY_NAME,
               a.GNBID,
               a.cell_id,
               b.THROUGHPUT
          from (SELECT distinct CITY_CODE, CITY_NAME, GNBID, CELL_ID
                  FROM seq_5g_siteinfo
                 where sdate = (select max(sdate) from seq_5g_siteinfo)) a
          inner join (SELECT GNBID,
                             CELL_ID,
                             sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0))
THROUGHPUT
                       FROM SDR_FLOW_CELL_THROUGHPUT
                      where sdate between
                      to_date(i_sdate, 'yyyymmdd') - idays and
                      to_date(i_sdate, 'yyyymmdd') --29 时间参数
                      group by GNBID, CELL_ID
                     having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) >
0 and sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) < 2048) b
            on a.GNBID = b.GNBID
           and a.cell_id = b.cell_id;
      commit;

    end;


    procedure STATS_5g_PHYSITE_FLOW_2M_DAY(i_sdate varchar2,idays number)
is

    begin
      delete from STATS_5g_PHYSITE_FLOW_2M_DAY where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');

      insert into STATS_5g_PHYSITE_FLOW_2M_DAY
      SELECT DISTINCT TO_DATE(i_sdate, 'yyyymmdd') SDATE,
                      A.CITY_CODE,
                      A.CITY_NAME,
                      A.PHYSTATION_ADDRESS,
                      A.GNBID,
                      B.THROUGHPUT
        FROM (SELECT DISTINCT CITY_CODE, CITY_NAME, PHYSTATION_ADDRESS,
GNBID
                FROM SEQ_5G_SITEINFO
```

```
                WHERE SDATE = (SELECT MAX(SDATE) FROM SEQ_5G_SITEINFO)) A
        INNER JOIN (SELECT D.CITY_NAME,
                           D.GNBID,
                           D.PHYSTATION_ADDRESS,
                           sum(THROUGHPUT) THROUGHPUT
                      FROM (select gnbid,
                                   cell_id,
                                   SUM(NVL(UL_THROUGHPUT_M +
DL_THROUGHPUT_M, 0)) THROUGHPUT
                              from SDR_FLOW_CELL_THROUGHPUT
                             where SDATE BETWEEN
                                   TO_DATE(i_sdate, 'yyyymmdd') - idays AND
                                   TO_DATE(i_sdate, 'yyyymmdd')
                             group by gnbid, cell_id) T,
                           (SELECT distinct CITY_NAME,
                                            PHYSTATION_ADDRESS,
                                            gnbid,
                                            cell_id
                              FROM SEQ_5G_SITEINFO
                             where SDATE =
                                   (SELECT MAX(SDATE) FROM
SEQ_5G_SITEINFO)) D
                     WHERE T.GNBID = D.GNBID
                       AND T.CELL_ID = D.CELL_ID
                     GROUP BY D.CITY_NAME, D.GNBID, D.PHYSTATION_ADDRESS
                    HAVING sum(THROUGHPUT) > 0) B
           ON A.GNBID = B.GNBID
          AND A.CITY_NAME = B.CITY_NAME
          AND A.PHYSTATION_ADDRESS = B.PHYSTATION_ADDRESS;
        commit;

    end;

    procedure STATS_5g_PHYSITE_FLOW_7d_DAY(i_sdate varchar2,idays number)
is

    begin
      delete from STATS_5g_PHYSITE_FLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
      insert into STATS_5g_PHYSITE_FLOW_7d_DAY
      select distinct SDATE, CITY_CODE, CITY_NAME, PHYSTATION_ADDRESS,
GNBID
      from (select to_date(i_sdate, 'yyyymmdd') sdate,
                   a.CITY_CODE,
                   a.CITY_NAME,
                   a.PHYSTATION_ADDRESS,
                   a.GNBID,
                   b.GNBID bgnbid
              from (SELECT distinct CITY_CODE,
                                    CITY_NAME,
                                    PHYSTATION_ADDRESS,
                                    GNBID
                      FROM seq_5g_siteinfo
                     where sdate = (select max(sdate) from
seq_5g_siteinfo)) a
              left join (SELECT d.city_name, d.GNBID,
d.PHYSTATION_ADDRESS,sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0))
THROUGHPUT
```

```
                            FROM SDR_FLOW_CELL_THROUGHPUT t,
seq_5g_siteinfo d
                           where t.sdate between
                                 to_date(i_sdate, 'yyyymmdd') - idays and
                                 to_date(i_sdate, 'yyyymmdd') --29 时间参数
                             and d.sdate = (select max(sdate) from
seq_5g_siteinfo)
                             and t.gnbid = d.gnbid
                             and t.cell_id = d.cell_id
                           group by d.city_name, d.GNBID,
d.PHYSTATION_ADDRESS
                          having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,
0)) > 0
                        ) b
                on a.gnbid = b.gnbid
               and a.city_name = b.city_name
               and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS)
     where bgnbid is null;
      commit;

    end;

    procedure STATS_5g_PHYSITE_LOFLOW_7d_DAY(i_sdate varchar2,idays
number) is

    begin
      delete from STATS_5g_PHYSITE_LOFLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
      insert into STATS_5g_PHYSITE_LOFLOW_7d_DAY
        select to_date(i_sdate, 'yyyymmdd') sdate,
               a.CITY_CODE,
               a.CITY_NAME,
               a.PHYSTATION_ADDRESS,
               a.GNBID,
               sum(b.THROUGHPUT) THROUGHPUT
          from (SELECT distinct CITY_CODE,
                                CITY_NAME,
                                PHYSTATION_ADDRESS,
                                GNBID,
                                CELL_ID
                  FROM seq_5g_siteinfo
                 where sdate = (select max(sdate) from seq_5g_siteinfo)) a
         inner join (SELECT GNBID,
                            CELL_ID,
                            sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0))
THROUGHPUT
                       FROM SDR_FLOW_CELL_THROUGHPUT
                      where sdate between to_date(i_sdate, 'yyyymmdd') -
idays and to_date(i_sdate, 'yyyymmdd') --29 时间参数
                      group by GNBID, CELL_ID) b
            on a.gnbid = b.gnbid
           and a.cell_id = b.cell_id
         group by to_date(i_sdate, 'yyyymmdd'),
                  CITY_CODE,
                  CITY_NAME,
                  PHYSTATION_ADDRESS,
                  a.GNBID
```

```
1306        having sum(nvl(THROUGHPUT, 0)) > 0 and sum(nvl(THROUGHPUT, 0)) <
    7168;
1307      commit;
1308
1309    end;
1310
1311    procedure STATS_5g_PHYSITE_FLOW_48h_hour(i_sdate varchar2,idays
    number) is
1312
1313    begin
1314      delete from STATS_5g_PHYSITE_FLOW_48h_hour where trunc(sdate) =
    to_date(i_sdate,'yyyymmdd');
1315      insert into STATS_5g_PHYSITE_FLOW_48h_hour
1316      select distinct SDATE, CITY_CODE, CITY_NAME, PHYSTATION_ADDRESS,
    GNBID
1317      from (select to_date(i_sdate, 'yyyymmdd') sdate,
1318                   a.CITY_CODE,
1319                   a.CITY_NAME,
1320                   a.PHYSTATION_ADDRESS,
1321                   a.GNBID,
1322                   b.GNBID bgnbid
1323              from (SELECT distinct CITY_CODE,
1324                                    CITY_NAME,
1325                                    PHYSTATION_ADDRESS,
1326                                    GNBID
1327                      FROM seq_5g_siteinfo
1328                     where sdate = (select max(sdate) from
    seq_5g_siteinfo)) a
1329              left join (SELECT d.city_name, d.GNBID,
    d.PHYSTATION_ADDRESS,sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0))
    THROUGHPUT
1330                           FROM SDR_FLOW_CELL_THROUGHPUT t,
    seq_5g_siteinfo d
1331                          where t.sdate between
1332                                to_date(i_sdate, 'yyyymmdd') - idays and
1333                                to_date(i_sdate, 'yyyymmdd') --29 时间参数
1334                            and d.sdate = (select max(sdate) from
    seq_5g_siteinfo)
1335                            and t.gnbid = d.gnbid
1336                            and t.cell_id = d.cell_id
1337                          group by d.city_name, d.GNBID,
    d.PHYSTATION_ADDRESS
1338                         having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,
    0)) > 0
1339                        ) b
1340                on a.gnbid = b.gnbid
1341               and a.city_name = b.city_name
1342               and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS)
1343     where bgnbid is null;
1344      commit;
1345
1346    end;
1347
1348
1349    procedure STATS_4g_SITE_FLOW_48h_hour(i_sdate varchar2,idays number)
    is
1350
1351    begin
```

```
      delete from STATS_4g_SITE_FLOW_48h_hour where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
      insert into STATS_4g_SITE_FLOW_48h_hour
      select SDATE, CITY_CODE, CITY_NAME, ENBID --, BENBID
       from (select to_date(i_sdate, 'yyyymmdd') sdate,
                    a.CITY_CODE,
                    a.CITY_NAME,
                    a.enbid,
                    b.enbid benbid
               from (SELECT distinct CITY_CODE, CITY_NAME, ENBID
                       FROM seq_4g_siteinfo  where sdate = (select
max(sdate) from seq_4g_siteinfo)) a
               left join (SELECT enbid
                           FROM sdr_flow_4g_enb_day
                          where to_date(sdate,'yyyymmdd') between
to_date(i_sdate, 'yyyymmdd') - idays and --47/24
                                to_date(i_sdate, 'yyyymmdd') --29 时间参数
                          group by enbid
                         having sum(nvl(THROUGHPUT, 0))>0
                         ) b
                 on a.enbid = b.enbid)
      where benbid is null;
      commit;


    end;




    procedure STATS_4g_CELL_FLOW_48h_hour(i_sdate varchar2,idays number)
is

    begin
      delete from STATS_4g_CELL_FLOW_48h_hour where trunc(sdate) =
trunc(to_date(i_sdate,'yyyymmdd'));
      insert into STATS_4g_CELL_FLOW_48h_hour
      select SDATE, CITY_CODE, CITY_NAME, ENBID, CELL_ID--, BENBID,
BCELL_ID
       from (select trunc(to_date(i_sdate, 'yyyymmdd')) sdate,
                    a.CITY_CODE,
                    a.CITY_NAME,
                    a.enbid,
                    a.cell_id,
                    b.enbid benbid,
                    b.cell_id bcell_id
               from (SELECT distinct CITY_CODE, CITY_NAME, ENBID, cell_id
                       FROM seq_4g_siteinfo where sdate = (select
max(sdate) from seq_4g_siteinfo)) a
               left join (SELECT enbid, cell_id
                           --FROM sdr_flow_4g_cell_hour
                           FROM sdr_flow_4g_cell_day
                          where to_date(sdate,'yyyymmdd') between
to_date(i_sdate, 'yyyymmdd') - idays and --47/24
                                to_date(i_sdate, 'yyyymmdd') --29 时间参数
                          group by enbid, cell_id
                         having sum(nvl(THROUGHPUT, 0))>0
                         ) b
```

```
1402                  on a.enbid = b.enbid
1403                 and a.cell_id = b.cell_id)
1404       where benbid is null;
1405       commit;
1406
1407
1408     end;
1409
1410
1411
1412
1413     procedure STATS_5g_SITE_FLOW_48h_hour(i_sdate varchar2,idays number)
    is
1414
1415     begin
1416       delete from STATS_5g_SITE_FLOW_48h_hour where trunc(sdate) =
    to_date(i_sdate,'yyyymmdd');
1417       insert into STATS_5g_SITE_FLOW_48h_hour
1418       select SDATE, CITY_CODE, CITY_NAME, GNBID --, BGNBID
1419        from (select to_date(i_sdate, 'yyyymmdd') sdate,
1420                     a.CITY_CODE,
1421                     a.CITY_NAME,
1422                     a.gnbid,
1423                     b.gnbid bgnbid
1424                from (SELECT distinct CITY_CODE, CITY_NAME, GNBID
1425                        FROM seq_5g_siteinfo  where sdate = (select
    max(sdate) from seq_5g_siteinfo)) a
1426                left join (SELECT gnbid
1427                            --FROM sdr_flow_5g_enb_hour
1428                            FROM SDR_FLOW_CELL_THROUGHPUT
1429                           where sdate between to_date(i_sdate, 'yyyymmdd')
    - idays and --47/24
1430                                 to_date(i_sdate, 'yyyymmdd') --29 时间参数
1431                           group by gnbid
1432                          having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,
    0))>0
1433                           ) b
1434                  on a.gnbid = b.gnbid)
1435       where bgnbid is null;
1436       commit;
1437
1438     end;
1439
1440
1441     procedure STATS_5g_CELL_FLOW_48h_hour(i_sdate varchar2,idays number)
    is
1442
1443     begin
1444       --5G小区:因为没有5G小区小时级粒度，暂时使用天粒度代替
1445       delete from STATS_5g_CELL_FLOW_48h_hour where trunc(sdate) =
    to_date(i_sdate,'yyyymmdd');
1446       insert into STATS_5g_CELL_FLOW_48h_hour
1447       select SDATE, CITY_CODE, CITY_NAME, GNBID, CELL_ID--, BGNBID,
    BCELL_ID
1448        from (select to_date(i_sdate, 'yyyymmdd') sdate,
1449                     a.CITY_CODE,
1450                     a.CITY_NAME,
1451                     a.GNBID,
```

```
                    a.cell_id,
                    b.GNBID bgnbid,
                    b.cell_id bcell_id
               from (SELECT distinct CITY_CODE, CITY_NAME, GNBID, CELL_ID
                       FROM seq_5g_siteinfo  where sdate = (select
max(sdate) from seq_5g_siteinfo)) a
               left join (SELECT GNBID, CELL_ID
                           FROM SDR_FLOW_CELL_THROUGHPUT
                          where sdate between to_date(i_sdate, 'yyyymmdd')
- idays and --47/24
                                to_date(i_sdate, 'yyyymmdd') --29 时间参数
                          group by GNBID, CELL_ID
                         having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,
0))>0
                          ) b
                 on a.GNBID = b.GNBID
                and a.cell_id = b.cell_id)
      where bgnbid is null;
      commit;


    end;


    procedure STATS_4G_ERROR_SITEINFO(i_sdate varchar2 default null) is
      istart varchar2(8);
    begin
      --delete from STATS_4G_ERROR_SITEINFO where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');

      istart :='20210720';
      --默认值就是对最新的工参进行计算
      if i_sdate is null then
        SELECT to_char(max(sdate),'yyyymmdd') into istart FROM
seq_4g_siteinfo;
      else
        istart :=i_sdate;
      end if;
      dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
istart=' || istart);
      dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
start');

      execute immediate 'truncate table STATS_4G_ERROR_SITEINFO';
      insert into STATS_4G_ERROR_SITEINFO
        (ERROR_CLASS,
         SDATE,
         CITY_CODE,
         CITY_NAME,
         DISTRICT_CODE,
         DISTRICT_NAME,
         NETWORK_NAME,
         PHYSTATION_ADDRESS,
         BBU_NAME,
         ENBID,
         CELL_NAME,
         CELL_ID,
         CGI,
```

```
1503          LON,
1504          LAT,
1505          DIRECTION,
1506          HEIGHT,
1507          M_DOWNTILT,
1508          E_DOWNTILT,
1509          STATION_TYPE,
1510          ISDIGITALINDOOR,
1511          DOWN_FREQ,
1512          VENDER,
1513          OWN_SCHOOLYARD,
1514          TOWERADDRESS_CODE,
1515          PROPERTY,
1516          SCENE,
1517          IS_SCENESITE,
1518          MARKETING_NETWORK,
1519          TERMINALAMOUNT_5G,
1520          SECTOR_INCOMING,
1521          IS_BUSY,
1522          IS_ALIVE,
1523          IS_ALIVE_UPDATE_TIME,
1524          CONSTRUCTION,
1525          IS_SAME_ADDRESS,
1526          SAME_ADDRESS_SITES,
1527          IS_SAME_ADDRESS_CT)
1528          --20210701 更新算法，为了只提取同站存在偏离大于2000米的小区，正常的不提取，
      因此需要改进算法：采用内存表进行计算。
1529          with pysite_bg_2000_4g as (
1530               select distinct t.city_name          city_nameh,
1531                               t.PHYSTATION_ADDRESS PHYSTATION_ADDRESSh,
1532                               t.enbid              enbidh,
1533                               t.lon                lont,
1534                               t.lat                latt,
1535                               h.lon                lonh,
1536                               h.lat                lath
1537                 from (SELECT distinct city_name, PHYSTATION_ADDRESS,
      enbid, lon, lat
1538                         FROM seq_4g_siteinfo
1539                        where STATION_TYPE = '宏站'
1540                          and SDATE = TO_DATE(istart, 'YYYYMMDD')
1541                          and CONSTRUCTION like '%联通%') t,
1542                       (SELECT distinct city_name, PHYSTATION_ADDRESS,
      enbid, lon, lat
1543                         FROM seq_4g_siteinfo
1544                        where STATION_TYPE = '宏站'
1545                          and SDATE = TO_DATE(istart, 'YYYYMMDD')
1546                          and CONSTRUCTION like '%联通%') h
1547               where t.PHYSTATION_ADDRESS = h.PHYSTATION_ADDRESS
1548                 and t.CITY_NAME = h.CITY_NAME
1549                 and t.enbid = H.enbid
1550                    --and t.enbid = 157628
1551                 and math_distance(t.lon, t.lat, h.lon, h.lat) > 2000)
1552         --20210707 经沟通，该规则弃用，因为不少站虽然经纬度是一样的，但是不同楼层算是
      不同的物理站
1553         /*select '[地市-基站ID-经纬度]-归属多个物理站' ERROR_CLASS,
1554                SDATE,
1555                CITY_CODE,
1556                CITY_NAME,
```

```
1557                DISTRICT_CODE,
1558                DISTRICT_NAME,
1559                NETWORK_NAME,
1560                PHYSTATION_ADDRESS,
1561                BBU_NAME,
1562                ENBID,
1563                CELL_NAME,
1564                CELL_ID,
1565                CGI,
1566                LON,
1567                LAT,
1568                DIRECTION,
1569                HEIGHT,
1570                M_DOWNTILT,
1571                E_DOWNTILT,
1572                STATION_TYPE,
1573                ISDIGITALINDOOR,
1574                DOWN_FREQ,
1575                VENDER,
1576                OWN_SCHOOLYARD,
1577                TOWERADDRESS_CODE,
1578                PROPERTY,
1579                SCENE,
1580                IS_SCENESITE,
1581                MARKETING_NETWORK,
1582                TERMINALAMOUNT_5G,
1583                SECTOR_INCOMING,
1584                IS_BUSY,
1585                IS_ALIVE,
1586                IS_ALIVE_UPDATE_TIME,
1587                CONSTRUCTION,
1588                IS_SAME_ADDRESS,
1589                SAME_ADDRESS_SITES,
1590                IS_SAME_ADDRESS_CT
1591          from seq_4g_siteinfo t,
1592          --20210701 更新算法，采用地市+物理站+enbid+lonlat共同判断是否归属多个物理站，修复小区拆分无法识别的问题。
1593                (select city_name city_namek, enbid enbidk, lon lonk, lat latk
1594                        from (SELECT distinct city_name, PHYSTATION_ADDRESS, enbid, lon, lat
1595                                FROM seq_4g_siteinfo
1596                               where SDATE = TO_DATE(i_sdate, 'YYYYMMDD'))
1597                      group by city_name, enbid, lon, lat
1598                     having count(*) > 1) h
1599          where t.enbid = h.enbidk
1600            and t.city_name=h.city_namek
1601            and t.lon = h.lonk
1602            and t.lat = h.latk
1603            AND T.SDATE = TO_DATE(i_sdate, 'YYYYMMDD')
1604
1605        union all*/
1606
1607        select '[基站ID-经纬度]-在多个行政区重复' ERROR_CLASS,
1608               SDATE,
1609               CITY_CODE,
1610               CITY_NAME,
1611               DISTRICT_CODE,
```

```
1612                DISTRICT_NAME,
1613                NETWORK_NAME,
1614                PHYSTATION_ADDRESS,
1615                BBU_NAME,
1616                ENBID,
1617                CELL_NAME,
1618                CELL_ID,
1619                CGI,
1620                LON,
1621                LAT,
1622                DIRECTION,
1623                HEIGHT,
1624                M_DOWNTILT,
1625                E_DOWNTILT,
1626                STATION_TYPE,
1627                ISDIGITALINDOOR,
1628                DOWN_FREQ,
1629                VENDER,
1630                OWN_SCHOOLYARD,
1631                TOWERADDRESS_CODE,
1632                PROPERTY,
1633                SCENE,
1634                IS_SCENESITE,
1635                MARKETING_NETWORK,
1636                TERMINALAMOUNT_5G,
1637                SECTOR_INCOMING,
1638                IS_BUSY,
1639                IS_ALIVE,
1640                IS_ALIVE_UPDATE_TIME,
1641                CONSTRUCTION,
1642                IS_SAME_ADDRESS,
1643                SAME_ADDRESS_SITES,
1644                IS_SAME_ADDRESS_CT
1645          from seq_4g_siteinfo t,
1646          --20210701 更新算法，加入经纬度判断逻辑，相同的经纬度只能属于一个行政区，修复拉远基站无法识别的问题。
1647                (select CITY_NAME CITY_NAMEh, enbid enbidh, lon lonh, lat lath
1648                        from (SELECT distinct CITY_NAME, DISTRICT_NAME, enbid, lon, lat
1649                                FROM seq_4g_siteinfo
1650                               where SDATE = TO_DATE(istart, 'YYYYMMDD')
1651                               and CONSTRUCTION like '%联通%')
1652                      group by CITY_NAME, enbid, lon, lat
1653                     having count(*) > 1) h
1654          where t.CITY_NAME = h.CITY_NAMEh
1655            and t.enbid = h.enbidh
1656            and t.lon = h.lonh
1657            and t.lat = h.lath
1658            and t.CONSTRUCTION like '%联通%'
1659            and t.SDATE = TO_DATE(istart, 'YYYYMMDD')
1660            and t.is_alive=1
1661
1662         UNION all
1663
1664         select '[基站ID]-在多个地市重复' ERROR_CLASS,
1665                SDATE,
1666                CITY_CODE,
```

```
1667                CITY_NAME,
1668                DISTRICT_CODE,
1669                DISTRICT_NAME,
1670                NETWORK_NAME,
1671                PHYSTATION_ADDRESS,
1672                BBU_NAME,
1673                ENBID,
1674                CELL_NAME,
1675                CELL_ID,
1676                CGI,
1677                LON,
1678                LAT,
1679                DIRECTION,
1680                HEIGHT,
1681                M_DOWNTILT,
1682                E_DOWNTILT,
1683                STATION_TYPE,
1684                ISDIGITALINDOOR,
1685                DOWN_FREQ,
1686                VENDER,
1687                OWN_SCHOOLYARD,
1688                TOWERADDRESS_CODE,
1689                PROPERTY,
1690                SCENE,
1691                IS_SCENESITE,
1692                MARKETING_NETWORK,
1693                TERMINALAMOUNT_5G,
1694                SECTOR_INCOMING,
1695                IS_BUSY,
1696                IS_ALIVE,
1697                IS_ALIVE_UPDATE_TIME,
1698                CONSTRUCTION,
1699                IS_SAME_ADDRESS,
1700                SAME_ADDRESS_SITES,
1701                IS_SAME_ADDRESS_CT
1702          from seq_4g_siteinfo t,
1703               (select enbid enbidk
1704                  from (SELECT distinct CITY_NAME, enbid
1705                          FROM seq_4g_siteinfo
1706                         where SDATE = TO_DATE(istart, 'YYYYMMDD')
1707                         and CONSTRUCTION like '%联通%')
1708                 group by enbid
1709                having count(*) > 1) h
1710         where t.enbid = h.enbidk
1711           AND T.SDATE = TO_DATE(istart, 'YYYYMMDD')
1712           and t.CONSTRUCTION like '%联通%'
1713           and t.is_alive=1
1714
1715        union all
1716
1717        select '[基站ID-扇区ID-经纬度-方位角-小区名]-在现网重复' ERROR_CLASS,
1718               SDATE,
1719               CITY_CODE,
1720               CITY_NAME,
1721               DISTRICT_CODE,
1722               DISTRICT_NAME,
1723               NETWORK_NAME,
1724               PHYSTATION_ADDRESS,
```

```
                BBU_NAME,
                ENBID,
                CELL_NAME,
                CELL_ID,
                CGI,
                LON,
                LAT,
                DIRECTION,
                HEIGHT,
                M_DOWNTILT,
                E_DOWNTILT,
                STATION_TYPE,
                ISDIGITALINDOOR,
                DOWN_FREQ,
                VENDER,
                OWN_SCHOOLYARD,
                TOWERADDRESS_CODE,
                PROPERTY,
                SCENE,
                IS_SCENESITE,
                MARKETING_NETWORK,
                TERMINALAMOUNT_5G,
                SECTOR_INCOMING,
                IS_BUSY,
                IS_ALIVE,
                IS_ALIVE_UPDATE_TIME,
                CONSTRUCTION,
                IS_SAME_ADDRESS,
                SAME_ADDRESS_SITES,
                IS_SAME_ADDRESS_CT
           from seq_4g_siteinfo t,
           --20210701 算法更新，对于算法判重，采用enbid-ci-lonlat-direction合并判重，解决小区分裂或合并导致无法识别的问题。
                (SELECT enbid     enbidh,
                        cell_id   cell_idh,
                        CELL_NAME CELL_NAMEh,
                        lon       lonh,
                        lat       lath,
                        direction directionh
                 FROM seq_4g_siteinfo
                where SDATE = TO_DATE(istart, 'YYYYMMDD')
                and CONSTRUCTION like '%联通%'
                group by enbid, cell_id, lon, lat, direction, CELL_NAME
               having count(*) > 1) h
          where t.enbid = h.enbidh
            AND T.cell_id = H.cell_idh
            AND T.CELL_NAME = H.CELL_NAMEh
            AND T.lon = H.lonh
            AND T.lat = H.lath
            AND T.direction = H.directionh
            AND T.SDATE = TO_DATE(istart, 'YYYYMMDD')
            and t.CONSTRUCTION like '%联通%'
            and t.is_alive=1

         UNION all

        --暂时过滤掉室分,如校园站能误判
        select distinct '物理站相同但宏站小区距离大于2000米' ERROR_CLASS,
```

```
                       SDATE,
                       CITY_CODE,
                       CITY_NAME,
                       DISTRICT_CODE,
                       DISTRICT_NAME,
                       NETWORK_NAME,
                       PHYSTATION_ADDRESS,
                       BBU_NAME,
                       ENBID,
                       CELL_NAME,
                       CELL_ID,
                       CGI,
                       LON,
                       LAT,
                       DIRECTION,
                       HEIGHT,
                       M_DOWNTILT,
                       E_DOWNTILT,
                       STATION_TYPE,
                       ISDIGITALINDOOR,
                       DOWN_FREQ,
                       VENDER,
                       OWN_SCHOOLYARD,
                       TOWERADDRESS_CODE,
                       PROPERTY,
                       SCENE,
                       IS_SCENESITE,
                       MARKETING_NETWORK,
                       TERMINALAMOUNT_5G,
                       SECTOR_INCOMING,
                       IS_BUSY,
                       IS_ALIVE,
                       IS_ALIVE_UPDATE_TIME,
                       CONSTRUCTION,
                       IS_SAME_ADDRESS,
                       SAME_ADDRESS_SITES,
                       IS_SAME_ADDRESS_CT
         from seq_4g_siteinfo s,
         --20210701 更新算法，使用city+physite+enbid作为一个基站标识去计算，如果存在多个小区经纬度距离大于2000，则只显示距离大于2000的小区对
              (SELECT CITY_NAMEH, PHYSTATION_ADDRESSH, ENBIDH, LONT, LATT
                 FROM pysite_bg_2000_4g
               union
               SELECT CITY_NAMEH, PHYSTATION_ADDRESSH, ENBIDH, LONH, LATH
                 FROM pysite_bg_2000_4g) k
        where s.city_name = k.city_nameh
          and s.PHYSTATION_ADDRESS = k.PHYSTATION_ADDRESSh
          and s.enbid = k.enbidh
          and s.CONSTRUCTION like '%联通%'
          and s.SDATE = TO_DATE(istart, 'YYYYMMDD')
          and s.is_alive=1

        union all

        SELECT '工参关键字段为空' ERROR_CLASS,
               SDATE,
               CITY_CODE,
               CITY_NAME,
```

```
1839                DISTRICT_CODE,
1840                DISTRICT_NAME,
1841                NETWORK_NAME,
1842                PHYSTATION_ADDRESS,
1843                BBU_NAME,
1844                ENBID,
1845                CELL_NAME,
1846                CELL_ID,
1847                CGI,
1848                LON,
1849                LAT,
1850                DIRECTION,
1851                HEIGHT,
1852                M_DOWNTILT,
1853                E_DOWNTILT,
1854                STATION_TYPE,
1855                ISDIGITALINDOOR,
1856                DOWN_FREQ,
1857                VENDER,
1858                OWN_SCHOOLYARD,
1859                TOWERADDRESS_CODE,
1860                PROPERTY,
1861                SCENE,
1862                IS_SCENESITE,
1863                MARKETING_NETWORK,
1864                TERMINALAMOUNT_5G,
1865                SECTOR_INCOMING,
1866                IS_BUSY,
1867                IS_ALIVE,
1868                IS_ALIVE_UPDATE_TIME,
1869                CONSTRUCTION,
1870                IS_SAME_ADDRESS,
1871                SAME_ADDRESS_SITES,
1872                IS_SAME_ADDRESS_CT
1873           FROM SEQ_4G_SITEINFO m
1874          where sdate = to_date(istart, 'yyyymmdd')
1875            and (CITY_CODE is null or CITY_NAME is null or
1876                DISTRICT_CODE is null or DISTRICT_NAME is null or
1877                NETWORK_NAME is null or PHYSTATION_ADDRESS is null or
1878                BBU_NAME is null or ENBID is null or CELL_NAME is null or
1879                CELL_ID is null
1880                --or CGI is null   20210419 建议去除
1881                or LON is null or LAT is null or DIRECTION is null or
1882                HEIGHT is null or M_DOWNTILT is null or E_DOWNTILT is null
    or
1883                STATION_TYPE is null or DOWN_FREQ is null or VENDER is
    null)
1884                and m.CONSTRUCTION like '%联通%'
1885                and m.is_alive=1;
1886
1887      COMMIT;
1888      dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
    end');
1889
1890    end;
1891
1892    procedure STATS_5G_ERROR_SITEINFO(i_sdate varchar2 default null) is
1893      istart varchar2(8);
```

```
    begin
      --delete from STATS_4G_ERROR_SITEINFO where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');

      istart := '20210720';
      --默认值就是对最新的工参进行计算
      if i_sdate is null then
        SELECT to_char(max(sdate), 'yyyymmdd')
          into istart
          FROM seq_5g_siteinfo;
      else
        istart := i_sdate;
      end if;

      dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
istart=' || istart);
      dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
start');

      -- 5g基站信息为空统计
      --delete from stats_5G_SITEINFO_ISNULL where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
      execute immediate 'truncate table STATS_5G_ERROR_SITEINFO';
      insert into STATS_5G_ERROR_SITEINFO
        (ERROR_CLASS,
         SDATE,
         CITY_CODE,
         CITY_NAME,
         DISTRICT_CODE,
         DISTRICT_NAME,
         NETWORK_NAME,
         PHYSTATION_ADDRESS,
         STATION_NAME,
         GNBID,
         CELL_NAME,
         CELL_ID,
         LON,
         LAT,
         DIRECTION,
         HEIGHT,
         M_DOWNTILT,
         E_DOWNTILT,
         STATION_TYPE,
         ISDIGITALINDOOR,
         DOWN_FREQ,
         VENDER,
         OWN_SCHOOLYARD,
         TOWERADDRESS_CODE,
         PROPERTY,
         SCENE,
         IS_SCENESITE,
         MARKETING_NETWORK,
         TERMINALAMOUNT_5G,
         SECTOR_INCOMING,
         IS_BUSY,
         IS_ALIVE,
         IS_ALIVE_UPDATE_TIME,
         CONSTRUCTION,
```

```
1948          IS_SAME_ADDRESS,
1949          SAME_ADDRESS_SITES,
1950          IS_SAME_ADDRESS_CT)
1951
1952        with pysite_bg_2000_5g as
1953         (select distinct t.city_name          city_nameh,
1954                          t.PHYSTATION_ADDRESS PHYSTATION_ADDRESSh,
1955                          t.gnbid              gnbidh,
1956                          t.lon                lont,
1957                          t.lat                latt,
1958                          h.lon                lonh,
1959                          h.lat                lath
1960            from (SELECT distinct city_name,
1961                                  PHYSTATION_ADDRESS,
1962                                  gnbid,
1963                                  lon,
1964                                  lat
1965                    FROM seq_5g_siteinfo
1966                   where SDATE = TO_DATE(istart, 'YYYYMMDD')
1967                     and STATION_TYPE = '宏站'
1968                     and CONSTRUCTION like '%联通%'
1969                     and is_alive = 1) t,
1970                 (SELECT distinct city_name,
1971                                  PHYSTATION_ADDRESS,
1972                                  gnbid,
1973                                  lon,
1974                                  lat
1975                    FROM seq_5g_siteinfo
1976                   where SDATE = TO_DATE(istart, 'YYYYMMDD')
1977                     and STATION_TYPE = '宏站'
1978                     and CONSTRUCTION like '%联通%'
1979                     and is_alive = 1) h
1980           where t.PHYSTATION_ADDRESS = h.PHYSTATION_ADDRESS
1981             and t.CITY_NAME = h.CITY_NAME
1982             and t.gnbid = h.gnbid
1983             and math_distance(t.lon, t.lat, h.lon, h.lat) > 2000)
1984
1985        select '[基站ID-扇区ID-经纬度-方位角]-归属多个物理站' ERROR_CLASS,
1986               SDATE,
1987               CITY_CODE,
1988               CITY_NAME,
1989               DISTRICT_CODE,
1990               DISTRICT_NAME,
1991               NETWORK_NAME,
1992               PHYSTATION_ADDRESS,
1993               STATION_NAME,
1994               GNBID,
1995               CELL_NAME,
1996               CELL_ID,
1997               LON,
1998               LAT,
1999               DIRECTION,
2000               HEIGHT,
2001               M_DOWNTILT,
2002               E_DOWNTILT,
2003               STATION_TYPE,
2004               ISDIGITALINDOOR,
2005               DOWN_FREQ,
```

```
               VENDER,
               OWN_SCHOOLYARD,
               TOWERADDRESS_CODE,
               PROPERTY,
               SCENE,
               IS_SCENESITE,
               MARKETING_NETWORK,
               TERMINALAMOUNT_5G,
               SECTOR_INCOMING,
               IS_BUSY,
               IS_ALIVE,
               IS_ALIVE_UPDATE_TIME,
               CONSTRUCTION,
               IS_SAME_ADDRESS,
               SAME_ADDRESS_SITES,
               IS_SAME_ADDRESS_CT
          from seq_5g_siteinfo t,
               -- 20210707 算法更新，加入基站id-小区id-经纬度-方位角 一起检测，用于支持拉远或合并基站
               (select GNBID     GNBIDh,
                       CELL_ID   CELL_IDh,
                       LON       LONh,
                       LAT       LATh,
                       DIRECTION DIRECTIONh
                  from (SELECT distinct PHYSTATION_ADDRESS,
                                        GNBID,
                                        CELL_ID,
                                        LON,
                                        LAT,
                                        DIRECTION
                          FROM seq_5g_siteinfo
                         where SDATE = TO_DATE(istart, 'YYYYMMDD')
                           and CONSTRUCTION like '%联通%')
                 group by GNBID, CELL_ID, LON, LAT, DIRECTION
                having count(*) > 1) h
         where t.gnbid = h.GNBIDh
           and t.cell_id = h.CELL_IDh
           and t.LON = h.LONh
           and t.LAT = h.LATh
           and t.DIRECTION = h.DIRECTIONh
           AND T.SDATE = TO_DATE(istart, 'YYYYMMDD')
           and T.CONSTRUCTION like '%联通%'
           and t.is_alive = 1

        union all

        select '[基站ID-物理站名]-在多个地市或行政区重复' ERROR_CLASS,
               SDATE,
               CITY_CODE,
               CITY_NAME,
               DISTRICT_CODE,
               DISTRICT_NAME,
               NETWORK_NAME,
               PHYSTATION_ADDRESS,
               STATION_NAME,
               GNBID,
               CELL_NAME,
               CELL_ID,
```

```
2063                LON,
2064                LAT,
2065                DIRECTION,
2066                HEIGHT,
2067                M_DOWNTILT,
2068                E_DOWNTILT,
2069                STATION_TYPE,
2070                ISDIGITALINDOOR,
2071                DOWN_FREQ,
2072                VENDER,
2073                OWN_SCHOOLYARD,
2074                TOWERADDRESS_CODE,
2075                PROPERTY,
2076                SCENE,
2077                IS_SCENESITE,
2078                MARKETING_NETWORK,
2079                TERMINALAMOUNT_5G,
2080                SECTOR_INCOMING,
2081                IS_BUSY,
2082                IS_ALIVE,
2083                IS_ALIVE_UPDATE_TIME,
2084                CONSTRUCTION,
2085                IS_SAME_ADDRESS,
2086                SAME_ADDRESS_SITES,
2087                IS_SAME_ADDRESS_CT
2088           from seq_5g_siteinfo t,
2089                -- 20210701 算法更新，加入物理站识别
2090                (select gnbid gnbidh, PHYSTATION_ADDRESS
PHYSTATION_ADDRESSh
2091                   from (SELECT distinct CITY_NAME,
2092                                         DISTRICT_NAME,
2093                                         PHYSTATION_ADDRESS,
2094                                         gnbid
2095                           FROM seq_5g_siteinfo
2096                          where SDATE = TO_DATE(istart, 'YYYYMMDD')
2097                            and CONSTRUCTION like '%联通%')
2098                  group by gnbid, PHYSTATION_ADDRESS
2099                 having count(*) > 1) h
2100          where t.gnbid = h.gnbidh
2101            AND T.SDATE = TO_DATE(istart, 'YYYYMMDD')
2102            AND T.PHYSTATION_ADDRESS = PHYSTATION_ADDRESSh
2103            and T.CONSTRUCTION like '%联通%'
2104            and t.is_alive = 1
2105
2106         /*UNION all
2107         --不同楼层的物理站，经客户建议，排除该规则-20210722
2108
2109         select '经纬度相同但物理站不同' ERROR_CLASS,
2110                SDATE,
2111                CITY_CODE,
2112                CITY_NAME,
2113                DISTRICT_CODE,
2114                DISTRICT_NAME,
2115                NETWORK_NAME,
2116                PHYSTATION_ADDRESS,
2117                STATION_NAME,
2118                GNBID,
2119                CELL_NAME,
```

```
2120                CELL_ID,
2121                LON,
2122                LAT,
2123                DIRECTION,
2124                HEIGHT,
2125                M_DOWNTILT,
2126                E_DOWNTILT,
2127                STATION_TYPE,
2128                ISDIGITALINDOOR,
2129                DOWN_FREQ,
2130                VENDER,
2131                OWN_SCHOOLYARD,
2132                TOWERADDRESS_CODE,
2133                PROPERTY,
2134                SCENE,
2135                IS_SCENESITE,
2136                MARKETING_NETWORK,
2137                TERMINALAMOUNT_5G,
2138                SECTOR_INCOMING,
2139                IS_BUSY,
2140                IS_ALIVE,
2141                IS_ALIVE_UPDATE_TIME,
2142                CONSTRUCTION,
2143                IS_SAME_ADDRESS,
2144                SAME_ADDRESS_SITES,
2145                IS_SAME_ADDRESS_CT
2146          from seq_5g_siteinfo t,
2147               (select LON LONh, LAT LATh
2148                  from (SELECT distinct LON, LAT, PHYSTATION_ADDRESS
2149                          FROM seq_5g_siteinfo
2150                         where SDATE = TO_DATE(istart, 'YYYYMMDD')
2151                           and CONSTRUCTION like '%联通%')
2152                 group by LON, LAT
2153                having count(*) > 1) h
2154         where t.LON = h.LONh
2155           AND T.LAT = H.lath
2156           AND T.SDATE = TO_DATE(istart, 'YYYYMMDD')
2157           and T.CONSTRUCTION like '%联通%'
2158           and t.is_alive = 1*/
2159
2160        UNION all
2161
2162        select '[基站ID-扇区ID-经纬度-方位角]-在现网重复' ERROR_CLASS,
2163               SDATE,
2164               CITY_CODE,
2165               CITY_NAME,
2166               DISTRICT_CODE,
2167               DISTRICT_NAME,
2168               NETWORK_NAME,
2169               PHYSTATION_ADDRESS,
2170               STATION_NAME,
2171               GNBID,
2172               CELL_NAME,
2173               CELL_ID,
2174               LON,
2175               LAT,
2176               DIRECTION,
2177               HEIGHT,
```

```
2178                M_DOWNTILT,
2179                E_DOWNTILT,
2180                STATION_TYPE,
2181                ISDIGITALINDOOR,
2182                DOWN_FREQ,
2183                VENDER,
2184                OWN_SCHOOLYARD,
2185                TOWERADDRESS_CODE,
2186                PROPERTY,
2187                SCENE,
2188                IS_SCENESITE,
2189                MARKETING_NETWORK,
2190                TERMINALAMOUNT_5G,
2191                SECTOR_INCOMING,
2192                IS_BUSY,
2193                IS_ALIVE,
2194                IS_ALIVE_UPDATE_TIME,
2195                CONSTRUCTION,
2196                IS_SAME_ADDRESS,
2197                SAME_ADDRESS_SITES,
2198                IS_SAME_ADDRESS_CT
2199           from seq_5g_siteinfo t,
2200                (SELECT gnbid     gnbidh,
2201                        cell_id   cell_idh,
2202                        Lon       lonh,
2203                        Lat       lath,
2204                        DIRECTION DIRECTIONh
2205                   FROM seq_5g_siteinfo
2206                  where SDATE = TO_DATE(istart, 'YYYYMMDD')
2207                    and CONSTRUCTION like '%联通%'
2208                  group by gnbid, cell_id, lon, lat, DIRECTION
2209                 having count(*) > 1) h
2210          where t.gnbid = h.gnbidh
2211            AND T.cell_id = H.cell_idh
2212            AND T.Lon = H.Lonh
2213            AND T.Lat = H.Lath
2214            AND T.DIRECTION = H.DIRECTIONh
2215            AND T.SDATE = TO_DATE(istart, 'YYYYMMDD')
2216            AND T.CONSTRUCTION like '%联通%'
2217            and t.is_alive = 1
2218
2219         UNION all
2220
2221         select distinct '物理站相同但宏站小区距离大于2000米' ERROR_CLASS,
2222                         SDATE,
2223                         CITY_CODE,
2224                         CITY_NAME,
2225                         DISTRICT_CODE,
2226                         DISTRICT_NAME,
2227                         NETWORK_NAME,
2228                         PHYSTATION_ADDRESS,
2229                         STATION_NAME,
2230                         GNBID,
2231                         CELL_NAME,
2232                         CELL_ID,
2233                         LON,
2234                         LAT,
2235                         DIRECTION,
```

```
                        HEIGHT,
                        M_DOWNTILT,
                        E_DOWNTILT,
                        STATION_TYPE,
                        ISDIGITALINDOOR,
                        DOWN_FREQ,
                        VENDER,
                        OWN_SCHOOLYARD,
                        TOWERADDRESS_CODE,
                        PROPERTY,
                        SCENE,
                        IS_SCENESITE,
                        MARKETING_NETWORK,
                        TERMINALAMOUNT_5G,
                        SECTOR_INCOMING,
                        IS_BUSY,
                        IS_ALIVE,
                        IS_ALIVE_UPDATE_TIME,
                        CONSTRUCTION,
                        IS_SAME_ADDRESS,
                        SAME_ADDRESS_SITES,
                        IS_SAME_ADDRESS_CT
          from seq_5g_siteinfo s,
               --20210701 更新算法，采用city+physite+gnbid作为物理站标识，只输出两两距离>2000的小区对
               (SELECT city_nameh, PHYSTATION_ADDRESSh, gnbidh, lont, latt
                  FROM pysite_bg_2000_5g
                union
                SELECT city_nameh, PHYSTATION_ADDRESSh, gnbidh, lonh, lath
                  FROM pysite_bg_2000_5g) k
         where s.city_name = k.city_nameh
           and s.PHYSTATION_ADDRESS = k.PHYSTATION_ADDRESSh
           and s.gnbid = k.gnbidh
           and S.CONSTRUCTION like '%联通%'
           and s.sdate=to_date(istart,'yyyymmdd')
           and s.is_alive = 1

        union all

        SELECT '工参关键字段为空' ERROR_CLASS,
               SDATE,
               CITY_CODE,
               CITY_NAME,
               DISTRICT_CODE,
               DISTRICT_NAME,
               NETWORK_NAME,
               PHYSTATION_ADDRESS,
               STATION_NAME,
               GNBID,
               CELL_NAME,
               CELL_ID,
               LON,
               LAT,
               DIRECTION,
               HEIGHT,
               M_DOWNTILT,
               E_DOWNTILT,
               STATION_TYPE,
```

```
2293                ISDIGITALINDOOR,
2294                DOWN_FREQ,
2295                VENDER,
2296                OWN_SCHOOLYARD,
2297                TOWERADDRESS_CODE,
2298                PROPERTY,
2299                SCENE,
2300                IS_SCENESITE,
2301                MARKETING_NETWORK,
2302                TERMINALAMOUNT_5G,
2303                SECTOR_INCOMING,
2304                IS_BUSY,
2305                IS_ALIVE,
2306                IS_ALIVE_UPDATE_TIME,
2307                CONSTRUCTION,
2308                IS_SAME_ADDRESS,
2309                SAME_ADDRESS_SITES,
2310                IS_SAME_ADDRESS_CT
2311           FROM SEQ_5G_SITEINFO m
2312          where sdate = to_date(i_sdate, 'yyyymmdd')
2313            and (CITY_CODE is null or CITY_NAME is null or
2314                DISTRICT_CODE is null or DISTRICT_NAME is null or
2315                NETWORK_NAME is null or PHYSTATION_ADDRESS is null or
2316                STATION_NAME is null or GNBID is null or CELL_NAME is null
    or
2317                CELL_ID is null or LON is null or LAT is null or
2318                DIRECTION is null or HEIGHT is null or M_DOWNTILT is null
    or
2319                E_DOWNTILT is null or STATION_TYPE is null or
2320                DOWN_FREQ is null or VENDER is null)
2321            and CONSTRUCTION like '%联通%'
2322            and m.is_alive = 1;
2323
2324       commit;
2325       dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
    end');
2326
2327     end;
2328
2329  procedure STATS_4G_COLLEGES_SITEINFO(i_sdate varchar2,outdoor_range
    number,indoor_range number) is
2330
2331      begin
2332
2333
2334        dropindex_ifexists('I_SEQ_4G_SITEINFO_GEO');
2335        execute immediate 'truncate table seq_4g_siteinfo_GEO';
2336        --delete from seq_4g_siteinfo_GEO where trunc(sdate) =
    to_date(i_sdate,'yyyymmdd');
2337     insert into SEQ_4G_SITEINFO_GEO
2338       SELECT SDATE,
2339              CITY_CODE,
2340              CITY_NAME,
2341              DISTRICT_CODE,
2342              DISTRICT_NAME,
2343              NETWORK_NAME,
2344              PHYSTATION_ADDRESS,
2345              BBU_NAME,
```

```
2346              ENBID,
2347              CELL_NAME,
2348              CELL_ID,
2349              CGI,
2350              LON,
2351              LAT,
2352              DIRECTION,
2353              HEIGHT,
2354              M_DOWNTILT,
2355              E_DOWNTILT,
2356              STATION_TYPE,
2357              ISDIGITALINDOOR,
2358              DOWN_FREQ,
2359              VENDER,
2360              OWN_SCHOOLYARD,
2361              TOWERADDRESS_CODE,
2362              PROPERTY,
2363              SCENE,
2364              IS_SCENESITE,
2365              MARKETING_NETWORK,
2366              TERMINALAMOUNT_5G,
2367              SECTOR_INCOMING,
2368              IS_BUSY,
2369              IS_ALIVE,
2370              IS_ALIVE_UPDATE_TIME,
2371              SDO_GEOMETRY(2001,
2372                           4326,
2373                           SDO_POINT_TYPE(LON, LAT, NULL),
2374                           NULL,
2375                           null) GEOM,
2376              CONSTRUCTION,
2377              IS_SAME_ADDRESS,
2378              SAME_ADDRESS_SITES,
2379              IS_SAME_ADDRESS_CT
2380         FROM seq_4g_siteinfo T
2381        WHERE SDATE = TO_DATE(i_sdate, 'YYYYMMDD');
2382        commit;
2383
2384        --call proc_add_sdo_metadata_4326('SEQ_4G_SITEINFO_GEO','GEOM');
2385
2386        execute immediate 'create index i_SEQ_4G_SITEINFO_GEO on
SEQ_4G_SITEINFO_GEO(GEOM) indextype is MDSYS.SPATIAL_INDEX
parameters(''layer_gtype=point'')';
2387
2388        execute immediate 'truncate table STATS_4G_COLLEGES_SITEINFO';
2389        insert into STATS_4G_COLLEGES_SITEINFO
2390          SELECT B.COLLEGE_NAME,
2391                 SDATE,
2392                 CITY_CODE,
2393                 a.CITY_NAME,
2394                 DISTRICT_CODE,
2395                 DISTRICT_NAME,
2396                 NETWORK_NAME,
2397                 PHYSTATION_ADDRESS,
2398                 BBU_NAME,
2399                 ENBID,
2400                 CELL_NAME,
2401                 CELL_ID,
```

```
2402                CGI,
2403                LON,
2404                LAT,
2405                DIRECTION,
2406                HEIGHT,
2407                M_DOWNTILT,
2408                E_DOWNTILT,
2409                STATION_TYPE,
2410                ISDIGITALINDOOR,
2411                DOWN_FREQ,
2412                VENDER,
2413                OWN_SCHOOLYARD,
2414                TOWERADDRESS_CODE,
2415                PROPERTY,
2416                SCENE,
2417                IS_SCENESITE,
2418                MARKETING_NETWORK,
2419                TERMINALAMOUNT_5G,
2420                SECTOR_INCOMING,
2421                IS_BUSY,
2422                IS_ALIVE,
2423                IS_ALIVE_UPDATE_TIME,
2424                CONSTRUCTION,
2425               IS_SAME_ADDRESS,
2426               SAME_ADDRESS_SITES,
2427               IS_SAME_ADDRESS_CT
2428          FROM SEQ_4G_SITEINFO_GEO A, SEQ_CFG_COLLEGES_GEO B
2429         WHERE SDO_WITHIN_DISTANCE(A.GEOM,
2430                                   B.GEOM,
2431                                   'distance=' || outdoor_range ||
2432                                   ' unit=m') = 'TRUE'
2433           AND A.CITY_NAME = B.CITY_NAME
2434           and STATION_TYPE = '宏站'
2435        union
2436        SELECT B.COLLEGE_NAME,
2437               SDATE,
2438               CITY_CODE,
2439               a.CITY_NAME,
2440               DISTRICT_CODE,
2441               DISTRICT_NAME,
2442               NETWORK_NAME,
2443               PHYSTATION_ADDRESS,
2444               BBU_NAME,
2445               ENBID,
2446               CELL_NAME,
2447               CELL_ID,
2448               CGI,
2449               LON,
2450               LAT,
2451               DIRECTION,
2452               HEIGHT,
2453               M_DOWNTILT,
2454               E_DOWNTILT,
2455               STATION_TYPE,
2456               ISDIGITALINDOOR,
2457               DOWN_FREQ,
2458               VENDER,
2459               OWN_SCHOOLYARD,
```

```
                TOWERADDRESS_CODE,
                PROPERTY,
                SCENE,
                IS_SCENESITE,
                MARKETING_NETWORK,
                TERMINALAMOUNT_5G,
                SECTOR_INCOMING,
                IS_BUSY,
                IS_ALIVE,
                IS_ALIVE_UPDATE_TIME,
                CONSTRUCTION,
               IS_SAME_ADDRESS,
               SAME_ADDRESS_SITES,
               IS_SAME_ADDRESS_CT
          FROM SEQ_4G_SITEINFO_GEO A, SEQ_CFG_COLLEGES_GEO B
         WHERE SDO_WITHIN_DISTANCE(A.GEOM,
                                   B.GEOM,
                                   'distance=' || indoor_range || '
unit=m') =
                'TRUE'
           AND A.CITY_NAME = B.CITY_NAME
           and STATION_TYPE = '室分';
       commit;
     end;


    procedure STATS_5G_COLLEGES_SITEINFO(i_sdate       varchar2,
                                         outdoor_range number,
                                         indoor_range  number) is

    begin

      dropindex_ifexists('I_SEQ_5G_SITEINFO_GEO');
      execute immediate 'truncate table SEQ_5G_SITEINFO_GEO';
      --delete from SEQ_5G_SITEINFO_GEO where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
      insert into SEQ_5G_SITEINFO_GEO
        SELECT SDATE,
               CITY_CODE,
               CITY_NAME,
               DISTRICT_CODE,
               DISTRICT_NAME,
               NETWORK_NAME,
               PHYSTATION_ADDRESS,
               STATION_NAME,
               GNBID,
               CELL_NAME,
               CELL_ID,
               LON,
               LAT,
               DIRECTION,
               HEIGHT,
               M_DOWNTILT,
               E_DOWNTILT,
               STATION_TYPE,
               ISDIGITALINDOOR,
               DOWN_FREQ,
               VENDER,
```

```
2516                OWN_SCHOOLYARD,
2517                TOWERADDRESS_CODE,
2518                PROPERTY,
2519                SCENE,
2520                IS_SCENESITE,
2521                MARKETING_NETWORK,
2522                TERMINALAMOUNT_5G,
2523                SECTOR_INCOMING,
2524                IS_BUSY,
2525                IS_ALIVE,
2526                IS_ALIVE_UPDATE_TIME,
2527                SDO_GEOMETRY(2001,
2528                             4326,
2529                             SDO_POINT_TYPE(LON, LAT, NULL),
2530                             NULL,
2531                             NULL) GEOM,
2532                CONSTRUCTION,
2533                IS_SAME_ADDRESS,
2534                SAME_ADDRESS_SITES,
2535                IS_SAME_ADDRESS_CT
2536           FROM SEQ_5G_SITEINFO T
2537          WHERE SDATE = TO_DATE(i_sdate, 'YYYYMMDD');
2538       commit;
2539
2540       --call proc_add_sdo_metadata_4326('SEQ_5G_SITEINFO_GEO','GEOM');
2541
2542       execute immediate 'create index i_SEQ_5G_SITEINFO_GEO on
SEQ_5G_SITEINFO_GEO(GEOM) indextype is MDSYS.SPATIAL_INDEX
parameters(''layer_gtype=point'')';
2543
2544       execute immediate 'truncate table STATS_5G_COLLEGES_SITEINFO';
2545       insert into STATS_5G_COLLEGES_SITEINFO
2546         SELECT B.COLLEGE_NAME,
2547                SDATE,
2548                CITY_CODE,
2549                a.CITY_NAME,
2550                DISTRICT_CODE,
2551                DISTRICT_NAME,
2552                NETWORK_NAME,
2553                PHYSTATION_ADDRESS,
2554                STATION_NAME,
2555                GNBID,
2556                CELL_NAME,
2557                CELL_ID,
2558                LON,
2559                LAT,
2560                DIRECTION,
2561                HEIGHT,
2562                M_DOWNTILT,
2563                E_DOWNTILT,
2564                STATION_TYPE,
2565                ISDIGITALINDOOR,
2566                DOWN_FREQ,
2567                VENDER,
2568                OWN_SCHOOLYARD,
2569                TOWERADDRESS_CODE,
2570                PROPERTY,
2571                SCENE,
```

```
2572                IS_SCENESITE,
2573                MARKETING_NETWORK,
2574                TERMINALAMOUNT_5G,
2575                SECTOR_INCOMING,
2576                IS_BUSY,
2577                IS_ALIVE,
2578                IS_ALIVE_UPDATE_TIME,
2579                CONSTRUCTION,
2580                IS_SAME_ADDRESS,
2581                SAME_ADDRESS_SITES,
2582                IS_SAME_ADDRESS_CT
2583           FROM SEQ_5G_SITEINFO_GEO A, SEQ_CFG_COLLEGES_GEO B
2584          WHERE SDO_WITHIN_DISTANCE(A.GEOM,
2585                                    B.GEOM,
2586                                    'distance=' || outdoor_range || '
unit=m') =
2587                'TRUE'
2588            AND A.CITY_NAME = B.CITY_NAME
2589            and STATION_TYPE = '宏站'
2590         union
2591         SELECT B.COLLEGE_NAME,
2592                SDATE,
2593                CITY_CODE,
2594                a.CITY_NAME,
2595                DISTRICT_CODE,
2596                DISTRICT_NAME,
2597                NETWORK_NAME,
2598                PHYSTATION_ADDRESS,
2599                STATION_NAME,
2600                GNBID,
2601                CELL_NAME,
2602                CELL_ID,
2603                LON,
2604                LAT,
2605                DIRECTION,
2606                HEIGHT,
2607                M_DOWNTILT,
2608                E_DOWNTILT,
2609                STATION_TYPE,
2610                ISDIGITALINDOOR,
2611                DOWN_FREQ,
2612                VENDER,
2613                OWN_SCHOOLYARD,
2614                TOWERADDRESS_CODE,
2615                PROPERTY,
2616                SCENE,
2617                IS_SCENESITE,
2618                MARKETING_NETWORK,
2619                TERMINALAMOUNT_5G,
2620                SECTOR_INCOMING,
2621                IS_BUSY,
2622                IS_ALIVE,
2623                IS_ALIVE_UPDATE_TIME,
2624                CONSTRUCTION,
2625                IS_SAME_ADDRESS,
2626                SAME_ADDRESS_SITES,
2627                IS_SAME_ADDRESS_CT
2628           FROM SEQ_5G_SITEINFO_GEO A, SEQ_CFG_COLLEGES_GEO B
```

```
2629          WHERE SDO_WITHIN_DISTANCE(A.GEOM,
2630                                    B.GEOM,
2631                                    'distance=' || indoor_range || '
unit=m') =
2632                'TRUE'
2633            AND A.CITY_NAME = B.CITY_NAME
2634            and STATION_TYPE = '室分';
2635        commit;
2636      end;
2637
2638
2639      -- 优化基站级判断采用物理站+enbid综合判断
2640      --DB dbscan密度聚类的结果
2641      procedure STATS_4G_DEVIATE_SUMMARY_DB(i_sdate varchar2 default null)
is
2642                istart varchar2(8);
2643      begin
2644        --delete from STATS_4G_ERROR_SITEINFO where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
2645
2646        istart :='20210720';
2647        --默认值就是对最新的工参进行计算
2648        if i_sdate is null then
2649          SELECT to_char(max(sdate),'yyyymmdd') into istart FROM
SEQ_MDT_GRID50_DBSCAN;
2650        else
2651          istart :=i_sdate;
2652        end if;
2653        dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
istart=' || istart);
2654        dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
start');
2655        --create table fdd_MDT_cell_position as
2656        --计算小区的虚拟坐标
2657        --SEQ_MDT_GRID50_DBSCAN 去毛刺后的栅格
2658        --SEQ_MDT_GRID50_CELL 原始栅格数据
2659        --带avg的为算数平均，未加权
2660        execute immediate 'truncate table stats_4G_DEVIATE_SUMMARY_DB';
2661        --insert into fdd_MDT_cell_position
2662
2663        insert into stats_4G_DEVIATE_SUMMARY_DB
2664        with fdd_MDT_cell_position as (
2665        select SDATE,
2666               CITY,
2667               ENODEB,
2668               SC_ECI,
2669               sum(gridx * rsrp_samples) / sum(rsrp_samples)*0.00045 lon, --
根据采样点加权计算
2670               sum(gridy * rsrp_samples) / sum(rsrp_samples)*0.00045 lat--,
2671               --注释掉绝对平均算法20210816
2672               --avg(GRIDX * 0.00045) avg_lon,
2673               --avg(GRIDY * 0.00045) avg_lat
2674          from (SELECT b.SDATE, b.CITY, b.ENODEB, b.SC_ECI, b.GRIDX,
b.GRIDY,b.RSRP_SAMPLES
2675                  FROM SEQ_MDT_GRID50_DBSCAN a, SEQ_MDT_GRID50_CELL b
2676                 where a.enodeb = b.enodeb
2677                   and a.gridx = b.gridx
2678                   and a.gridy = b.gridy
```

```
                and a.sdate = b.sdate
                and a.sdate = to_date(istart,'yyyymmdd')) t
      group by SDATE, CITY, ENODEB, SC_ECI),
    --commit;
    --根据小区的虚拟坐标和工参，按照物理站名采用平均的方法计算物理站的虚拟坐标
    --带avg的为算数平均，未加权
    --execute immediate 'truncate table fdd_MDT_ENB_position';
    --insert into fdd_MDT_ENB_position
    fdd_MDT_ENB_position as (
      select a.sdate,
             city,
             enodeb,
             avg(a.lon) lon,
             avg(a.lat) lat,
             --注释掉绝对平均算法20210816
             --avg(a.avg_lon) avg_lon,
             --avg(a.avg_lat) avg_lat,
             PHYSTATION_ADDRESS
        from fdd_MDT_cell_position a, seq_4g_siteinfo b
       where a.enodeb = b.enbid
         and mod(a.sc_eci, 256) = b.cell_id
         and b.sdate = (select max(sdate) from seq_4g_siteinfo)
       group by a.sdate, city, enodeb, PHYSTATION_ADDRESS),
     --commit;
     --关联工参数据，计算小区级偏离距离
     --此步骤可能会出现：工参中的小区id与mr中的mod(sc_eci,256)不相等，导致无法关联。
     --create table fdd_MDT_cell_position_distance as
     --execute immediate 'truncate table fdd_MDT_cell_position_distance';
     --insert into fdd_MDT_cell_position_distance
     fdd_MDT_cell_position_distance as (
     SELECT b.lon MDT_LON,
            b.lat MDT_LAT,
            math_distance(A.LON, A.LAT, B.lon, B.lat) deviate_DISTANCE,--小区虚拟坐标与小区工参坐标的距离
            --注释掉绝对平均算法20210816
            --B.avg_lon AVG_MDT_LON,
            --B.avg_lat AVG_MDT_LAT,
            --math_distance(A.LON, A.LAT, B.avg_lon, B.avg_lat) avg_deviate_DISTANCE,
            b.CITY,
            a.CITY_CODE,
            a.CITY_NAME,
            a.DISTRICT_CODE,
            a.DISTRICT_NAME,
            a.NETWORK_NAME,
            a.PHYSTATION_ADDRESS,
            a.BBU_NAME,
            a.ENBID,
            a.CELL_NAME,
            SC_ECI,
            a.CELL_ID,
            a.CGI,
            a.LON,
            a.LAT,
            a.DIRECTION,
            a.HEIGHT,
            a.M_DOWNTILT,
```

```
2734              a.E_DOWNTILT,
2735              a.STATION_TYPE,
2736              a.DOWN_FREQ,
2737              a.VENDER,
2738              a.SCENE,
2739              a.IS_ALIVE,
2740              a.IS_ALIVE_UPDATE_TIME
2741         FROM seq_4g_siteinfo A, fdd_MDT_cell_position B, CFG_CITY C
2742        WHERE A.SDATE = (select max(sdate) from seq_4g_siteinfo)
2743          AND A.CITY_NAME = C.CITY_NAME
2744          AND C.CITY = B.CITY
2745          AND A.ENBID = B.ENODEB
2746          AND mod(b.sc_eci,256) = a.cell_id
2747          AND A.STATION_TYPE = '宏站'
2748          --AND math_distance(A.LON, A.LAT, B.lon, B.lat)>3000
2749          ),
2750          --commit;
2751        --计算基站级偏离距离
2752        --考虑了物理站名
2753        --execute immediate 'truncate table stats_4G_SITEINFO_DEVIATE';
2754        --insert into stats_4G_SITEINFO_DEVIATE
2755        stats_4G_SITEINFO_DEVIATE as (
2756          SELECT b.lon enb_lon,
2757                 b.lat enb_lat,
2758                 --math_distance(b.lon, b.lat, a.lon, a.lat)
    enb_DEVIATE_DISTANCE,
2759                 max(math_distance(a.lon, a.lat, b.lon,
    b.lat))over(partition by a.city,a.PHYSTATION_ADDRESS,a.enbid)
    enb_DEVIATE_DISTANCE,--物理站虚拟位置与小区工参位置距离
2760                 --注释掉绝对平均算法20210816
2761                 --b.avg_lon enb_avg_lon,
2762                 --b.avg_lat enb_avg_lat,
2763                 --math_distance(b.avg_lon, b.avg_lat, a.lon, a.lat)
    enb_avg_DEVIATE_DISTANCE,
2764                 a.MDT_LON,
2765                 a.MDT_LAT,
2766                 a.DEVIATE_DISTANCE,--小区虚拟坐标与小区工参坐标的距离
2767                 --注释掉绝对平均算法20210816
2768                 --a.AVG_MDT_LON,
2769                 --a.AVG_MDT_LAT,
2770                 --a.AVG_DEVIATE_DISTANCE,
2771                 a.CITY,
2772                 a.CITY_CODE,
2773                 a.CITY_NAME,
2774                 a.DISTRICT_CODE,
2775                 a.DISTRICT_NAME,
2776                 a.NETWORK_NAME,
2777                 a.PHYSTATION_ADDRESS,
2778                 a.BBU_NAME,
2779                 a.ENBID,
2780                 a.CELL_NAME,
2781                 a.SC_ECI,
2782                 a.CELL_ID,
2783                 a.CGI,
2784                 a.LON,
2785                 a.LAT,
2786                 a.DIRECTION,
2787                 a.HEIGHT,
```

```
2788                 a.M_DOWNTILT,
2789                 a.E_DOWNTILT,
2790                 a.STATION_TYPE,
2791                 a.DOWN_FREQ,
2792                 a.VENDER,
2793                 a.SCENE,
2794                 a.IS_ALIVE,
2795                 a.IS_ALIVE_UPDATE_TIME
2796            FROM fdd_MDT_cell_position_distance a, fdd_MDT_ENB_position b
2797           where a.city = b.city
2798             and a.enbid = b.enodeb
2799             and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS)
2800       --commit;
2801
2802    --create table stats_4G_DEVIATE_SUMMARY AS
2803    --实现偏离度行转列
2804    --execute immediate 'truncate table stats_4G_DEVIATE_SUMMARY';
2805     select t2.*
2806       from (select CITY_NAME,
2807                    DISTRICT_NAME,
2808                    CITY_CODE,
2809                    DISTRICT_CODE,
2810                    PHYSTATION_ADDRESS,
2811                    BBU_NAME,
2812                    ENBID,
2813                    --LON,
2814                    --LAT,
2815                    STATION_TYPE,
2816                    VENDER,
2817                    first_value(SCENE) over(partition by CITY_NAME,
DISTRICT_NAME, PHYSTATION_ADDRESS, BBU_NAME, ENBID order by
DEVIATE_DISTANCE desc) SCENE,
2818                    enb_DEVIATE_DISTANCE,--物理站虚拟位置与小区工参位置距离
2819                    SC_ECI,
2820                    CELL_NAME,
2821                    --CELL_ID,
2822                    DEVIATE_DISTANCE,--小区虚拟坐标与小区工参坐标的距离
2823                    Dense_rank() over(partition by CITY_NAME,
DISTRICT_NAME, PHYSTATION_ADDRESS, BBU_NAME, ENBID order by
DEVIATE_DISTANCE desc) rk
2824               from stats_4G_SITEINFO_DEVIATE a
2825              where not exists
2826              (SELECT distinct CITY_NAME, ENODEBID, PHYSTATION_ADDRESS
2827                       FROM SEQ_4G_SITEINFO_DEVIATE_White b
2828                      where back_info like '%远距离覆盖需求%'
2829                        and a.city_name = b.city_name
2830                        and a.enbid = enodebid
2831                        and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS)
2832                        and enb_DEVIATE_DISTANCE>1000  --20210816 过滤 物理
站虚拟位置与小区工参位置距离>1000
2833                        )
2834     pivot(max(sc_eci) ECI, max(cell_name) CELL_NAME,
max(DEVIATE_DISTANCE) DISTANCE
2835        for rk in('1', '2', '3', '4', '5', '6', '7', '8', '9')) t2;
2836     commit;
2837
2838
2839     dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') ||
```

```
2840                            ' : end');
2841
2842      end;
2843
2844
2845      -- i_sdate 为MDT栅格数据的日期
2846      procedure STATS_4G_DEVIATE_SUMMARY_QL(i_sdate varchar2 default null)
is
2847                istart varchar2(8);
2848      begin
2849        --delete from STATS_4G_ERROR_SITEINFO where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
2850
2851        istart :='20210720';
2852        --默认值就是对最新的工参进行计算
2853        if i_sdate is null then
2854          SELECT to_char(max(sdate),'yyyymmdd') into istart FROM
SEQ_MDT_GRID50_CELL;
2855        else
2856          istart :=i_sdate;
2857        end if;
2858        -- 全量加权算法
2859        execute immediate 'truncate table STATS_4G_DEVIATE_SUMMARY_QL';
2860        insert into STATS_4G_DEVIATE_SUMMARY_ql
2861        WITH CELL_LON AS
2862         (SELECT SDATE,
2863                 CITY,
2864                 ENODEB,
2865                 SC_ECI,
2866                 sum(RSRP_SAMPLES * GRIDX) / sum(RSRP_SAMPLES) * 0.00045
lon,
2867                 sum(RSRP_SAMPLES * GRIDY) / sum(RSRP_SAMPLES) * 0.00045 lAT
2868            FROM SEQ_MDT_GRID50_CELL
2869           where sdate = to_date(istart,'yyyymmdd')
2870           GROUP BY SDATE, CITY, ENODEB, SC_ECI),
2871        --SELECT * FROM CELL_LON where enodeb=166775
2872        PHYSTATION_LON AS
2873         (SELECT CITY, B.ENBID, PHYSTATION_ADDRESS, AVG(A.LON) LON,
AVG(A.LAT) LAT
2874            FROM CELL_LON A, SEQ_4G_SITEINFO B
2875           WHERE B.SDATE = (select max(sdate) from SEQ_4G_SITEINFO)
2876             AND A.ENODEB = B.ENBID
2877             AND MOD(A.SC_ECI, 256) = B.CELL_ID
2878           GROUP BY CITY, B.ENBID, PHYSTATION_ADDRESS),
2879        --SELECT * FROM PHYSTATION_LON where ENBID=166775
2880        cell_deviate AS
2881         (SELECT distinct max(math_distance(a.lon, a.lat, b.lon, b.lat))
over(partition by a.city, b.PHYSTATION_ADDRESS, b.enbid) enb_distance,
2882                         --math_distance(a.lon, a.lat, b.lon, b.lat)
enb_distance,
2883                         a.city,
2884                         b.CITY_NAME,
2885                         b.DISTRICT_NAME,
2886                         b.CITY_CODE,
2887                         b.DISTRICT_CODE,
2888                         b.PHYSTATION_ADDRESS,
2889                         b.STATION_TYPE,
2890                         b.VENDER,
```

```
                        b.scene,
                        b.BBU_NAME,
                        b.ENBID,
                        b.CELL_NAME,
                        b.CELL_ID,
                        b.LON,
                        b.LAT,
                        a.LON                enb_lon,
                        a.LAT                enb_lat
          FROM PHYSTATION_LON A, SEQ_4G_SITEINFO B
         WHERE B.SDATE = (select max(sdate) from SEQ_4G_SITEINFO)
           AND A.ENBID = B.ENBID
           AND A.PHYSTATION_ADDRESS = B.PHYSTATION_ADDRESS
           and b.station_type = '宏站'),
      --SELECT * FROM cell_deviate where enbid=166775
      cell_deviate3000 as
       (select b.*
          from (select distinct CITY_NAME, PHYSTATION_ADDRESS, ENBID
                  from cell_deviate
                 where enb_distance > 3000) a,
               cell_deviate b
         where a.city_name = b.city_name
           and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
           and a.enbid = b.enbid),
      --SELECT * FROM cell_deviate3000 where enbid=166775
      site_deviate3000 as
       (select a.*, math_distance(a.lon, a.lat, b.lon, b.lat)
cell_distance
          from cell_deviate3000 a, CELL_LON b
         where a.city = b.city
           and a.enbid = b.enodeb
           and a.cell_id = mod(b.sc_eci, 256))

      --select * from site_deviate3000 where enbid=166775
      select t2.*
        from (select CITY_NAME,
                     DISTRICT_NAME,
                     CITY_CODE,
                     DISTRICT_CODE,
                     PHYSTATION_ADDRESS,
                     BBU_NAME,
                     ENBID,
                     --LON,
                     --LAT,
                     STATION_TYPE,
                     VENDER,
                     --SCENE,
                     first_value(SCENE) over(partition by CITY_NAME,
DISTRICT_NAME, PHYSTATION_ADDRESS, BBU_NAME, ENBID order by cell_DISTANCE
desc) SCENE,
                     enb_DISTANCE,
                     ENBID * 256 + cell_id cell_id,
                     CELL_NAME,
                     --CELL_ID,
                     cell_DISTANCE,
                     Dense_rank() over(partition by CITY_NAME,
DISTRICT_NAME, PHYSTATION_ADDRESS, BBU_NAME, ENBID order by cell_DISTANCE
desc) rk
```

```
2944                 from site_deviate3000 a)
2945       pivot (max(cell_id) ECI, max(cell_name) CELL_NAME,
max(cell_DISTANCE) DISTANCE for rk in('1',
2946
                 '2',
2947
                 '3',
2948
                 '4',
2949
                 '5',
2950
                 '6',
2951
                 '7',
2952
                 '8',
2953
                 '9')) t2;
2954
2955
                 commit;
2956     end;
2957
2958     --更新有业务量无工参统计

2959     procedure STATS_HAS_FLOW_NO_4G_GC(i_sdate varchar2 default null) is
2960               istart varchar2(8);
2961               iend varchar2(8);
2962     begin
2963       --delete from STATS_4G_ERROR_SITEINFO where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
2964       if i_sdate is null then
2965         select min(sdate) into istart
2966         from (select max(sdate) sdate
2967                 from sdr_flow_4g_cell_day
2968               union
2969               select max(sdate) sdate
2970                 from sdr_flow_4g_enb_day
2971               union
2972               select to_char(sysdate, 'yyyymmdd')
2973                 from dual);
2974         select to_char(to_date(istart,'yyyymmdd')-6,'yyyymmdd') into iend
from dual;
2975       else
2976         select to_char(to_date(i_sdate,'yyyymmdd')-6,'yyyymmdd') into iend
from dual;
2977       end if;
2978
2979       dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
istart=' || istart || ',  iend=' ||iend);
2980       dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
start');
2981
2982       execute immediate 'truncate table STATS_HAS_FLOW_NO_4G_GC';
2983       insert into STATS_HAS_FLOW_NO_4G_GC
2984       SELECT iend ||'-'||istart "数据时间",
2985              ENBID || '-' || CELL_ID "基站-小区",
```

```
2986              ENBID "基站ID",
2987              CELL_ID "小区ID",
2988              a_city_name "地市名称",
2989              --b_city_name,
2990              --case when a_city_name is null then b_city_name else
    a_city_name end  "地市名称",
2991              THROUGHPUT "业务量Byte",
2992              case
2993                when THROUGHPUT > 0 then
2994                 '是'
2995                else
2996                 '否'
2997              end "是否有业务量",
2998              case
2999                when BENBID is null then
3000                 '否'
3001                else
3002                 '是'
3003              end "是否工参小区",
3004              CELL_NAME "小区名称-工参",
3005              b_city_name "地市名称-工参",
3006              CONSTRUCTION "承建方-工参",
3007              case when enbidc is null then '否' else '是' end "是否存在工参质
    量问题"
3008         FROM (SELECT a.enbid,
3009                      a.cell_id,
3010                      max(a.city_name) a_city_name,
3011                      max(b.city_name) b_city_name,
3012                      sum(a.throughput) throughput,
3013                      max(b.cell_name) cell_name,
3014                      max(b.enbid) benbid,
3015                      max(CONSTRUCTION) CONSTRUCTION,
3016                      max(c.enbid) enbidc
3017                 FROM (select enbid,
3018                              cell_id,
3019                              max(city_name) city_name,
3020                              sum(throughput) throughput
3021                         from sdr_flow_4g_cell_day k
3022                         left join sdr_flow_city_conf t
3023                           on (k.layer2id = t.city_code)
3024                        where k.sdate between iend  and istart and
    k.enbid>0
3025                        group by enbid, cell_id) a
3026                 left join (select enbid,
3027                                   cell_id,
3028                                   CONSTRUCTION,
3029                                   max(cell_name) cell_name,
3030                                   max(city_name) city_name
3031                              from seq_4g_siteinfo
3032                             where sdate = (select max(sdate) from
    seq_4g_siteinfo)
3033                             group by enbid, cell_id,CONSTRUCTION) b
3034                   on (a.enbid = b.enbid
3035                  and a.cell_id = b.cell_id)
3036                  left join (select distinct enbid,cell_id from
    STATS_4G_ERROR_SITEINFO) c
3037                  on (a.enbid = c.enbid
3038                  and a.cell_id = c.cell_id)
```

```
3039                --where a.enbid=70947
3040                 group by a.enbid, a.cell_id);
3041        commit;
3042        execute immediate 'truncate table STATS_4G_ERR_FLAG';
3043        insert into STATS_4G_ERR_FLAG
3044          with a_2d as (
3045          select distinct a.ENBID,a.cell_id
3046            from seq_4g_siteinfo a,STATS_4g_CELL_FLOW_7d_DAY b
3047           where a.sdate = (select max(sdate) from seq_4g_siteinfo)
3048             and a.is_alive = 1
3049             and b.sdate = (select max(sdate) from
STATS_4g_CELL_FLOW_7d_DAY)
3050             and a.enbid = b.enbid
3051             and a.cell_id = b.cell_id),
3052          b_err as (
3053          SELECT distinct ENBID, CELL_ID
3054            FROM STATS_4G_ERROR_SITEINFO
3055           where sdate = (select max(sdate) from STATS_4G_ERROR_SITEINFO)
3056             and is_alive = 1)
3057
3058          select '4G小区2天无业务|工参不全不准' FLAG, t.* from (select * from
a_2d INTERSECT select * from b_err) t
3059          union all
3060          select '4G小区2天无业务', t.* from (select * from a_2d MINUS select
* from b_err) t
3061          union all
3062          select '工参不全不准', t.* from (select * from b_err MINUS select *
from a_2d) t;
3063        commit;
3064        dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
end');
3065      end;
3066
3067     procedure STATS_HAS_FLOW_NO_5G_GC(i_sdate varchar2 default null) is
3068                istart varchar2(8);
3069                iend varchar2(8);
3070     begin
3071        --delete from STATS_4G_ERROR_SITEINFO where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
3072        if i_sdate is null then
3073        select to_char(min(sdate), 'yyyymmdd')
3074          into istart
3075          from (select city, max(sdate) sdate
3076                  from SDR_FLOW_CELL_THROUGHPUT
3077                 group by city
3078                union
3079                select 'system', trunc(sysdate)
3080                  from dual);
3081          select to_char(to_date(istart,'yyyymmdd')-6,'yyyymmdd') into iend
from dual;
3082        else
3083          select to_char(to_date(i_sdate,'yyyymmdd')-6,'yyyymmdd') into iend
from dual;
3084        end if;
3085
3086        dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
istart=' || istart || ',  iend=' ||iend);
```

```
3087        dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
start');
3088
3089        execute immediate 'truncate table STATS_HAS_FLOW_NO_5G_GC';
3090        insert into STATS_HAS_FLOW_NO_5G_GC
3091        select iend ||'-'||istart "数据时间",
3092               GNBID || '-' || CELL_ID "基站-小区",
3093               GNBID "基站gnbid",
3094               CELL_ID "小区id",
3095               CITY "城市",
3096               VENDER "厂家",
3097               THROUGHPUT "业务量MByte",
3098               case
3099                 when THROUGHPUT > 0 then
3100                  '是'
3101                 else
3102                  '否'
3103               end "是否有业务量",
3104               case
3105                 when bgnbid is null then
3106                  '否'
3107                 else
3108                  '是'
3109               end "是否工参小区",
3110               CELL_NAME "小区名称-工参",
3111               city_name "城市名称-工参",
3112               CONSTRUCTION "承建方-工参",
3113               case when gnbidc is null then '否' else '是' end "是否存在工参质
量问题"
3114          from (select a.GNBID,
3115                       a.CELL_ID,
3116                       max(a.CITY) CITY,
3117                       max(a.VENDER) VENDER,
3118                       sum(a.THROUGHPUT) THROUGHPUT,
3119                       max(b.gnbid) bgnbid,
3120                       max(b.cell_name) cell_name,
3121                       max(b.city_name) city_name,
3122                       max(CONSTRUCTION) CONSTRUCTION,
3123                       max(c.gnbid) gnbidc
3124                  from (SELECT GNBID,
3125                               CELL_ID,
3126                               max(CITY) CITY,
3127                               max(VENDER) VENDER,
3128                               sum(nvl(UL_THROUGHPUT_M, 0) +
nvl(DL_THROUGHPUT_M, 0)) THROUGHPUT
3129                          FROM SDR_FLOW_CELL_THROUGHPUT
3130                         where sdate between to_date(iend, 'yyyymmdd') and
3131                               to_date(istart, 'yyyymmdd')
3132                         group by GNBID, CELL_ID) a
3133                  left join (select GNBID,
3134                                    CELL_ID,
3135                                    max(CITY_name) CITY_name,
3136                                    max(cell_name) cell_name,
3137                                    max(CONSTRUCTION) CONSTRUCTION
3138                               from seq_5g_siteinfo
3139                              where sdate = (select max(sdate) from
seq_5g_siteinfo)
3140                              group by GNBID, CELL_ID) b
```

```
                 on (a.gnbid = b.gnbid and a.cell_id = b.cell_id)
                left join (SELECT distinct gnbid,cell_id FROM
STATS_5G_ERROR_SITEINFO) c
                on (a.gnbid = c.gnbid and a.cell_id = c.cell_id)
                group by a.GNBID, a.CELL_ID);
      commit;
      execute immediate 'truncate table STATS_5G_ERR_FLAG';
      insert into STATS_5G_ERR_FLAG
        with a_2d as (
        select distinct a.gnbid,a.cell_id
          from seq_5g_siteinfo a,STATS_5g_CELL_FLOW_7d_DAY b
         where a.sdate = (select max(sdate) from seq_5g_siteinfo)
           and a.is_alive = 1
           and b.sdate = (select max(sdate) from
STATS_5g_CELL_FLOW_7d_DAY)
           and a.gnbid = b.gnbid
           and a.cell_id = b.cell_id),
        b_err as (
        SELECT distinct gnbid, CELL_ID
          FROM STATS_5G_ERROR_SITEINFO
         where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
           and is_alive = 1)

        select '5G小区2天无业务|工参不全不准' FLAG, t.* from (select * from
a_2d INTERSECT select * from b_err) t
        union all
        select '5G小区2天无业务', t.* from (select * from a_2d MINUS select
* from b_err) t
        union all
        select '工参不全不准', t.* from (select * from b_err MINUS select *
from a_2d) t;
        commit;

      dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') ||' :
start');
     end;
end pkg_basedata_governance_v4;


```

8、课题联系人信息

```
# 合作方
# 1）联通客户：陈学、连坤（信息化）
# 2）大屏开发：东方国信-秦亚伟；掌沃建-安全
# 3）工单流程：国信-未丽红、周智等
```

9、资料备份

```
目前所有资料都存放于D:\gongcanzhili\02_gczl目录。
├─01业务量备份
├─02工参数据
├─03报表导出-刘子龙
├─04plsql脚本备份
├─05其它备份
```

```
|   └偏离度聚类备份
├─06基站偏离度
├─07软件程序
├─08接口上传
├─09工参资源大屏治理_备份
|   ├─业务量
|   ├─作战地图工参
|   ├─华为SEQ业务量数据及字段说明
|   ├─基站偏离统计
|   ├─外部输出
|   |   ├─4G有业务量但无工参
|   |   └─5G有业务量但无工参
|   ├─大屏原型与报告
|   ├─工作总结
|   ├─工参治理系统设计-初版
|   ├─接口规范
|   ├─网络需求工单
|   └─表结构说明
└─10接口程序源码
    ├─oracle_2_esb
    └─oracle_2_excel
```