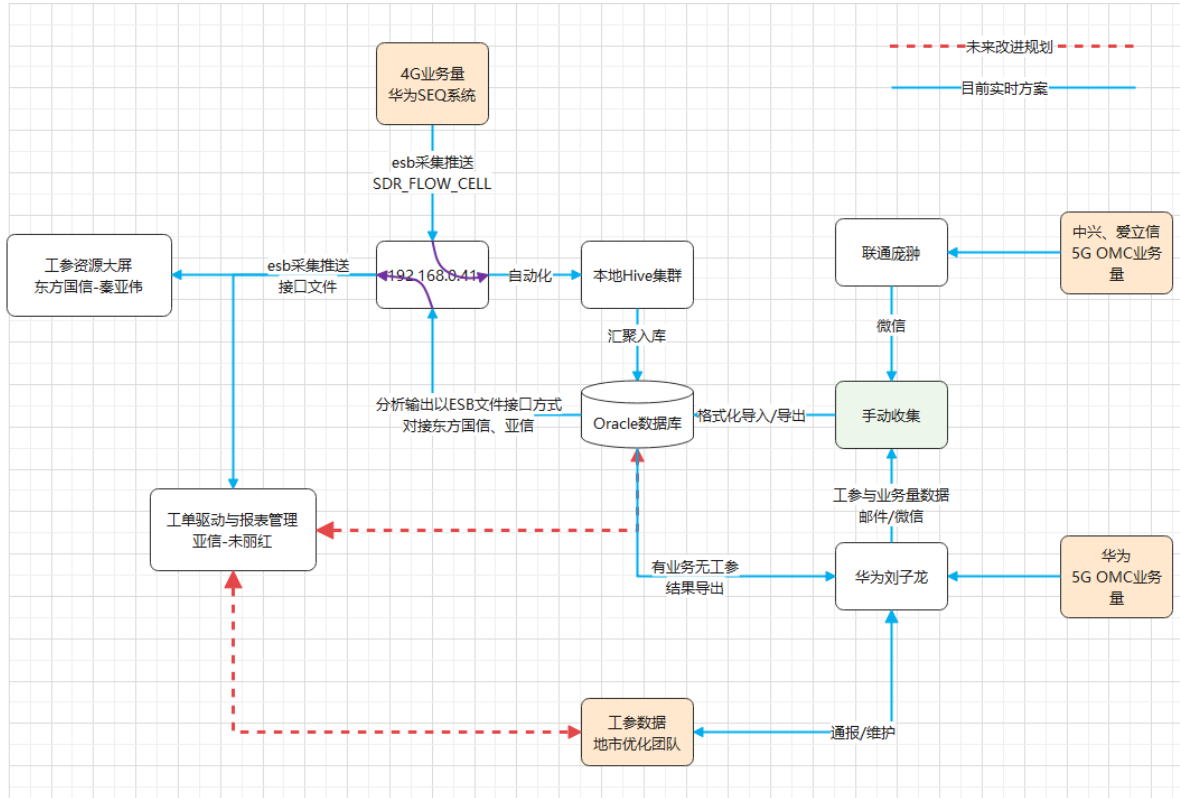


工参治理大屏总结

1、数据流图



2、数据源说明

- 1 # 1、工参数据
- 2 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。
- 3
- 4 # 2、4G业务量数据
- 5 华为SEQ定时输出-->省内ESB-->省内ESB服务器
- 6
- 7 # 3、5G业务量数据
- 8 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 # 1、4G工参数据
- 2 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。
- 3
- 4 # 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：
- 5 1) 工参字段与数据库字段映射（参考附表1）

6 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。

7

8 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。

9

10 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)

11

12 5) 备份数据库工参

13 SELECT * FROM seq_4g_siteinfo --4G工参数据表

14

15 # 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

16 call pkg_basedata_governance_v4.backup_4g_siftinfo();

17 # 查看历史备份

18 SELECT sdate,count(*) t FROM SEQ_4G_SITEINFO_backup group by sdate

19

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

21

22 7) 执行如下命令即可

23 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.ctf

1 # 2、5G工参数据

2 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。

3

4 # 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:

5 1) 工参字段与数据库字段映射 (参考附表1)

6 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。

7

8 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。

9

10 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

11

12 5) 备份数据库工参

13 SELECT * FROM seq_5g_siteinfo --5G工参数据表

14

15 # 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

16 call pkg_basedata_governance_v4.backup_5g_siftinfo();

17 # 查看历史备份

18 SELECT sdate,count(*) t FROM SEQ_5G_SITEINFO_backup group by sdate

19

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

21

22 7) 执行如下命令即可

23 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.ctf

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

```

1  # 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）
2  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)
3  sqmdb_local.sdr_flow_cell_15min #入库后的原始XDR数据表名
4  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。
5  #该环节由李旋提供运维。
6
7  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:
8  sqmdb_local.sdr_flow_4g_cell_day #hive中汇聚出的4g小区天表
9  sqmdb_local.sdr_flow_4g_cell_hour #hive中汇聚出的4g小区小时表
10 sqmdb_local.sdr_flow_4g_enb_day #hive中汇聚出的4g基站天表
11 sqmdb_local.sdr_flow_4g_enb_hour #hive中汇聚出的4g基站小时表
12
13 3) 汇聚与入库命令 192.168.0.9 定时任务crontab
14 #sqmdb_local汇聚及入库
15 30 22 * * * /home/do/hj/hive2ora/sdr_flow.sh >>
   /home/do/hj/hive2ora/sdr_flow.log 2>&1 &
16
17 # 192.168.0.9 /home/do/hj/hive2ora/sdr_flow.sh文件内容
18 #####
19 # File Name: sdr_flow.sh
20 # Author: haojian
21 # mail: hjwiki@gmail.com
22 # Created Time: 2021-03-31 20:12:17
23 # 在hive汇聚数据并将结果转存到oracle
24 #####
25 #!/bin/bash
26 set -x
27 yesterday=`date -d "-1 day" "+%Y%m%d"`
28 cd /home/do/zhanglei
29 sh pkg.sh $yesterday > /home/do/hj/hive2ora/log/pkg$yesterday.log 2>&1
30 cd /home/do/hj/hive2ora
31 ./hdfsget.sh $yesterday > /home/do/hj/hive2ora/log/hive2ora$yesterday.log
   2>&1
32
33 # pkg.sh内容键附表3。
34 #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。

```

1  [do@hbunicom-web zhanglei]$ cat pkg.sh
2  startDate=$1
3
4  hsq1="
5  set hive.execution.engine=mr;
6  set mapreduce.job.queueName=hbase;
7  set io.sort.mb=200;
8  set hive.tez.container.size=2048;
9  SET mapreduce.reduce.memory.mb=8192;
10 SET mapreduce.map.memory.mb=4096;
11 set hive.input.format=org.apache.hadoop.hive.ql.io.CombineHiveInputFormat;
12 set mapred.max.split.size=256000000;
13 set mapred.min.split.size.per.node=100000000;
14 set mapred.min.split.size.per.rack=100000000;
15 set hive.merge.mapfiles = true;
16 set hive.merge.tezfiles=true;
17 set hive.merge.mapredfiles = true;
18 set hive.merge.size.per.task = 256000000;
19 set hive.merge.smallfiles.avgsize=16000000;
20 set hive.exec.dynamic.partition.mode=nonstrict;
21 set hive.exec.dynamic.partition =true;
22 set hive.exec.max.dynamic.partitions.pernode=100;
23 set hive.exec.max.dynamic.partitions =1000;
24 set hive.exec.max.created.files =100000;
25 use sqmdb_local;
26
27
28 insert overwrite table sdr_flow_4g_cell_hour partition(sdate)
29 select
30 enbid,
31 cell_id,
32 sum(l4_dw_throughput) l4_dw_throughput,
33 sum(l4_ul_throughput) l4_ul_throughput,
34 sum(throughput) throughput,
35 layer2id,
36 layer3id,
37 sdate
38 from (SELECT from_unixtime(cast(starttime as int),'yyyyMMddHH') sdate,
39 conv((case when length(cgisai) = 12 then substr(cgisai, 6, 5) else
40 substr(cgisai, 6, 4) end), 16,10) enbid,
41 conv((case when length(cgisai) = 12 then substr(cgisai, -2) else
42 substr(cgisai, -4) end), 16,10) cell_id,
43 l4_dw_throughput,
44 l4_ul_throughput,
45 (l4_dw_throughput + l4_ul_throughput) throughput,
46 layer2id,
47 layer3id
48 FROM sdr_flow_cell_15min t
49 WHERE T.RAT = 6
50 and length(cgisai) = 12
51 and p_hour >="`$startDate`00'
52 and p_hour <
53 from_unixtime(unix_timestamp(date_add(from_unixtime(unix_timestamp('$startDate', 'yyyymmdd'), 'yyyymmdd'), 3), 'yyyymmdd'), 'yyyymmddhh00')
54 and from_unixtime(cast(starttime as int),'yyyyMMdd') = "`$startDate`" ) m

```



```

52 group by enbid, cell_id, sdate, layer2id, layer3id ;
53
54
55 insert overwrite table sdr_flow_4g_cell_day partition(sdate)
56 select
57 enbid,
58 cell_id,
59 sum(14_dw_throughput) 14_dw_throughput,
60 sum(14_ul_throughput) 14_ul_throughput,
61 sum(throughput) throughput,
62 layer2id,
63 layer3id,
64 sdate
65 from (SELECT from_unixtime(cast(starttime as int),'yyyyMMdd') sdate,
66 conv((case when length(cgisai) = 12 then substr(cgisai, 6, 5) else
67 substr(cgisai, 6, 4) end),16,10) enbid,
68 conv((case when length(cgisai) = 12 then substr(cgisai, -2) else
69 substr(cgisai, -4) end),16,10) cell_id,
70 14_dw_throughput,
71 14_ul_throughput,
72 (14_dw_throughput + 14_ul_throughput) throughput,
73 layer2id,
74 layer3id
75 FROM sdr_flow_cell_15min t
76 WHERE T.RAT = 6
77 and length(cgisai) = 12
78 and p_hour >= '$startDate'00'
79 and p_hour <
80 from_unixtime(unix_timestamp(date_add(from_unixtime(unix_timestamp('$startDate', 'yyyymmdd'), 'yyyy-mm-dd'), 3), 'yyyy-mm-dd'), 'yyyymmddhh00')
81 and from_unixtime(cast(starttime as int),'yyyyMMdd') = '$startDate') m
82 group by sdate, enbid, cell_id, layer2id, layer3id;
83
84 insert overwrite table sdr_flow_4g_enb_hour partition(sdate)
85 select enbid,
86 sum(14_dw_throughput) 14_dw_throughput,
87 sum(14_ul_throughput) 14_ul_throughput,
88 sum(throughput) throughput,
89 layer2id,
90 layer3id,
91 sdate
92 from (SELECT from_unixtime(cast(starttime as int),'yyyyMMddHH') sdate,
93 conv((case when length(cgisai) = 12 then substr(cgisai, 6, 5) else
94 substr(cgisai, 6, 4) end),16,10) enbid,
95 14_dw_throughput,
96 14_ul_throughput,
97 (14_dw_throughput + 14_ul_throughput) throughput,
98 layer2id,
99 layer3id
100 FROM sdr_flow_cell_15min t
101 WHERE T.RAT = 6
102 and length(cgisai) = 12
103 and p_hour >= '$startDate'00'
104 and p_hour <
105 from_unixtime(unix_timestamp(date_add(from_unixtime(unix_timestamp('$startDate', 'yyyymmdd'), 'yyyy-mm-dd'), 3), 'yyyy-mm-dd'), 'yyyymmddhh00')
106 and from_unixtime(cast(starttime as int),'yyyyMMdd') = '$startDate') m
107 group by sdate, enbid, layer2id, layer3id;

```

```

103
104
105 insert overwrite table sdr_flow_4g_enb_day partition(sdate)
106 select enbid,
107 sum(14_dw_throughput) 14_dw_throughput,
108 sum(14_ul_throughput) 14_ul_throughput,
109 sum(throughput) throughput,
110 layer2id,
111 layer3id,
112 sdate
113 from (SELECT from_unixtime(cast(starttime as int),'yyyyMMdd') sdate,
114 conv((case when length(cgisai) = 12 then substr(cgisai, 6, 5) else
115 substr(cgisai, 6, 4) end),16,10) enbid,
116 conv((case when length(cgisai) = 12 then substr(cgisai, -2) else
117 substr(cgisai, -4) end),16,10) cell_id,
118 14_dw_throughput,
119 14_ul_throughput,
120 (14_dw_throughput + 14_ul_throughput) throughput,
121 layer2id,
122 layer3id
123 FROM sdr_flow_cell_15min t
124 WHERE T.RAT = 6
125 and length(cgisai) = 12
126 and p_hour >= '$startDate'00'
127 and p_hour <
128 from_unixtime(unix_timestamp(date_add(from_unixtime(unix_timestamp('$startDate', 'yyyymmdd'), 'yyyy-mm-dd'), 3), 'yyyy-mm-dd'), 'yyyymmddhh00')
129 and from_unixtime(cast(starttime as int),'yyyyMMdd') = '$startDate') m
130 group by sdate, enbid, layer2id, layer3id;
131 "
132
133 #echo $hsq1 #>./h.sql
134
135 hive -e "$hsq1"

```

2.3 5G业务量

- 1 # 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、爱立信由庞种每周一发送上周一至上周末的数据（通常不准时）。
- 2
- 3 # 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。
- 4
- 5 # 2) 导入的目标表：sdr_flow_cell_throughput

2.4 业务量数据检查

```

1 --按地市统计
2 SELECT * FROM
3 (
4 SELECT sdate,city,COUNT(*) COUNT_NUM from sdr_flow_cell_throughput
5 where sdate>sysdate-20
6 GROUP BY sdate,city
7 )

```

```

8 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'))
9 ORDER BY sdate
10
11 --按厂家统计
12 SELECT * FROM
13 (
14 SELECT sdate,vender,COUNT(*) COUNT_NUM from sdr_flow_cell_throughput
15 where sdate>sysdate-20
16 GROUP BY sdate,vender
17 )
18 PIVOT (SUM(COUNT_NUM) total FOR vender IN ('华为', '中兴', '爱立信'))
19 ORDER BY sdate
20

```

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

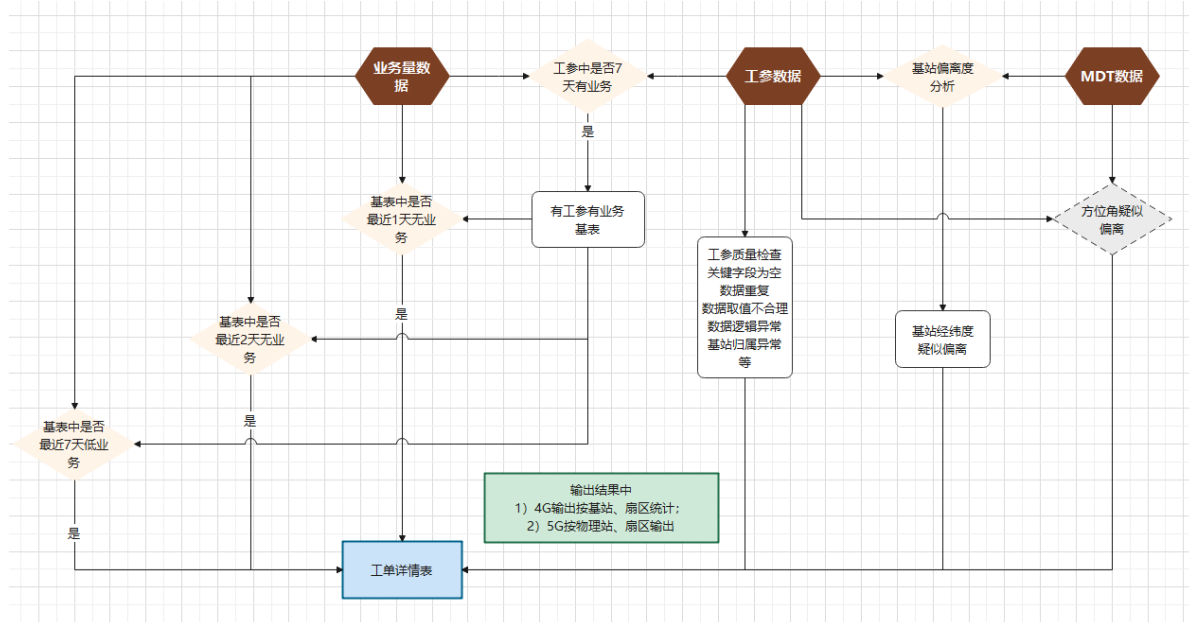
```

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

```

625      from (select max(sdate) sdate from (select city_name, max(sdate) sdate from STATS_5g_CELL_FLOW_2M_DAY group by city
626      union
627      select trunc(sysdate) from dual
628      );
629      dbms_output.put_line('cal_5g_data: start date - ' || to_char(istart, 'yyyymmdd'));
630      select min(sdate) sdate_min
631      into iend
632      from (select city, max(sdate) sdate from SDR_FLOW_CELL_THROUGHPUT group by city
633      union
634      select 'system',trunc(sysdate) from dual
635      );
636      dbms_output.put_line('cal_5g_data: end date - ' || to_char(iend, 'yyyymmdd'));
637      --获得结果表的最小日期
638      --dbms_output.put_line(istart);
639      --dbms_output.put_line(iend);
640      --计算时间差，用于实现遍历的时间列表
641      igap := iend - istart;
642      --dbms_output.put_line(igap);
643      if igap >= 1 then
644      for v in (select istart + level sdate from dual connect by level <= igap) loop
645      dbms_output.put_line('cal_5g_data: date - ' || to_char(v.sdate, 'yyyymmdd') || ' start process');
646      sdr_flow_5g_cell_day(to_char(v.sdate, 'yyyymmdd'));
647      sdr_flow_5g_enb_day(to_char(v.sdate, 'yyyymmdd'));
648      -----
649      STATS_5g_SITE_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
650      STATS_5g_phySITE_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
651      STATS_5g_CELL_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
652      -----
653      STATS_5g_SITE_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);
654      STATS_5g_phySITE_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);
655      STATS_5g_CELL_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);
656      -----
657      STATS_5g_SITE_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);
658      STATS_5g_phySITE_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);
659      STATS_5g_CELL_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);
660      -----
661      STATS_5g_SITE_LOWFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
662      STATS_5g_CELL_LOWFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
663      STATS_5g_phySITE_LOWFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
664

```

3.3 逻辑说明

- 1 # 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 # 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 # 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 # 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 # 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 # 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 # 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 # 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  # 1) 基站经纬度偏离目前每月计算一次。
2  # 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。
3  create table hb_mdt_grid_cell_manual as
4  select
5  day sdate,
6  city,
7  floor(s_cell_id/256) enbid,
8  s_cell_id eci,
9  floor(orig_lon/0.00045) gridx,
10 floor(orig_lat/0.00045) gridy,
11 sum(case when (s_rsrp<-20 and s_rsrp>-140) then 1 else 0 end) rsrp_samples

```

```

12 from fastdo_lte.tdlte_mro_locate_hour
13 where orig_lon between 113 and 120
14 and orig_lat between 36 and 43
15 and day = '20210512'
16 and s_cell_id >0
17 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

```

1 # 2) 汇聚结果导出
2 # 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。
3
4 # 单地市导出
5 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_z1 where ct='BAODING' and rsrp_samples>5
and ds='20210726'" > BAODING.csv
6
7 # 所有地市批量导出
8 for city in
{BAODING,CANGZHOU,CHENGDE,HANDAN,HENGSHUI,LANGFANG,QINHUANGDAO,SHIJIAZHUANG,
TANGSHAN,XINGTAI,XIONGAN,ZHANGJIAKOU}; do echo
9 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_z1 where ct='$city' and rsrp_samples>5
and ds='20210726'" > $city.csv';
10 done

```

3.5.3 聚类处理

```

1 # python脚本路径在133.96.92.137
C:\Users\zhanglei\PycharmProjects\untitled2\dbscan.py, 内容如下
2
3 # -*- coding: utf-8 -*-
4 """
5 -----
6     File Name:   deviate_hebei
7     Author:     ZhangLei
8     date:       2021/05/08
9     -----
10 """
11
12 import sklearn.cluster as skc
13 import pandas as pd
14 from pandas import DataFrame
15 import os
16 import multiprocessing as mp
17 import logging
18
19
20 # 主处理函数
21 def process(icity):
22     logging.basicConfig(level=logging.DEBUG,
23                         filename=icity + '.log',

```



```

24         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
25         # a是追加模式, 默认如果不写的话, 就是追加模式
26         format='%(%asctime)s - [%(filename)s-%(funcName)s-
line:%(lineno)d] - %(levelname)s: %(message)s'
27         # 日志格式
28     )
29
30     name = mp.current_process().name
31     logging.debug("进程:" + name + "启动!")
32     print("进程:" + name + "启动!")
33     os.chdir(r'./MDT_GRID')
34
35     # 后期需自定义修改文件名,用于适配
36     infile = os.path.join(os.getcwd(), icity + '.csv')
37     logging.debug('文件地址:' + infile)
38     # 打印输入的日志文件
39     logging.debug('输入文件:' + infile)
40
41     # 注意: 需要根据文件的分隔符进行修改
42     # data = pd.read_csv(infile,
43     #                     usecols=['sdate', 'city', 'enbid', 'eci', 'gridx',
'gridy', 'rsrp_samples'], sep='\t')
44     data = pd.read_csv(infile,
45     usecols=['sdate', 'city', 'enbid', 'eci', 'gridx',
'gridy', 'rsrp_samples'], sep=',')
46
47     enodebs = data['enbid'].drop_duplicates() # 遍历enodeb
48
49     logging.debug('待处理基站数量:' + str(len(enodebs)))
50
51     result_circle = []
52
53     for enbid in enodebs:
54         logging.debug("###处理基站编号: " + str(enbid))
55         temp = data[data['enbid'].isin([enbid])]
56         X = temp[['gridx', 'gridy']].drop_duplicates()
57         for min_sample in range(41, 1, -2):
58             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
59             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
60             raito = len(labels[labels[:] == -1]) / len(labels) # 计算噪声点个
    数占总数的比例
61             if raito < 0.2:
62                 sample_select = min_sample
63                 logging.debug('满足噪声比例的MIN_SAMPLE值: ' +
str(sample_select) + ', 噪声比为: ' + str(raito))
64                 break
65
66             # 获取分簇的数目
67             # labels结果如[ 0 1 0 2 0 1 1 -1 0 2 2], 一般-1表示噪声点
68             n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
69
70             logging.debug('分簇的数目: %d' % n_clusters_)
71
72             # print("轮廓系数: %0.3f" % metrics.silhouette_score(X, labels)) #轮廓
    系数评价聚类的好坏

```

```

73     cluster_grid = X[labels != -1]
74
75     # 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
76     for index, row in cluster_grid.iterrows():
77         result_circle.append([enbid, row["gridx"], row["gridy"]])
78
79     outputfile = os.path.join(os.getcwd(), icity + '_DBSCAN_SITE.csv')
80     logging.debug('输出文件:' + outputfile)
81     data_yuanxin = DataFrame(result_circle)
82     data_yuanxin.columns = ['enbid', 'gridx', 'gridy']
83     data_yuanxin.to_csv(outputfile, index=False, encoding='gbk')
84
85
86 if __name__ == '__main__':
87     # 单进程运行
88     # process('HENGSHUI')
89     # exit(0)
90
91     # 多线程运行
92     citys = ['BAODING', 'CANGZHOU', 'CHENGDE', 'HANDAN', 'HENGSHUI',
    'LANGFANG', 'QINHUANGDAO',
93             'SHIJIAZHUANG', 'TANGSHAN', 'XINGTAI', 'XIONGAN',
    'ZHANGJIAKOU']
94     for v in citys:
95         p = mp.Process(name='子线程_' + v, target=process, args=(v,))
96         p.start()
97     p.join()
98
99     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

```

1 # 检查并修改dbscan_site.ctl、mdt_site.ctl文件内容，默认不需要更新。
2 # 在当前目录的命令行窗口，逐行执行sqlldr.txt中的语句，即：
3 sqlldr c##fast491/\ "F@st491*321\"@192.168.0.64:1521/fast
   control=C:\Users\zhanglei\PycharmProjects\untitled2\MDT_GRID\mdt_site.ctl
4
5 sqlldr c##fast491/\ "F@st491*321\"@192.168.0.64:1521/fast
   control=C:\Users\zhanglei\PycharmProjects\untitled2\MDT_GRID\dbscan_site.ctl
6
7

```

3.5.5 偏离分析

```

1 # 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)。
2 # 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):
3
4 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
5 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 --1、初始化表(清空所有结果表), 按需执行
2 --call pkg_basedata_governance_v4.ini_4g_table();--会清空所有4g结果表, 谨慎
3 --call pkg_basedata_governance_v4.ini_5g_table();--会清空所有5g结果表, 谨慎
4
5 --2、批量执行计算, 目前暂不使用。
6 call pkg_basedata_governance_v4.cal_4g_data();
7 call pkg_basedata_governance_v4.cal_5g_data();
8
9 --3、按指定日期手动计算, 目前的使用方式
10 # 业务量分析处理
11 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
12 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
13
14 # 工参质量检查
15 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
16 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
17
18 # 基站偏离度检查
19 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
20 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
21 # 逻辑步骤说明
22 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;
23 2) fdd_MDT_ENB_position: 以fdd_MDT_cell_position、seq_4g_siteinfo, 计算物理站的虚拟经纬度;
24 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;
25 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);
26 5) 按需根据偏离距离过滤并进行投诉转置
27
28
29 --4、统计有业务无工参的小区, 需手动调用
30 # 有业务无工参的报表统计
31 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

```

```
32 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

```
1 # 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。
2 # 4接口用到。
3
4 # 1) v1.4 版本报表-省级粒度-excel展示汇报
5 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
6
7 # 2) v1.4 版本报表-市级粒度-excel展示汇报
8 SELECT * FROM v_STATS_BIGSCREEN_WORKORDER_21 --20210707 update
9
10 # 3) v1.4 版本报表-市级粒度-excel展示汇报--暂未使用
11 SELECT * FROM v_STATS_BIGSCREEN_WORKORDER_32 --20210707 update
12
13 # 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
14 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
15
16
17 # 5) v1.4 版本报表-新增按照运营商进行统计, excel展示汇报
18 SELECT * FROM v_STATS_BIGSCREEN_SUMMARY_22_Y --20210707 update
19
20 # 6) v1.4 版本报表-统计透视, 按承建方透视统计, excel展示汇报
21 SELECT *
22 FROM (SELECT * FROM v_STATS_BIGSCREEN_SUMMARY_22_Y)
23
24 PIVOT(SUM(CELL_TOTAL) CELL_TOTAL, sum(CELL_NUM) CELL_NUM, sum(RATE) RATE
25 FOR CONSTRUCTION IN('联通扇区', '电信扇区'))
26 ORDER BY DATA_TYPE
```

5.3 推送结果到ESB接口机

```
1 # 在数据检查无误后, 需要将数据上传接口机。
2 # 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。
3 # D:\gongcanzhili\02_gczl\10接口程序源码\oracle_2_esb\gooracle.exe
4
5 # 日志信息如下:
6 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
7 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
8 Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1315: Ftp连接成功
9 Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1322: Ftp登录成功
10 Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1329: 当前工作目
录:/data
11 Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1336: 切换目录成
功:/data/bigscreen
12 Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1357: 创建目
录:20210817 成功
13 Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1365: 切换目录成功
```

```
14 Test- 2021/08/17 09:03:52 query_data_from_oracle_2_csv.go:1371: 打印当前目
录:20210817
15 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...
16 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
17 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...
18 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
19 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...
20 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
21
22 # 2) 检查文件上传结果: 登录192.168.0.41服务器
23 [do@hbunicom-nifi-41 20210817]$ pwd
24 /data/bigscreen/20210817
25 [do@hbunicom-nifi-41 20210817]$ ll
26 总用量 176268
27 -rw-r--r-- 1 do do 377585 8月 17 09:01 gcToyxgx_00_bigscreen_20210817.csv
28 -rw-r--r-- 1 do do 12100785 8月 17 09:01
gcToyxgx_01_4g_site_youyewu_7d_20210817.csv
29 -rw-r--r-- 1 do do 41556169 8月 17 09:01
gcToyxgx_02_4g_cell_youyewu_7d_20210817.csv
30 -rw-r--r-- 1 do do 1893669 8月 17 09:01
gcToyxgx_03_5g_site_youyewu_7d_20210817.csv
31 -rw-r--r-- 1 do do 7883155 8月 17 09:01
gcToyxgx_04_5g_cell_youyewu_7d_20210817.csv
32 -rw-r--r-- 1 do do 45842 8月 17 09:01
gcToyxgx_05_4g_site_wuyewu_2d_20210817.csv
33 -rw-r--r-- 1 do do 497571 8月 17 09:01
gcToyxgx_06_4g_cell_wuyewu_2d_20210817.csv
34 -rw-r--r-- 1 do do 2592161 8月 17 09:01
gcToyxgx_07_4g_site_diyewu_2d_20210817.csv
35 -rw-r--r-- 1 do do 14921970 8月 17 09:01
gcToyxgx_08_4g_cell_diyewu_2d_20210817.csv
36 -rw-r--r-- 1 do do 11789 8月 17 09:01
gcToyxgx_09_5g_site_wuyewu_2d_20210817.csv
37 -rw-r--r-- 1 do do 70762 8月 17 09:01
gcToyxgx_10_5g_cell_wuyewu_2d_20210817.csv
38 -rw-r--r-- 1 do do 73151 8月 17 09:01
gcToyxgx_11_5g_site_diyewu_2d_20210817.csv
39 -rw-r--r-- 1 do do 303511 8月 17 09:01
gcToyxgx_12_5g_cell_diyewu_2d_20210817.csv
40 -rw-r--r-- 1 do do 64798 8月 17 09:01
gcToyxgx_13_4g_site_wuyewu_1d_20210817.csv
41 -rw-r--r-- 1 do do 742229 8月 17 09:01
gcToyxgx_14_4g_cell_wuyewu_1d_20210817.csv
42 -rw-r--r-- 1 do do 18369 8月 17 09:01
gcToyxgx_15_5g_site_wuyewu_1d_20210817.csv
43 -rw-r--r-- 1 do do 107904 8月 17 09:01
gcToyxgx_16_5g_cell_wuyewu_1d_20210817.csv
44 -rw-r--r-- 1 do do 5168 8月 17 09:01
gcToyxgx_17_4g_cell_errors_20210817.csv
45 -rw-r--r-- 1 do do 4040 8月 17 09:01
gcToyxgx_18_5g_cell_errors_20210817.csv
```

```

46 | -rw-r--r-- 1 do do 70428 8月 17 09:01
    | gcToyxgx_19_4g_site_deviate_20210817.csv
47 | -rw-r--r-- 1 do do 83563954 8月 17 09:01
    | gcToyxgx_20_4g_seq_siteinfo_20210817.csv
48 | -rw-r--r-- 1 do do 12876151 8月 17 09:01
    | gcToyxgx_21_5g_seq_siteinfo_20210817.csv
49 | -rw-r--r-- 1 do do 567631 8月 17 09:01
    | gcToyxgx_22_4g_site_deviate1km_20210817.csv
50 | -rw-r--r-- 1 do do 93183 8月 17 09:01
    | gcToyxgx_23_4g_site_deviate2km_20210817.csv
51 | -rw-rw-r-- 1 do do 64 8月 17 09:04 note.txt
52 | [do@hbunicom-nifi-41 20210817]$
53 |
54 | # 3) 其它一些注意事项
55 | 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 # 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
- 3 # 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。
- 4
- 5 # 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_bigsreen_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/bigsreen/20210517/gcToyxgx_00_bigsreen_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

```

1 | # 00 --大屏报表数据
2 | select DATA_TYPE      "业务名称",
3 |        district_name  "行政区名称",
4 |        district_code  "行政区编码",
5 |        site_num       "小区数量",
6 |        DATA_NUM      "单位数量",
7 |        rate           "百分比",
8 |        CONSTRUCTION    "承建方"
9 | from v_STATS_BIGSCREEN_SUMMARY_32
10 |

```



```

66         lat,
67         vender,
68         construction
69     from seq_4g_siteinfo
70     where sdate = (select max(sdate) from seq_4g_siteinfo)) a,
71     (select distinct city_name, enbid, cell_id, THROUGHPUT
72     from STATS_4g_CELL_FLOW_2M_DAY
73     where sdate =
74         (select max(sdate) from STATS_4g_CELL_FLOW_2M_DAY)) b
75     where /*a.city_name=b.city_name and*/
76     a.enbid = b.enbid
77     and a.cell_id = b.cell_id
78
79 # 03 1月有业务的5g基站数
80     select distinct a.city_name "地市名称",
81         a.DISTRICT_NAME "行政区名称",
82         a.CITY_CODE "地市编码",
83         a.DISTRICT_CODE "行政区编码",
84         a.PHYSTATION_ADDRESS "物理站名称",
85         a.STATION_NAME "基站名称",
86         to_char(a.gnbid) "基站GNBID",
87         a.vender "厂家",
88         round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位
是Mbyte, 需要除以1024
89     ,
90     CONSTRUCTION "承建方"
91     from (select distinct city_name,
92         DISTRICT_NAME,
93         CITY_CODE,
94         DISTRICT_CODE,
95         PHYSTATION_ADDRESS,
96         STATION_NAME,
97         gnbid,
98         vender,
99         first_value(construction) over(partition by
district_name, district_code, gnbid) construction
100     from seq_5g_siteinfo
101     where sdate = (select max(sdate) from seq_5g_siteinfo)
102     and is_alive = 1) a,
103     (select distinct city_name, PHYSTATION_ADDRESS, gnbid,
THROUGHPUT
104     from STATS_5g_PHYSITE_FLOW_2M_DAY
105     where sdate =
106         (select max(sdate) from STATS_5g_PHYSITE_FLOW_2M_DAY)) b
107     where /* a.city_name=b.city_name and*/
108     a.gnbid = b.gnbid
109     and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
110
111 # 04 1月有业务的5g小区数
112     select distinct a.city_name "地市名称",
113         a.DISTRICT_NAME "行政区名称",
114         a.CITY_CODE "地市编码",
115         a.DISTRICT_CODE "行政区编码",
116         a.PHYSTATION_ADDRESS "物理站名称",
117         a.STATION_NAME "基站名称",
118         to_char(a.gnbid) "基站GNBID",
119         a.cell_name "小区名称",
120         a.cell_id "小区id",

```

```

121         a.lon "小区经度",
122         a.lat "小区纬度",
123         a.vender "厂家",
124         round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位
是Mbyte, 需要除以1024
125     ,
126     CONSTRUCTION "承建方"
127     from (select distinct city_name,
128             DISTRICT_NAME,
129             CITY_CODE,
130             DISTRICT_CODE,
131             PHYSTATION_ADDRESS,
132             STATION_NAME,
133             gnbid,
134             cell_name,
135             cell_id,
136             lon,
137             lat,
138             vender,
139             construction
140         from seq_5g_siteinfo
141         where sdate = (select max(sdate) from seq_5g_siteinfo)
142             and is_alive = 1) a,
143     (select distinct city_name, cell_id, gnbid, THROUGHPUT
144         from STATS_5g_CELL_FLOW_2M_DAY
145         where sdate =
146             (select max(sdate) from STATS_5g_CELL_FLOW_2M_DAY)) b
147     where /*a.city_name=b.city_name and */
148         a.gnbid = b.gnbid
149         and a.cell_id = b.cell_id
150
151     # 05 7天无业务4g基站数
152     select distinct a.CITY_NAME "地市名称",
153         a.DISTRICT_NAME "行政区名称",
154         a.CITY_CODE "地市编码",
155         a.DISTRICT_CODE "行政区编码",
156         a.PHYSTATION_ADDRESS "物理站名称",
157         a.BBU_NAME "BBU名称",
158         to_char(a.ENBID) "基站enbid",
159         a.vender "厂家",
160         0 "业务量",
161         construction "承建方"
162     from (select distinct CITY_NAME,
163             DISTRICT_NAME,
164             CITY_CODE,
165             DISTRICT_CODE,
166             PHYSTATION_ADDRESS,
167             BBU_NAME,
168             ENBID,
169             vender,
170             first_value(construction) over(partition by
district_name, district_code, enbid) construction
171         from seq_4g_siteinfo
172         where sdate = (select max(sdate) from seq_4g_siteinfo)
173             and is_alive = 1) a,
174     (select distinct city_name, enbid
175         from STATS_4g_SITE_FLOW_7d_DAY
176         where sdate =

```

```

177         (select max(sdate) from STATS_4g_SITE_FLOW_7d_DAY)) b
178     where /*a.city_name= b.city_name and*/
179     a.enbid = b.enbid
180
181 # 06 7天无业务4g小区数
182     select distinct a.CITY_NAME      "地市名称",
183                    a.DISTRICT_NAME  "行政区名称",
184                    a.CITY_CODE      "地市编码",
185                    a.DISTRICT_CODE  "行政区编码",
186                    a.PHYSTATION_ADDRESS "物理站名称",
187                    a.BBU_NAME      "BBU名称",
188                    to_char(a.ENBID)  "基站enbid",
189                    a.cell_id        "小区id",
190                    a.lon            "小区经度",
191                    a.lat            "小区纬度",
192                    a.vender        "厂家",
193                    0 "业务量",
194                    construction    "承建方"
195     from (select distinct CITY_NAME,
196                    DISTRICT_NAME,
197                    CITY_CODE,
198                    DISTRICT_CODE,
199                    PHYSTATION_ADDRESS,
200                    BBU_NAME,
201                    ENBID,
202                    cell_id,
203                    lon,
204                    lat,
205                    vender,
206                    construction
207     from seq_4g_siteinfo
208     where sdate = (select max(sdate) from seq_4g_siteinfo)
209     and is_alive = 1) a,
210     (select distinct city_name, enbid, cell_id
211     from STATS_4g_CELL_FLOW_7d_DAY
212     where sdate =
213         (select max(sdate) from STATS_4g_CELL_FLOW_7d_DAY)) b
214     where /*a.city_name= b.city_name and*/
215     a.enbid = b.enbid
216     and a.cell_id = b.cell_id
217
218 # 07 7天低业务4g基站数
219     select distinct a.CITY_NAME "地市名称",
220                    a.DISTRICT_NAME "行政区名称",
221                    a.CITY_CODE "地市编码",
222                    a.DISTRICT_CODE "行政区编码",
223                    a.PHYSTATION_ADDRESS "物理站名称",
224                    a.BBU_NAME "BBU名称",
225                    to_char(a.ENBID) "基站enbid",
226                    a.vender "厂家",
227                    round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
228     THROUGHPUT 单位是byte, 需要除以1024*1024*1024
229                    ,
230                    CONSTRUCTION "承建方"
231     from (select distinct CITY_NAME,
232                    DISTRICT_NAME,
233                    CITY_CODE,
234                    DISTRICT_CODE,

```

```

234         PHYSTATION_ADDRESS,
235         BBU_NAME,
236         ENBID,
237         vender,
238         first_value(construction) over(partition by
district_name, district_code, enbid) construction
239     from seq_4g_siteinfo
240     where sdate = (select max(sdate) from seq_4g_siteinfo)
241     and is_alive = 1) a,
242     (select distinct city_name, enbid, THROUGHPUT
243     from STATS_4g_SITE_lowFLOW_7d_DAY
244     where sdate =
245         (select max(sdate) from STATS_4g_SITE_lowFLOW_7d_DAY)) b
246     where /*a.city_name= b.city_name and*/
247     a.enbid = b.enbid
248
249 # 08 7天低业务4g小区数
250     select distinct a.CITY_NAME "地市名称",
251     a.DISTRICT_NAME "行政区名称",
252     a.CITY_CODE "地市编码",
253     a.DISTRICT_CODE "行政区编码",
254     a.PHYSTATION_ADDRESS "物理站名称",
255     a.BBU_NAME "BBU名称",
256     to_char(a.ENBID) "基站enbid",
257     a.cell_id "小区id",
258     a.lon "小区经度",
259     a.lat "小区纬度",
260     a.vender "厂家",
261     round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte, 需要除以1024*1024*1024
262     ,
263     CONSTRUCTION "承建方"
264     from (select distinct CITY_NAME,
265     DISTRICT_NAME,
266     CITY_CODE,
267     DISTRICT_CODE,
268     PHYSTATION_ADDRESS,
269     BBU_NAME,
270     ENBID,
271     cell_id,
272     lon,
273     lat,
274     vender,
275     construction
276     from seq_4g_siteinfo
277     where sdate = (select max(sdate) from seq_4g_siteinfo)
278     and is_alive = 1) a,
279     (select distinct city_name, enbid, cell_id, THROUGHPUT
280     from STATS_4g_CELL_lowFLOW_7d_DAY
281     where sdate =
282         (select max(sdate) from STATS_4g_CELL_lowFLOW_7d_DAY)) b
283     where /*a.city_name= b.city_name and*/
284     a.enbid = b.enbid
285     and a.cell_id = b.cell_id
286
287 # 09 7天无业务5g基站数 4492
288     select distinct a.city_name "地市名称",
289     a.DISTRICT_NAME "行政区名称",

```

```

290         a.CITY_CODE           "地市编码",
291         a.DISTRICT_CODE       "行政区编码",
292         a.PHYSTATION_ADDRESS  "物理站名称",
293         a.STATION_NAME        "基站名称",
294         to_char(a.gnbid)       "基站GNBID",
295         a.vender              "厂家",
296         0 "业务量",
297         CONSTRUCTION "承建方"
298     from (select distinct city_name,
299             DISTRICT_NAME,
300             CITY_CODE,
301             DISTRICT_CODE,
302             PHYSTATION_ADDRESS,
303             STATION_NAME,
304             gnbid,
305             vender,
306             first_value(construction) over(partition by
district_name, district_code, gnbid) construction
307         from seq_5g_siteinfo
308         where sdate = (select max(sdate) from seq_5g_siteinfo)
309             and is_alive = 1) a,
310     (select distinct city_name, PHYSTATION_ADDRESS, GNBID
311     from STATS_5g_phySITE_FLOW_7d_DAY
312     where sdate =
313         (select max(sdate) from STATS_5g_phySITE_FLOW_7d_DAY)) b
314     where /*a.city_name= b.city_name and*/
315     a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
316     and a.GNBID = b.GNBID
317
318     # 10 7天无业务5g小区数
319     select distinct a.city_name           "地市名称",
320         a.DISTRICT_NAME                   "行政区名称",
321         a.CITY_CODE                       "地市编码",
322         a.DISTRICT_CODE                   "行政区编码",
323         a.PHYSTATION_ADDRESS              "物理站名称",
324         a.STATION_NAME                    "基站名称",
325         to_char(a.gnbid)                  "基站GNBID",
326         a.cell_name                       "小区名称",
327         a.cell_id                         "小区id",
328         a.lon                             "小区经度",
329         a.lat                             "小区纬度",
330         a.vender                          "厂家",
331         0 "业务量",
332         CONSTRUCTION "承建方"
333     from (select distinct city_name,
334             DISTRICT_NAME,
335             CITY_CODE,
336             DISTRICT_CODE,
337             PHYSTATION_ADDRESS,
338             STATION_NAME,
339             gnbid,
340             cell_name,
341             cell_id,
342             lon,
343             lat,
344             vender,
345             construction
346         from seq_5g_siteinfo

```

```

347         where sdate = (select max(sdate) from seq_5g_siteinfo)
348             and is_alive = 1) a,
349     (select distinct city_name, GNBID, cell_id
350      from STATS_5g_CELL_FLOW_7d_DAY
351      where sdate =
352             (select max(sdate) from STATS_5g_CELL_FLOW_7d_DAY)) b
353 where /*a.city_name= b.city_name and*/
354 a.GNBID = b.GNBID
355 and a.cell_id = b.cell_id
356
357 # 11 7天低业务5g基站数
358 select distinct a.city_name "地市名称",
359                a.DISTRICT_NAME "行政区名称",
360                a.CITY_CODE "地市编码",
361                a.DISTRICT_CODE "行政区编码",
362                a.PHYSTATION_ADDRESS "物理站名称",
363                a.STATION_NAME "基站名称",
364                to_char(a.gnbid) "基站GNBID",
365                a.vender "厂家",
366                round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位
是Mbyte, 需要除以1024,
367                ,
368                CONSTRUCTION "承建方"
369 from (select distinct city_name,
370                    DISTRICT_NAME,
371                    CITY_CODE,
372                    DISTRICT_CODE,
373                    PHYSTATION_ADDRESS,
374                    STATION_NAME,
375                    gnbid,
376                    vender,
377                    first_value(construction) over(partition by
district_name, district_code, gnbid) construction
378 from seq_5g_siteinfo
379 where sdate = (select max(sdate) from seq_5g_siteinfo)
380             and is_alive = 1) a,
381     (select distinct city_name, PHYSTATION_ADDRESS, GNBID,
THROUGHPUT
382      from STATS_5g_PHYSITE_LOFLOW_7d_DAY
383      where sdate =
384             (select max(sdate) from STATS_5g_PHYSITE_LOFLOW_7d_DAY))
b
385 where /*a.city_name= b.city_name and*/
386 a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
387 and a.GNBID = b.GNBID
388
389 # 12 7天低业务5g小区数
390 select distinct a.city_name "地市名称",
391                a.DISTRICT_NAME "行政区名称",
392                a.CITY_CODE "地市编码",
393                a.DISTRICT_CODE "行政区编码",
394                a.PHYSTATION_ADDRESS "物理站名称",
395                a.STATION_NAME "基站名称",
396                to_char(a.gnbid) "基站GNBID",
397                a.cell_name "小区名称",
398                a.cell_id "小区id",
399                a.lon "小区经度",
400                a.lat "小区纬度",

```

```

401         a.vender "厂家",
402         round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单位
是Mbyte, 需要除以1024
403     ,
404     CONSTRUCTION "承建方"
405     from (select distinct city_name,
406             DISTRICT_NAME,
407             CITY_CODE,
408             DISTRICT_CODE,
409             PHYSTATION_ADDRESS,
410             STATION_NAME,
411             gnbid,
412             cell_name,
413             cell_id,
414             lon,
415             lat,
416             vender,
417             construction
418         from seq_5g_siteinfo
419         where sdate = (select max(sdate) from seq_5g_siteinfo)
420             and is_alive = 1) a,
421     (select distinct city_name, GNBID, cell_id, THROUGHPUT
422     from STATS_5g_CELL_LOWFLOW_7d_DAY
423     where sdate =
424         (select max(sdate) from STATS_5g_CELL_LOWFLOW_7d_DAY)) b
425     where /*a.city_name= b.city_name and*/
426     a.GNBID = b.GNBID
427     and a.cell_id = b.cell_id
428
429     # 13 48小时无业务4g基站数
430     select distinct a.CITY_NAME           "地市名称",
431         a.DISTRICT_NAME           "行政区名称",
432         a.CITY_CODE           "地市编码",
433         a.DISTRICT_CODE           "行政区编码",
434         a.PHYSTATION_ADDRESS "物理站名称",
435         a.BBU_NAME           "BBU名称",
436         to_char(a.ENBID)           "基站enbid",
437         a.vender           "厂家",
438         0 "业务量",
439         CONSTRUCTION "承建方"
440     from (select distinct CITY_NAME,
441             DISTRICT_NAME,
442             CITY_CODE,
443             DISTRICT_CODE,
444             PHYSTATION_ADDRESS,
445             BBU_NAME,
446             ENBID,
447             vender,
448             first_value(construction) over(partition by
district_name, district_code, enbid) construction
449         from seq_4g_siteinfo
450         where sdate = (select max(sdate) from seq_4g_siteinfo)
451             and is_alive = 1) a,
452     (select distinct city_name, enbid
453     from STATS_4g_SITE_FLOW_48h_hour
454     where sdate =
455         (select max(sdate) from STATS_4g_SITE_FLOW_48h_hour)) b
456     where /*a.city_name= b.city_name and*/

```

```

457     a.enbid = b.enbid
458
459 # 14 48小时无业务4g小区数
460     select distinct a.CITY_NAME           "地市名称",
461                    a.DISTRICT_NAME       "行政区名称",
462                    a.CITY_CODE           "地市编码",
463                    a.DISTRICT_CODE       "行政区编码",
464                    a.PHYSTATION_ADDRESS  "物理站名称",
465                    a.BBU_NAME            "BBU名称",
466                    to_char(a.ENBID)      "基站enbid",
467                    a.cell_id             "小区id",
468                    a.lon                  "小区经度",
469                    a.lat                  "小区纬度",
470                    a.vender              "厂家",
471                    0 "业务量",
472                    CONSTRUCTION "承建方"
473     from (select distinct CITY_NAME,
474                    DISTRICT_NAME,
475                    CITY_CODE,
476                    DISTRICT_CODE,
477                    PHYSTATION_ADDRESS,
478                    BBU_NAME,
479                    ENBID,
480                    cell_id,
481                    lon,
482                    lat,
483                    vender,
484                    construction
485     from seq_4g_siteinfo
486     where sdate = (select max(sdate) from seq_4g_siteinfo)
487     and is_alive = 1) a,
488     (select distinct city_name, enbid, cell_id
489     from STATS_4g_CELL_FLOW_48h_hour
490     where sdate =
491     (select max(sdate) from STATS_4g_CELL_FLOW_48h_hour)) b
492     where /*a.city_name= b.city_name and*/
493     a.enbid = b.enbid
494     and a.cell_id = b.cell_id
495
496 # 15 48小时无业务5g基站数
497     select distinct a.city_name           "地市名称",
498                    a.DISTRICT_NAME       "行政区名称",
499                    a.CITY_CODE           "地市编码",
500                    a.DISTRICT_CODE       "行政区编码",
501                    a.PHYSTATION_ADDRESS  "物理站名称",
502                    a.STATION_NAME        "基站名称",
503                    to_char(a.gnbid)      "基站GNBID",
504                    a.vender              "厂家",
505                    0 "业务量",
506                    CONSTRUCTION "承建方"
507     from (select distinct city_name,
508                    DISTRICT_NAME,
509                    CITY_CODE,
510                    DISTRICT_CODE,
511                    PHYSTATION_ADDRESS,
512                    STATION_NAME,
513                    gnbid,
514                    vender,

```



```

515         first_value(construction) over(partition by
district_name, district_code, gnbid) construction
516     from seq_5g_siteinfo
517     where sdate = (select max(sdate) from seq_5g_siteinfo)
518         and is_alive = 1) a,
519     (select distinct city_name, PHYSTATION_ADDRESS, GNBID
520     from STATS_5g_phySITE_FLOW_48h_hour
521     where sdate =
522         (select max(sdate) from STATS_5g_phySITE_FLOW_48h_hour))
b
523     where /*a.city_name= b.city_name and*/
524     a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
525     and a.GNBID = b.GNBID
526
527 # 16 48小时无业务5g小区数
528     select distinct a.city_name           "地市名称",
529                    a.DISTRICT_NAME       "行政区名称",
530                    a.CITY_CODE           "地市编码",
531                    a.DISTRICT_CODE       "行政区编码",
532                    a.PHYSTATION_ADDRESS  "物理站名称",
533                    a.STATION_NAME        "基站名称",
534                    to_char(a.gnbid)      "基站GNBID",
535                    a.cell_name           "小区名称",
536                    a.cell_id             "小区id",
537                    a.lon                 "小区经度",
538                    a.lat                 "小区纬度",
539                    a.vender              "厂家",
540                    0 "业务量",
541                    CONSTRUCTION "承建方"
542     from (select distinct city_name,
543                    DISTRICT_NAME,
544                    CITY_CODE,
545                    DISTRICT_CODE,
546                    PHYSTATION_ADDRESS,
547                    STATION_NAME,
548                    gnbid,
549                    cell_name,
550                    cell_id,
551                    lon,
552                    lat,
553                    vender,
554                    construction
555     from seq_5g_siteinfo
556     where sdate = (select max(sdate) from seq_5g_siteinfo)
557         and is_alive = 1) a,
558     (select distinct city_name, GNBID, cell_id
559     from STATS_5g_CELL_FLOW_48h_hour
560     where sdate =
561         (select max(sdate) from STATS_5g_CELL_FLOW_48h_hour)) b
562     where /*a.city_name = b.city_name
563     and*/ a.GNBID = b.GNBID
564     and a.cell_id = b.cell_id
565
566 # 17 4g小区错误或字段不全
567     select distinct ERROR_CLASS           "异常类型",
568                    CITY_NAME             "城市名称",
569                    DISTRICT_NAME         "行政区名称",
570                    CITY_CODE             "城市编码",

```

```

571 DISTRICT_CODE "行政区编码",
572 NETWORK_NAME "网络类型",
573 PHYSTATION_ADDRESS "物理站名称",
574 BBU_NAME "BBU名称",
575 to_char(ENBID) "基站ENBID",
576 CELL_NAME "小区名称",
577 CELL_ID "小区id",
578 LON "小区经度",
579 LAT "小区纬度",
580 DIRECTION "小区方位角",
581 HEIGHT "天线挂高",
582 M_DOWNTILT "机械下倾",
583 E_DOWNTILT "电子下倾",
584 STATION_TYPE "基站类型",
585 ISDIGITALINDOOR "是否电子室分",
586 DOWN_FREQ "下行频点",
587 VENDER "厂家",
588 OWN_SCHOOLYARD "是否校园站",
589 TOWERADDRESS_CODE "铁塔站址编号",
590 PROPERTY "产权归属",
591 SCENE "场景",
592 IS_SCENESITE "是否场景站",
593 MARKETING_NETWORK "销售网络",
594 TERMINALAMOUNT_5G "终端数量5G",
595 SECTOR_INCOMING "扇区收入",
596 IS_BUSY "是否超忙",
597 CONSTRUCTION "承建方",
598 IS_SAME_ADDRESS "是否共站址",
599 SAME_ADDRESS_SITES "对应共站址基站ID",
600 IS_SAME_ADDRESS_CT "共站址运营商"
601 from STATS_4G_ERROR_SITEINFO
602 where sdate = (select max(sdate) from STATS_4G_ERROR_SITEINFO)
603 and is_alive = 1
604

```

18 5g小区错误或字段不全

```

605 select distinct ERROR_CLASS "异常类型",
606 CITY_NAME "城市名称",
607 DISTRICT_NAME "行政区名称",
608 CITY_CODE "城市编码",
609 DISTRICT_CODE "行政区编码",
610 NETWORK_NAME "网络类型",
611 PHYSTATION_ADDRESS "物理站名称",
612 STATION_NAME "基站名称",
613 to_char(GNBID) "基站GNBID",
614 CELL_NAME "小区名称",
615 CELL_ID "小区id",
616 LON "小区经度",
617 LAT "小区纬度",
618 DIRECTION "方位角",
619 HEIGHT "天线挂高",
620 M_DOWNTILT "机械下倾",
621 E_DOWNTILT "电子下倾",
622 STATION_TYPE "基站类型",
623 ISDIGITALINDOOR "是否数字化室分",
624 DOWN_FREQ "下行频点",
625 VENDER "厂家",
626 OWN_SCHOOLYARD "是否校园站",
627 TOWERADDRESS_CODE "铁塔地址编码",
628

```

```

629         PROPERTY "产权归属",
630         SCENE "场景",
631         IS_SCENESITE "是否场景站",
632         MARKETING_NETWORK "销售网格",
633         TERMINALAMOUNT_5G "终端数量5G",
634         SECTOR_INCOMING "扇区收入",
635         IS_BUSY "是否超忙",
636         CONSTRUCTION "承建方",
637         IS_SAME_ADDRESS "是否共站址",
638         SAME_ADDRESS_SITES "对应共站址基站ID",
639         IS_SAME_ADDRESS_CT "共站址运营商"
640     from STATS_5G_ERROR_SITEINFO
641     where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
642         and is_alive = 1
643
644 # 19 4g经纬度偏移>3000小区数
645     select CITY_NAME "城市名称",
646            DISTRICT_NAME "行政区名称",
647            CITY_CODE "城市编码",
648            DISTRICT_CODE "行政区编码",
649            PHYSTATION_ADDRESS "物理站址",
650            BBU_NAME "BBU名称",
651            to_char(ENBID) "基站enbid",
652            STATION_TYPE "基站类型",
653            VENDER "设备厂家",
654            SCENE "覆盖厂家",
655            DEVIATE_DISTANCE "基站偏离距离",
656            C1_ECI "小区1_eci",
657            C1_CELL_NAME "小区1_小区名称",
658            C1_DISTANCE "小区1_偏离距离",
659            C2_ECI "小区2_eci",
660            C2_CELL_NAME "小区2_小区名称",
661            C2_DISTANCE "小区2_偏离距离",
662            C3_ECI "小区3_eci",
663            C3_CELL_NAME "小区3_小区名称",
664            C3_DISTANCE "小区3_偏离距离",
665            C4_ECI "小区4_eci",
666            C4_CELL_NAME "小区4_小区名称",
667            C4_DISTANCE "小区4_偏离距离",
668            C5_ECI "小区5_eci",
669            C5_CELL_NAME "小区5_小区名称",
670            C5_DISTANCE "小区5_偏离距离",
671            C6_ECI "小区6_eci",
672            C6_CELL_NAME "小区6_小区名称",
673            C6_DISTANCE "小区6_偏离距离",
674            C7_ECI "小区7_eci",
675            C7_CELL_NAME "小区7_小区名称",
676            C7_DISTANCE "小区7_偏离距离",
677            C8_ECI "小区8_eci",
678            C8_CELL_NAME "小区8_小区名称",
679            C8_DISTANCE "小区8_偏离距离",
680            C9_ECI "小区9_eci",
681            C9_CELL_NAME "小区9_小区名称",
682            C9_DISTANCE "小区9_偏离距离",
683            CONSTRUCTION "承建方"
684     from stats_4G_DEVIATE_SUMMARY_DB
685     where DEVIATE_DISTANCE>3000
686

```

```

687 # 20 导出4g工参数数据
688     SELECT SDATE "数据日期",
689           CITY_CODE "城市编码",
690           CITY_NAME "城市名称",
691           DISTRICT_CODE "行政区编码",
692           DISTRICT_NAME "行政区名称",
693           NETWORK_NAME "网络类型",
694           PHYSTATION_ADDRESS "物理站名称",
695           BBU_NAME "BBU名称",
696           to_char(ENBID) "基站ENBID",
697           CELL_NAME "小区名称",
698           CELL_ID "小区id",
699           to_char('46001' || enbid * 256 + cell_id) "小区全球唯一标识",
700           LON "小区经度",
701           LAT "小区纬度",
702           DIRECTION "小区方位角",
703           HEIGHT "天线挂高",
704           M_DOWNTILT "机械下倾",
705           E_DOWNTILT "电子下倾",
706           STATION_TYPE "基站类型",
707           ISDIGITALINDOOR "是否电子室分",
708           DOWN_FREQ "下行频点",
709           VENDER "厂家",
710           OWN_SCHOOLYARD "是否校园站",
711           TOWERADDRESS_CODE "铁塔站址编号",
712           PROPERTY "产权归属",
713           SCENE "场景",
714           IS_SCENESITE "是否场景站",
715           MARKETING_NETWORK "销售网络",
716           TERMINALAMOUNT_5G "终端数量5G",
717           SECTOR_INCOMING "扇区收入",
718           IS_BUSY "是否超忙",
719           IS_ALIVE "是否有业务",
720           IS_ALIVE_UPDATE_TIME "业务更新时间",
721           CONSTRUCTION "承建方",
722           IS_SAME_ADDRESS "是否共站址",
723           SAME_ADDRESS_SITES "对应共站址基站ID",
724           IS_SAME_ADDRESS_CT "共站址运营商"
725     FROM seq_4g_siteinfo
726     where sdate = (select max(sdate) from seq_4g_siteinfo)
727           and is_alive = 1
728
729
730 # 21 导出5g工参数数据
731     SELECT SDATE "数据日期",
732           CITY_CODE "城市编码",
733           CITY_NAME "城市名称",
734           DISTRICT_CODE "行政区编码",
735           DISTRICT_NAME "行政区名称",
736           NETWORK_NAME "网络类型",
737           PHYSTATION_ADDRESS "物理站名称",
738           STATION_NAME "基站名称",
739           to_char(GNBID) "基站GNBID",
740           CELL_NAME "小区名称",
741           CELL_ID "小区id",
742           to_char('46001' || gnbid * 4096 + cell_id) "小区全球唯一识别码",
743           LON "小区经度",
744           LAT "小区纬度",

```

```

745     DIRECTION "方位角",
746     HEIGHT "天线挂高",
747     M_DOWNTILT "机械下倾",
748     E_DOWNTILT "电子下倾",
749     STATION_TYPE "基站类型",
750     ISDIGITALINDOOR "是否数字化室分",
751     DOWN_FREQ "下行频点",
752     VENDER "厂家",
753     OWN_SCHOOLYARD "是否校园站",
754     TOWERADDRESS_CODE "铁塔地址编码",
755     PROPERTY "产权归属",
756     SCENE "场景",
757     IS_SCENESITE "是否场景站",
758     MARKETING_NETWORK "销售网格",
759     TERMINALAMOUNT_5G "终端数量5G",
760     SECTOR_INCOMING "扇区收入",
761     IS_BUSY "是否超忙",
762     IS_ALIVE "是否有业务",
763     IS_ALIVE_UPDATE_TIME "业务更新时间",
764     CONSTRUCTION "承建方",
765     IS_SAME_ADDRESS "是否共站址",
766     SAME_ADDRESS_SITES "对应共站址基站ID",
767     IS_SAME_ADDRESS_CT "共站址运营商"
768 FROM seq_5g_siteinfo
769 where sdate = (select max(sdate) from seq_5g_siteinfo)
770 and is_alive = 1

```

6.3 接口程序及源码

```

1 # 开发语言：接口对接程序采用go语言编写；
2 # 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。
3
4 # 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。
5
6 # 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。
7
8 # 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。

```

```

1 package main
2
3 import (
4     "database/sql"
5     "encoding/csv"
6     "fmt"
7     "io"
8     "log"
9     "os"
10    "path"
11    "time"
12

```

```

13     "github.com/jlaffaye/ftp"
14     _ "github.com/mattn/go-oci8"
15 )
16
17 var logger *log.Logger
18 var file *os.File
19 var err error
20
21 func init() {
22     file, err = os.OpenFile("test.log",
23 os.O_APPEND|os.O_CREATE|os.O_WRONLY, 666)
24     if err != nil {
25         log.Fatal(err)
26     }
27
28     logger = log.New(file, "", log.LstdFlags)
29     logger.SetPrefix("Test- ") // 设置日志前缀
30     logger.SetFlags(log.Ldate | log.Ltime | log.Lshortfile)
31     /*
32         const (
33             // 字段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。
34             // 在所有项目后会有一个冒号: 2009/01/23 01:23:23.123123
35 /a/b/c/d.go:23: message
36             Ldate          = 1 << iota    // 日期: 2009/01/23
37             Ltime          // 时间: 01:23:23
38             Lmicroseconds // 微秒分辨率: 01:23:23.123123 (用
39 于增强Ltime位)
40             Llongfile      // 文件全路径名+行号:
41 /a/b/c/d.go:23
42             Lshortfile    // 文件无路径名+行号: d.go:23 (会覆
43 盖掉Llongfile)
44             LstdFlags     = Ldate | Ltime // 标准logger的初始值
45         )
46     */
47 }
48
49 func sqlExec(db *sql.DB, sqlStmt string) error {
50     res, err := db.Exec(sqlStmt)
51     if err != nil {
52         logger.Fatal("sqlExec执行失败:" + err.Error())
53     }
54
55     num, err := res.RowsAffected()
56     if err != nil {
57         logger.Fatal("获取查询影响行数失败:" + err.Error())
58     }
59
60     logger.Printf("SQL Execute success rows affected %d\n", num)
61     return nil
62 }
63
64 func sqlQuery2Csv(db *sql.DB, sqlStmt string, newFileName string) error {
65     rows, err := db.Query(sqlStmt)
66     if err != nil {
67         logger.Fatal("sqlQuery执行失败,err: %s,err sql:%s", err, sqlStmt)
68     }
69     defer rows.Close()
70     //返回所有列

```

```

66     cols, _ := rows.Columns()
67     //这里表示一行所有列的值, 用[]byte表示
68     vals := make([][]byte, len(cols))
69     //这里表示一行填充数据
70     scans := make([]interface{}, len(cols))
71     //这里scans引用vals, 把数据填充到[]byte里
72     for k, _ := range vals {
73         scans[k] = &vals[k]
74     }
75     // 统计行数
76     i := 0
77     //result := make(map[int]map[string]string)
78     //这样可以追加写
79     nfs, err := os.OpenFile(newFileName, os.O_RDWR|os.O_CREATE|os.O_TRUNC,
0666)
80     if err != nil {
81         logger.Fatalf("can not create file, err is %v", err)
82     }
83     defer nfs.Close()
84
85     nfs.Seek(0, io.SeekEnd)
86
87     w := csv.NewWriter(nfs)
88
89     //设置属性
90     w.Comma = ','
91     w.UseCRLF = true
92     //row := []string{"1", "2", "3", "4", "5,6"}
93     // 写入文件
94     err = w.Write(cols)
95
96     for rows.Next() {
97         //填充数据
98         rows.Scan(scans...)
99         //每行数据
100        var row []string
101        //把vals中的数据复制到row中
102        for _, v := range vals {
103            // key := cols[k]
104            //这里把[]byte数据转成string
105            row = append(row, string(v))
106        }
107        //放入结果集
108        //result[i] = row;
109        err = w.Write(row)
110        i++
111    }
112    w.Flush()
113
114    if err != nil {
115        logger.Fatalf("写入csv失败", err)
116    }
117
118    // logger.Printf("SQL Query success rows queried %d\n", i)
119    logger.Printf("查询结果导出csv成功: "+newFileName+", 行数: %d\n", i)
120    return nil
121 }
122

```

```

123 //判断文件或文件夹是否存在
124 func isExist(path string) bool {
125     _, err := os.Stat(path)
126     if err != nil {
127         if os.IsExist(err) {
128             return true
129         }
130         if os.IsNotExist(err) {
131             return false
132         }
133         return false
134     }
135     return true
136 }
137
138 func main() {
139     // 20210520 修改文件夹及文件名日期为月粒度
140     dateDir := time.Now().Format("20060102")
141     // dateDir := time.Now().Format("200601")
142     fileDir := "./" + dateDir
143     if !isExist(fileDir) {
144         os.Mkdir(fileDir, 0766)
145     }
146     db, err := sql.Open("oci8", fmt.Sprintf("%s/%s@s", "c##fast491",
147 "F@st491*321", "192.168.0.64:1521/fast"))
148     if err != nil {
149         logger.Fatalf("oracle登录失败:" + err.Error())
150     }
151     defer db.Close()
152     logger.Printf("数据库已登录")
153
154     err = db.Ping()
155     if err != nil {
156         logger.Fatalf("oracle链接不可达:" + err.Error())
157     }
158
159     waitFiles := make([]string, 0, 5)
160
161     logger.Printf("查询结果导出csv")
162
163     // 导出文件-0 大屏展示数据
164     fileName := "gcToyxgx_00_bigscreen_" + dateDir + ".csv"
165     filepath := fileDir + "/" + fileName
166     exportsql := `
167     select DATA_TYPE      "业务名称",
168            district_name  "行政区名称",
169            district_code  "行政区编码",
170            site_num       "小区数量",
171            DATA_NUM      "单位数量",
172            rate           "百分占比",
173            CONSTRUCTION   "承建方"
174     from v_STATS_BIGSCREEN_SUMMARY_32
175 `
176     // sql末尾不能有分号
177     err = sqlQuery2Csv(db, exportsql, filepath)
178     if err != nil {
179         logger.Fatalf("从数据库导出数据到csv失败:" + err.Error())
180     } else {

```



```

180     if exist(filepath) {
181         waitFiles = append(waitFiles, filepath)
182     }
183 }
184
185 // 导出文件-1 7天有业务的4g基站数
186 fileName = "gcToyxgx_01_4g_site_youyewu_7d_" + dateDir + ".csv"
187 filepath = fileDir + "/" + fileName
188 exportSql = `
189 select a.CITY_NAME "地市名称",
190        a.DISTRICT_NAME "行政区名称",
191        a.CITY_CODE "地市编码",
192        a.DISTRICT_CODE "行政区编码",
193        a.PHYSTATION_ADDRESS "物理站名称",
194        a.BBU_NAME "BBU名称",
195        to_char(a.ENBID) "基站enbid",
196        a.vender "厂家",
197        round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte, 需要除以1024*1024*1024
198        ,
199        CONSTRUCTION "承建方"
200 from (select distinct CITY_NAME,
201                    DISTRICT_NAME,
202                    CITY_CODE,
203                    DISTRICT_CODE,
204                    PHYSTATION_ADDRESS,
205                    BBU_NAME,
206                    ENBID,
207                    vender,
208                    first_value(construction) over(partition by
district_name, district_code, enbid) construction
209        from seq_4g_siteinfo
210        where sdate = (select max(sdate) from seq_4g_siteinfo)
211        and is_alive = 1) a,
212 (select distinct city_name, enbid, THROUGHPUT
213    from STATS_4g_SITE_FLOW_2M_DAY
214    where sdate =
215        (select max(sdate) from STATS_4g_SITE_FLOW_2M_DAY)) b
216 where /*a.city_name=b.city_name and*/
217 a.enbid = b.enbid
218 `
219 err = sqlQuery2Csv(db, exportSql, filepath)
220 if err != nil {
221     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
222 } else {
223     if exist(filepath) {
224         waitFiles = append(waitFiles, filepath)
225     }
226 }
227 // 导出文件-2 7天有业务的4g小区数
228 fileName = "gcToyxgx_02_4g_cell_youyewu_7d_" + dateDir + ".csv"
229 filepath = fileDir + "/" + fileName
230 exportSql = `
231 select distinct a.CITY_NAME "地市名称",
232                a.DISTRICT_NAME "行政区名称",
233                a.CITY_CODE "地市编码",
234                a.DISTRICT_CODE "行政区编码",
235                a.PHYSTATION_ADDRESS "物理站名称",

```

```

236         a.BBU_NAME "BBU名称",
237         to_char(a.ENBID) "基站enbid",
238         a.cell_id "小区id",
239         a.lon "小区经度",
240         a.lat "小区纬度",
241         a.vender "厂家",
242         round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte, 需要除以1024*1024*1024
243     ,
244     CONSTRUCTION "承建方"
245 from (select distinct CITY_NAME,
246         DISTRICT_NAME,
247         CITY_CODE,
248         DISTRICT_CODE,
249         PHYSTATION_ADDRESS,
250         BBU_NAME,
251         ENBID,
252         cell_id,
253         lon,
254         lat,
255         vender,
256         construction
257     from seq_4g_siteinfo
258     where sdate = (select max(sdate) from seq_4g_siteinfo)) a,
259 (select distinct city_name, enbid, cell_id, THROUGHPUT
260     from STATS_4g_CELL_FLOW_2M_DAY
261     where sdate =
262         (select max(sdate) from STATS_4g_CELL_FLOW_2M_DAY)) b
263 where /*a.city_name=b.city_name and*/
264     a.enbid = b.enbid
265     and a.cell_id = b.cell_id
266 `
267 err = sqlQuery2Csv(db, exportsql, filepath)
268 if err != nil {
269     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
270 } else {
271     if isExist(filepath) {
272         waitFiles = append(waitFiles, filepath)
273     }
274 }
275 // 导出文件-3 7天有业务的5g基站数
276 fileName = "gcToyxgx_03_5g_site_youyewu_7d_" + dateDir + ".csv"
277 filepath = fileDir + "/" + fileName
278 exportsql = `
279 select distinct a.city_name "地市名称",
280         a.DISTRICT_NAME "行政区名称",
281         a.CITY_CODE "地市编码",
282         a.DISTRICT_CODE "行政区编码",
283         a.PHYSTATION_ADDRESS "物理站名称",
284         a.STATION_NAME "基站名称",
285         to_char(a.gnbid) "基站GNBID",
286         a.vender "厂家",
287         round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单
单位是Mbyte, 需要除以1024
288     ,
289     CONSTRUCTION "承建方"
290 from (select distinct city_name,
291         DISTRICT_NAME,

```

```

292         CITY_CODE,
293         DISTRICT_CODE,
294         PHYSTATION_ADDRESS,
295         STATION_NAME,
296         gnbid,
297         vender,
298         first_value(construction) over(partition by
district_name, district_code, gnbid) construction
299     from seq_5g_siteinfo
300     where sdate = (select max(sdate) from seq_5g_siteinfo)
301           and is_alive = 1) a,
302     (select distinct city_name, PHYSTATION_ADDRESS, gnbid,
THROUGHPUT
303         from STATS_5g_PHYSITE_FLOW_2M_DAY
304         where sdate =
305             (select max(sdate) from STATS_5g_PHYSITE_FLOW_2M_DAY))
b
306     where /* a.city_name=b.city_name and*/
307           a.gnbid = b.gnbid
308           and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
309     `
310 err = sqlQuery2Csv(db, exportSql, filepath)
311 if err != nil {
312     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
313 } else {
314     if isExist(filepath) {
315         waitFiles = append(waitFiles, filepath)
316     }
317 }
318 // 导出文件-4 7天有业务的5g小区数
319 fileName = "gcToyxgx_04_5g_cell_youyewu_7d_" + dateDir + ".csv"
320 filepath = fileDir + "/" + fileName
321 exportSql = `
322 select distinct a.city_name "地市名称",
323                a.DISTRICT_NAME "行政区名称",
324                a.CITY_CODE "地市编码",
325                a.DISTRICT_CODE "行政区编码",
326                a.PHYSTATION_ADDRESS "物理站名称",
327                a.STATION_NAME "基站名称",
328                to_char(a.gnbid) "基站GNBID",
329                a.cell_name "小区名称",
330                a.cell_id "小区id",
331                a.lon "小区经度",
332                a.lat "小区纬度",
333                a.vender "厂家",
334                round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单
位是Mbyte, 需要除以1024
335            ,
336            CONSTRUCTION "承建方"
337     from (select distinct city_name,
338                DISTRICT_NAME,
339                CITY_CODE,
340                DISTRICT_CODE,
341                PHYSTATION_ADDRESS,
342                STATION_NAME,
343                gnbid,
344                cell_name,
345                cell_id,

```

```

346         lon,
347         lat,
348         vender,
349         construction
350     from seq_5g_siteinfo
351     where sdate = (select max(sdate) from seq_5g_siteinfo)
352         and is_alive = 1) a,
353     (select distinct city_name, cell_id, gnbid, THROUGHPUT
354     from STATS_5g_CELL_FLOW_2M_DAY
355     where sdate =
356         (select max(sdate) from STATS_5g_CELL_FLOW_2M_DAY)) b
357     where /*a.city_name=b.city_name and */
358     a.gnbid = b.gnbid
359     and a.cell_id = b.cell_id
360 `
361 err = sqlQuery2Csv(db, exportSql, filepath)
362 if err != nil {
363     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
364 } else {
365     if isExist(filepath) {
366         waitFiles = append(waitFiles, filepath)
367     }
368 }
369 // 导出文件-5 7天无业务4g基站数
370 fileName = "gcToyxgx_05_4g_site_wuyewu_2d_" + dateDir + ".csv"
371 filepath = fileDir + "/" + fileName
372 exportSql = `
373     select distinct a.CITY_NAME "地市名称",
374         a.DISTRICT_NAME "行政区名称",
375         a.CITY_CODE "地市编码",
376         a.DISTRICT_CODE "行政区编码",
377         a.PHYSTATION_ADDRESS "物理站名称",
378         a.BBU_NAME "BBU名称",
379         to_char(a.ENBID) "基站enbid",
380         a.vender "厂家",
381         0 "业务量",
382         construction "承建方"
383     from (select distinct CITY_NAME,
384         DISTRICT_NAME,
385         CITY_CODE,
386         DISTRICT_CODE,
387         PHYSTATION_ADDRESS,
388         BBU_NAME,
389         ENBID,
390         vender,
391         first_value(construction) over(partition by
district_name, district_code, enbid) construction
392     from seq_4g_siteinfo
393     where sdate = (select max(sdate) from seq_4g_siteinfo)
394         and is_alive = 1) a,
395     (select distinct city_name, enbid
396     from STATS_4g_SITE_FLOW_7d_DAY
397     where sdate =
398         (select max(sdate) from STATS_4g_SITE_FLOW_7d_DAY)) b
399     where /*a.city_name= b.city_name and*/
400     a.enbid = b.enbid
401 `
402 err = sqlQuery2Csv(db, exportSql, filepath)

```

```

403     if err != nil {
404         logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
405     } else {
406         if isExist(filepath) {
407             waitFiles = append(waitFiles, filepath)
408         }
409     }
410     // 导出文件-6 7天无业务4g小区数
411     fileName = "gcToyxgx_06_4g_cell_wuyewu_2d_" + dateDir + ".csv"
412     filepath = fileDir + "/" + fileName
413     exportsql = `
414     select distinct a.CITY_NAME          "地市名称",
415                    a.DISTRICT_NAME     "行政区名称",
416                    a.CITY_CODE         "地市编码",
417                    a.DISTRICT_CODE     "行政区编码",
418                    a.PHYSTATION_ADDRESS "物理站名称",
419                    a.BBU_NAME          "BBU名称",
420                    to_char(a.ENBID)    "基站enbid",
421                    a.cell_id           "小区id",
422                    a.lon               "小区经度",
423                    a.lat               "小区纬度",
424                    a.vender           "厂家",
425                    0 "业务量",
426                    construction "承建方"
427     from (select distinct CITY_NAME,
428                    DISTRICT_NAME,
429                    CITY_CODE,
430                    DISTRICT_CODE,
431                    PHYSTATION_ADDRESS,
432                    BBU_NAME,
433                    ENBID,
434                    cell_id,
435                    lon,
436                    lat,
437                    vender,
438                    construction
439            from seq_4g_siteinfo
440            where sdate = (select max(sdate) from seq_4g_siteinfo
441                    and is_alive = 1) a,
442            (select distinct city_name, enbid, cell_id
443            from STATS_4g_CELL_FLOW_7d_DAY
444            where sdate =
445                    (select max(sdate) from STATS_4g_CELL_FLOW_7d_DAY)) b
446     where /*a.city_name= b.city_name and*/
447     a.enbid = b.enbid
448     and a.cell_id = b.cell_id
449     `
450     err = sqlQuery2Csv(db, exportsql, filepath)
451     if err != nil {
452         logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
453     } else {
454         if isExist(filepath) {
455             waitFiles = append(waitFiles, filepath)
456         }
457     }
458     // 导出文件-7 7天低业务4g基站数
459     fileName = "gcToyxgx_07_4g_site_diyewu_2d_" + dateDir + ".csv"
460     filepath = fileDir + "/" + fileName

```

```

461     exportSql = `
462         select distinct a.CITY_NAME "地市名称",
463                        a.DISTRICT_NAME "行政区名称",
464                        a.CITY_CODE "地市编码",
465                        a.DISTRICT_CODE "行政区编码",
466                        a.PHYSTATION_ADDRESS "物理站名称",
467                        a.BBU_NAME "BBU名称",
468                        to_char(a.ENBID) "基站enbid",
469                        a.vender "厂家",
470                        round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte, 需要除以1024*1024*1024
471
472                        ,
473                        CONSTRUCTION "承建方"
474         from (select distinct CITY_NAME,
475                        DISTRICT_NAME,
476                        CITY_CODE,
477                        DISTRICT_CODE,
478                        PHYSTATION_ADDRESS,
479                        BBU_NAME,
480                        ENBID,
481                        vender,
482                        first_value(construction) over(partition by
district_name, district_code, enbid) construction
483         from seq_4g_siteinfo
484         where sdate = (select max(sdate) from seq_4g_siteinfo)
485         and is_alive = 1) a,
486         (select distinct city_name, enbid, THROUGHPUT
487         from STATS_4g_SITE_lowFLOW_7d_DAY
488         where sdate =
(select max(sdate) from STATS_4g_SITE_lowFLOW_7d_DAY))
b
489     where /*a.city_name= b.city_name and*/
490     a.enbid = b.enbid
491 `
492     err = sqlQuery2Csv(db, exportSql, filepath)
493     if err != nil {
494         logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
495     } else {
496         if isExist(filepath) {
497             waitFiles = append(waitFiles, filepath)
498         }
499     }
500
501     // 导出文件-8 7天低业务4g小区数
502     fileName = "gcToyxgx_08_4g_cell_diyewu_2d_" + dateDir + ".csv"
503     filepath = fileDir + "/" + fileName
504     exportSql = `
505         select distinct a.CITY_NAME "地市名称",
506                        a.DISTRICT_NAME "行政区名称",
507                        a.CITY_CODE "地市编码",
508                        a.DISTRICT_CODE "行政区编码",
509                        a.PHYSTATION_ADDRESS "物理站名称",
510                        a.BBU_NAME "BBU名称",
511                        to_char(a.ENBID) "基站enbid",
512                        a.cell_id "小区id",
513                        a.lon "小区经度",
514                        a.lat "小区纬度",
515                        a.vender "厂家",

```

```

516         round(b.THROUGHPUT / 1024 / 1024 / 1024, 2) "业务量" --
THROUGHPUT 单位是byte, 需要除以1024*1024*1024
517     ,
518     CONSTRUCTION "承建方"
519     from (select distinct CITY_NAME,
520             DISTRICT_NAME,
521             CITY_CODE,
522             DISTRICT_CODE,
523             PHYSTATION_ADDRESS,
524             BBU_NAME,
525             ENBID,
526             cell_id,
527             lon,
528             lat,
529             vender,
530             construction
531     from seq_4g_siteinfo
532     where sdate = (select max(sdate) from seq_4g_siteinfo)
533     and is_alive = 1) a,
534     (select distinct city_name, enbid, cell_id, THROUGHPUT
535     from STATS_4g_CELL_lowFLOW_7d_DAY
536     where sdate =
537     (select max(sdate) from STATS_4g_CELL_lowFLOW_7d_DAY))
b
538     where /*a.city_name= b.city_name and*/
539     a.enbid = b.enbid
540     and a.cell_id = b.cell_id
541     `
542     err = sqlQuery2Csv(db, exportSql, filepath)
543     if err != nil {
544         logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
545     } else {
546         if isExist(filepath) {
547             waitFiles = append(waitFiles, filepath)
548         }
549     }
550
551     // 导出文件-9 7天无业务5g基站数
552     fileName = "gcToyxgx_09_5g_site_wuyewu_2d_" + dateDir + ".csv"
553     filepath = fileDir + "/" + fileName
554     exportSql = `
555     select distinct a.city_name           "地市名称",
556                    a.DISTRICT_NAME      "行政区名称",
557                    a.CITY_CODE          "地市编码",
558                    a.DISTRICT_CODE      "行政区编码",
559                    a.PHYSTATION_ADDRESS "物理站名称",
560                    a.STATION_NAME       "基站名称",
561                    to_char(a.gnbid)     "基站GNBID",
562                    a.vender             "厂家",
563                    0 "业务量",
564                    CONSTRUCTION "承建方"
565     from (select distinct city_name,
566             DISTRICT_NAME,
567             CITY_CODE,
568             DISTRICT_CODE,
569             PHYSTATION_ADDRESS,
570             STATION_NAME,
571             gnbid,

```

```

572         vender,
573         first_value(construction) over(partition by
district_name, district_code, gnbid) construction
574     from seq_5g_siteinfo
575     where sdate = (select max(sdate) from seq_5g_siteinfo)
576         and is_alive = 1) a,
577     (select distinct city_name, PHYSTATION_ADDRESS, GNBID
578     from STATS_5g_phySITE_FLOW_7d_DAY
579     where sdate =
580         (select max(sdate) from STATS_5g_phySITE_FLOW_7d_DAY))
b
581     where /*a.city_name= b.city_name and*/
582     a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
583     and a.GNBID = b.GNBID
584     `
585     err = sqlQuery2Csv(db, exportSql, filepath)
586     if err != nil {
587         logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
588     } else {
589         if isExist(filepath) {
590             waitFiles = append(waitFiles, filepath)
591         }
592     }
593
594     // 导出文件-10 7天无业务5g小区数
595     fileName = "gcToyxgx_10_5g_cell_wuyewu_2d_" + dateDir + ".csv"
596     filepath = fileDir + "/" + fileName
597     exportSql = `
598     select distinct a.city_name           "地市名称",
599                 a.DISTRICT_NAME         "行政区名称",
600                 a.CITY_CODE             "地市编码",
601                 a.DISTRICT_CODE         "行政区编码",
602                 a.PHYSTATION_ADDRESS    "物理站名称",
603                 a.STATION_NAME          "基站名称",
604                 to_char(a.gnbid)        "基站GNBID",
605                 a.cell_name             "小区名称",
606                 a.cell_id               "小区id",
607                 a.lon                   "小区经度",
608                 a.lat                   "小区纬度",
609                 a.vender                "厂家",
610                 0 "业务量",
611                 CONSTRUCTION "承建方"
612     from (select distinct city_name,
613             DISTRICT_NAME,
614             CITY_CODE,
615             DISTRICT_CODE,
616             PHYSTATION_ADDRESS,
617             STATION_NAME,
618             gnbid,
619             cell_name,
620             cell_id,
621             lon,
622             lat,
623             vender,
624             construction
625     from seq_5g_siteinfo
626     where sdate = (select max(sdate) from seq_5g_siteinfo)
627         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
658                    位是Mbyte, 需要除以1024,
659                    ,
660                    CONSTRUCTION "承建方"
661     from (select distinct city_name,
662                    DISTRICT_NAME,
663                    CITY_CODE,
664                    DISTRICT_CODE,
665                    PHYSTATION_ADDRESS,
666                    STATION_NAME,
667                    gnbid,
668                    vender,
669                    first_value(construction) over(partition by
670                    district_name, district_code, gnbid) construction
671     from seq_5g_siteinfo
672     where sdate = (select max(sdate) from seq_5g_siteinfo)
673     and is_alive = 1) a,
674     (select distinct city_name, PHYSTATION_ADDRESS, GNBID,
675     THROUGHPUT
676     from STATS_5g_PHYSITE_LOFLOW_7d_DAY
677     where sdate =
678         (select max(sdate) from
679     STATS_5g_PHYSITE_LOFLOW_7d_DAY)) b
680     where /*a.city_name= b.city_name and*/
681     a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
682     and a.GNBID = b.GNBID
683     `
684 err = sqlQuery2Csv(db, exportSql, filepath)
685 if err != nil {

```

```

682     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
683 } else {
684     if isExist(filepath) {
685         waitFiles = append(waitFiles, filepath)
686     }
687 }
688
689 // 导出文件-12 7天低业务5g小区数
690 fileName = "gcToyxgx_12_5g_cell_diyewu_2d_" + dateDir + ".csv"
691 filepath = fileDir + "/" + fileName
692 exportSql = `
693 select distinct a.city_name "地市名称",
694                a.DISTRICT_NAME "行政区名称",
695                a.CITY_CODE "地市编码",
696                a.DISTRICT_CODE "行政区编码",
697                a.PHYSTATION_ADDRESS "物理站名称",
698                a.STATION_NAME "基站名称",
699                to_char(a.gnbid) "基站GNBID",
700                a.cell_name "小区名称",
701                a.cell_id "小区id",
702                a.lon "小区经度",
703                a.lat "小区纬度",
704                a.vender "厂家",
705                round(b.THROUGHPUT / 1024, 2) "业务量" --THROUGHPUT 单
是Mbyte, 需要除以1024
706                ,
707                CONSTRUCTION "承建方"
708 from (select distinct city_name,
709                    DISTRICT_NAME,
710                    CITY_CODE,
711                    DISTRICT_CODE,
712                    PHYSTATION_ADDRESS,
713                    STATION_NAME,
714                    gnbid,
715                    cell_name,
716                    cell_id,
717                    lon,
718                    lat,
719                    vender,
720                    construction
721         from seq_5g_siteinfo
722        where sdate = (select max(sdate) from seq_5g_siteinfo)
723              and is_alive = 1) a,
724 (select distinct city_name, GNBID, cell_id, THROUGHPUT
725  from STATS_5g_CELL_LOWFLOW_7d_DAY
726  where sdate =
727         (select max(sdate) from STATS_5g_CELL_LOWFLOW_7d_DAY))
b
728 where /*a.city_name= b.city_name and*/
729 a.GNBID = b.GNBID
730 and a.cell_id = b.cell_id
731 `
732 err = sqlQuery2Csv(db, exportSql, filepath)
733 if err != nil {
734     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
735 } else {
736     if isExist(filepath) {
737         waitFiles = append(waitFiles, filepath)

```

```

738     }
739 }
740
741 // 导出文件-13 48小时无业务4g基站数
742 fileName = "gcToyxgx_13_4g_site_wuyewu_1d_" + dateDir + ".csv"
743 filepath = fileDir + "/" + fileName
744 exportSql = `
745 select distinct a.CITY_NAME           "地市名称",
746                a.DISTRICT_NAME       "行政区名称",
747                a.CITY_CODE           "地市编码",
748                a.DISTRICT_CODE       "行政区编码",
749                a.PHYSTATION_ADDRESS  "物理站名称",
750                a.BBU_NAME            "BBU名称",
751                to_char(a.ENBID)       "基站enbid",
752                a.vender              "厂家",
753                0 "业务量",
754                CONSTRUCTION "承建方"
755 from (select distinct CITY_NAME,
756                DISTRICT_NAME,
757                CITY_CODE,
758                DISTRICT_CODE,
759                PHYSTATION_ADDRESS,
760                BBU_NAME,
761                ENBID,
762                vender,
763                first_value(construction) over(partition by
district_name, district_code, enbid) construction
764 from seq_4g_siteinfo
765 where sdate = (select max(sdate) from seq_4g_siteinfo)
766 and is_alive = 1) a,
767 (select distinct city_name, enbid
768 from STATS_4g_SITE_FLOW_48h_hour
769 where sdate =
770 (select max(sdate) from STATS_4g_SITE_FLOW_48h_hour)) b
771 where /*a.city_name= b.city_name and*/
772 a.enbid = b.enbid
773 `
774 err = sqlQuery2Csv(db, exportSql, filepath)
775 if err != nil {
776     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
777 } else {
778     if isExist(filepath) {
779         waitFiles = append(waitFiles, filepath)
780     }
781 }
782
783 // 导出文件-14 48小时无业务4g小区数
784 fileName = "gcToyxgx_14_4g_cell_wuyewu_1d_" + dateDir + ".csv"
785 filepath = fileDir + "/" + fileName
786 exportSql = `
787 select distinct a.CITY_NAME           "地市名称",
788                a.DISTRICT_NAME       "行政区名称",
789                a.CITY_CODE           "地市编码",
790                a.DISTRICT_CODE       "行政区编码",
791                a.PHYSTATION_ADDRESS  "物理站名称",
792                a.BBU_NAME            "BBU名称",
793                to_char(a.ENBID)       "基站enbid",
794                a.cell_id             "小区id",

```

```

795         a.lon           "小区经度",
796         a.lat           "小区纬度",
797         a.vender        "厂家",
798         0 "业务量",
799         CONSTRUCTION "承建方"
800     from (select distinct CITY_NAME,
801             DISTRICT_NAME,
802             CITY_CODE,
803             DISTRICT_CODE,
804             PHYSTATION_ADDRESS,
805             BBU_NAME,
806             ENBID,
807             cell_id,
808             lon,
809             lat,
810             vender,
811             construction
812     from seq_4g_siteinfo
813     where sdate = (select max(sdate) from seq_4g_siteinfo)
814     and is_alive = 1) a,
815     (select distinct city_name, enbid, cell_id
816     from STATS_4g_CELL_FLOW_48h_hour
817     where sdate =
818     (select max(sdate) from STATS_4g_CELL_FLOW_48h_hour)) b
819 where /*a.city_name= b.city_name and*/
820 a.enbid = b.enbid
821 and a.cell_id = b.cell_id
822 `
823 err = sqlQuery2Csv(db, exportSql, filepath)
824 if err != nil {
825     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
826 } else {
827     if isExist(filepath) {
828         waitFiles = append(waitFiles, filepath)
829     }
830 }
831
832 // 导出文件-15 48小时无业务5g基站数
833 fileName = "gcToyxgx_15_5g_site_wuyewu_1d_" + dateDir + ".csv"
834 filepath = fileDir + "/" + fileName
835 exportSql = `
836 select distinct a.city_name           "地市名称",
837                 a.DISTRICT_NAME      "行政区名称",
838                 a.CITY_CODE          "地市编码",
839                 a.DISTRICT_CODE      "行政区编码",
840                 a.PHYSTATION_ADDRESS "物理站名称",
841                 a.STATION_NAME       "基站名称",
842                 to_char(a.gnbid)     "基站GNBID",
843                 a.vender              "厂家",
844                 0 "业务量",
845                 CONSTRUCTION "承建方"
846     from (select distinct city_name,
847             DISTRICT_NAME,
848             CITY_CODE,
849             DISTRICT_CODE,
850             PHYSTATION_ADDRESS,
851             STATION_NAME,
852             gnbid,

```

```

853         vender,
854         first_value(construction) over(partition by
district_name, district_code, gnbid) construction
855     from seq_5g_siteinfo
856     where sdate = (select max(sdate) from seq_5g_siteinfo)
857         and is_alive = 1) a,
858     (select distinct city_name, PHYSTATION_ADDRESS, GNBID
859     from STATS_5g_physITE_FLOW_48h_hour
860     where sdate =
861         (select max(sdate) from
STATS_5g_physITE_FLOW_48h_hour)) b
862     where /*a.city_name= b.city_name and*/
863     a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
864     and a.GNBID = b.GNBID
865     `
866     err = sqlQuery2Csv(db, exportSql, filepath)
867     if err != nil {
868         logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
869     } else {
870         if isExist(filepath) {
871             waitFiles = append(waitFiles, filepath)
872         }
873     }
874
875     // 导出文件-16 48小时无业务5g小区数
876     fileName = "gcToyxgx_16_5g_cell_wuyewu_1d_" + dateDir + ".csv"
877     filepath = fileDir + "/" + fileName
878     exportsql = `
879     select distinct a.city_name           "地市名称",
880                   a.DISTRICT_NAME       "行政区名称",
881                   a.CITY_CODE           "地市编码",
882                   a.DISTRICT_CODE       "行政区编码",
883                   a.PHYSTATION_ADDRESS  "物理站名称",
884                   a.STATION_NAME        "基站名称",
885                   to_char(a.gnbid)      "基站GNBID",
886                   a.cell_name           "小区名称",
887                   a.cell_id             "小区id",
888                   a.lon                 "小区经度",
889                   a.lat                 "小区纬度",
890                   a.vender              "厂家",
891                   0 "业务量",
892                   CONSTRUCTION "承建方"
893     from (select distinct city_name,
894                   DISTRICT_NAME,
895                   CITY_CODE,
896                   DISTRICT_CODE,
897                   PHYSTATION_ADDRESS,
898                   STATION_NAME,
899                   gnbid,
900                   cell_name,
901                   cell_id,
902                   lon,
903                   lat,
904                   vender,
905                   construction
906     from seq_5g_siteinfo
907     where sdate = (select max(sdate) from seq_5g_siteinfo)
908         and is_alive = 1) a,

```

```

909         (select distinct city_name, GNBID, cell_id
910         from STATS_5g_CELL_FLOW_48h_hour
911         where sdate =
912         (select max(sdate) from STATS_5g_CELL_FLOW_48h_hour)) b
913     where /*a.city_name = b.city_name
914         and*/ a.GNBID = b.GNBID
915         and a.cell_id = b.cell_id
916     `
917 err = sqlQuery2Csv(db, exportsql, filepath)
918 if err != nil {
919     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
920 } else {
921     if isExist(filepath) {
922         waitFiles = append(waitFiles, filepath)
923     }
924 }
925
926 // 导出文件-17 4g小区错误或字段不全
927 fileName = "gcToyxgx_17_4g_cell_errors_" + dateDir + ".csv"
928 filepath = fileDir + "/" + fileName
929 exportsql = `
930 select distinct ERROR_CLASS      "异常类型",
931                CITY_NAME        "城市名称",
932                DISTRICT_NAME    "行政区名称",
933                CITY_CODE        "城市编码",
934                DISTRICT_CODE    "行政区编码",
935                NETWORK_NAME     "网络类型",
936                PHYSTATION_ADDRESS "物理站名称",
937                BBU_NAME         "BBU名称",
938                to_char(ENBID)    "基站ENBID",
939                CELL_NAME        "小区名称",
940                CELL_ID          "小区id",
941                LON              "小区经度",
942                LAT              "小区纬度",
943                DIRECTION        "小区方位角",
944                HEIGHT           "天线挂高",
945                M_DOWNTILT       "机械下倾",
946                E_DOWNTILT       "电子下倾",
947                STATION_TYPE     "基站类型",
948                ISDIGITALINDOOR  "是否电子室分",
949                DOWN_FREQ        "下行频点",
950                VENDER           "厂家",
951                OWN_SCHOOLYARD    "是否校园站",
952                TOWERADDRESS_CODE "铁塔站址编号",
953                PROPERTY         "产权归属",
954                SCENE            "场景",
955                IS_SCENESITE     "是否场景站",
956                MARKETING_NETWORK "销售网络",
957                TERMINALAMOUNT_5G "终端数量5G",
958                SECTOR_INCOMING  "扇区收入",
959                IS_BUSY          "是否超忙",
960                CONSTRUCTION     "承建方",
961                IS_SAME_ADDRESS  "是否共站址",
962                SAME_ADDRESS_SITES "对应共站址基站ID",
963                IS_SAME_ADDRESS_CT "共站址运营商"
964     from STATS_4G_ERROR_SITEINFO
965     where sdate = (select max(sdate) from STATS_4G_ERROR_SITEINFO)
966         and is_alive = 1

```

```

967
968 err = sqlQuery2Csv(db, exportsql, filepath)
969 if err != nil {
970     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
971 } else {
972     if isExist(filepath) {
973         waitFiles = append(waitFiles, filepath)
974     }
975 }
976
977 // 导出文件-18 5g小区错误或字段不全
978 fileName = "gcToyxgx_18_5g_cell_errors_" + dateDir + ".csv"
979 filepath = fileDir + "/" + fileName
980 exportsql = `
981 select distinct ERROR_CLASS "异常类型",
982                 CITY_NAME "城市名称",
983                 DISTRICT_NAME "行政区名称",
984                 CITY_CODE "城市编码",
985                 DISTRICT_CODE "行政区编码",
986                 NETWORK_NAME "网络类型",
987                 PHYSTATION_ADDRESS "物理站名称",
988                 STATION_NAME "基站名称",
989                 to_char(GNBID) "基站GNBID",
990                 CELL_NAME "小区名称",
991                 CELL_ID "小区id",
992                 LON "小区经度",
993                 LAT "小区纬度",
994                 DIRECTION "方位角",
995                 HEIGHT "天线挂高",
996                 M_DOWNTILT "机械下倾",
997                 E_DOWNTILT "电子下倾",
998                 STATION_TYPE "基站类型",
999                 ISDIGITALINDOOR "是否数字化室分",
1000                DOWN_FREQ "下行频点",
1001                VENDER "厂家",
1002                OWN_SCHOOLYARD "是否校园站",
1003                TOWERADDRESS_CODE "铁塔地址编码",
1004                PROPERTY "产权归属",
1005                SCENE "场景",
1006                IS_SCENESITE "是否场景站",
1007                MARKETING_NETWORK "销售网格",
1008                TERMINALAMOUNT_5G "终端数量5G",
1009                SECTOR_INCOMING "扇区收入",
1010                IS_BUSY "是否超忙",
1011                CONSTRUCTION "承建方",
1012                IS_SAME_ADDRESS "是否共站址",
1013                SAME_ADDRESS_SITES "对应共站址基站ID",
1014                IS_SAME_ADDRESS_CT "共站址运营商"
1015
1016                from STATS_5G_ERROR_SITEINFO
1017                where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
1018                and is_alive = 1
1019
1020
1021 err = sqlQuery2Csv(db, exportsql, filepath)
1022 if err != nil {
1023     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
1024 } else {
1025     if isExist(filepath) {

```

```

1025         waitFiles = append(waitFiles, filepath)
1026     }
1027 }
1028
1029 // 导出文件-19 4g经纬度偏移>3000小区数
1030 fileName = "gcToyxgx_19_4g_site_deviate_" + dateDir + ".csv"
1031 filepath = fileDir + "/" + fileName
1032 exportsql = `
1033 select CITY_NAME           "城市名称",
1034        DISTRICT_NAME      "行政区名称",
1035        CITY_CODE          "城市编码",
1036        DISTRICT_CODE      "行政区编码",
1037        PHYSTATION_ADDRESS "物理站址",
1038        BBU_NAME           "BBU名称",
1039        ENBID              "基站enbid",
1040        STATION_TYPE       "基站类型",
1041        VENDER             "设备厂家",
1042        SCENE              "覆盖厂家",
1043        DEVIATE_DISTANCE   "基站偏离距离",
1044        to_char(C1_ECI)    "小区1_eci",
1045        C1_CELL_NAME       "小区1_小区名称",
1046        C1_DISTANCE        "小区1_偏离距离",
1047        to_char(C2_ECI)    "小区2_eci",
1048        C2_CELL_NAME       "小区2_小区名称",
1049        C2_DISTANCE        "小区2_偏离距离",
1050        to_char(C3_ECI)    "小区3_eci",
1051        C3_CELL_NAME       "小区3_小区名称",
1052        C3_DISTANCE        "小区3_偏离距离",
1053        to_char(C4_ECI)    "小区4_eci",
1054        C4_CELL_NAME       "小区4_小区名称",
1055        C4_DISTANCE        "小区4_偏离距离",
1056        to_char(C5_ECI)    "小区5_eci",
1057        C5_CELL_NAME       "小区5_小区名称",
1058        C5_DISTANCE        "小区5_偏离距离",
1059        to_char(C6_ECI)    "小区6_eci",
1060        C6_CELL_NAME       "小区6_小区名称",
1061        C6_DISTANCE        "小区6_偏离距离",
1062        to_char(C7_ECI)    "小区7_eci",
1063        C7_CELL_NAME       "小区7_小区名称",
1064        C7_DISTANCE        "小区7_偏离距离",
1065        to_char(C8_ECI)    "小区8_eci",
1066        C8_CELL_NAME       "小区8_小区名称",
1067        C8_DISTANCE        "小区8_偏离距离",
1068        to_char(C9_ECI)    "小区9_eci",
1069        C9_CELL_NAME       "小区9_小区名称",
1070        C9_DISTANCE        "小区9_偏离距离"
1071 from stats_4G_DEVIATE_SUMMARY_DB
1072 `
1073 err = sqlQuery2Csv(db, exportsql, filepath)
1074 if err != nil {
1075     logger.Fatal("从数据库导出数据到csv失败:" + err.Error())
1076 } else {
1077     if isExist(filepath) {
1078         waitFiles = append(waitFiles, filepath)
1079     }
1080 }
1081
1082 // 导出文件-20 4g全量工参数据

```



```

1083 filename = "gcToyxgx_20_4g_seq_siteinfo_" + dateDir + ".csv"
1084 filepath = fileDir + "/" + fileName
1085 exportSql = `
1086     SELECT SDATE "数据日期",
1087           CITY_CODE "城市编码",
1088           CITY_NAME "城市名称",
1089           DISTRICT_CODE "行政区编码",
1090           DISTRICT_NAME "行政区名称",
1091           NETWORK_NAME "网络类型",
1092           PHYSTATION_ADDRESS "物理站名称",
1093           BBU_NAME "BBU名称",
1094           to_char(ENBID) "基站ENBID",
1095           CELL_NAME "小区名称",
1096           CELL_ID "小区id",
1097           to_char('46001' || enbid * 256 + cell_id) "小区全球唯一标识",
1098           LON "小区经度",
1099           LAT "小区纬度",
1100           DIRECTION "小区方位角",
1101           HEIGHT "天线挂高",
1102           M_DOWNTILT "机械下倾",
1103           E_DOWNTILT "电子下倾",
1104           STATION_TYPE "基站类型",
1105           ISDIGITALINDOOR "是否电子室分",
1106           DOWN_FREQ "下行频点",
1107           VENDER "厂家",
1108           OWN_SCHOOLYARD "是否校园站",
1109           TOWERADDRESS_CODE "铁塔站址编号",
1110           PROPERTY "产权归属",
1111           SCENE "场景",
1112           IS_SCENESITE "是否场景站",
1113           MARKETING_NETWORK "销售网络",
1114           TERMINALAMOUNT_5G "终端数量5G",
1115           SECTOR_INCOMING "扇区收入",
1116           IS_BUSY "是否超忙",
1117           IS_ALIVE "是否有业务",
1118           IS_ALIVE_UPDATE_TIME "业务更新时间",
1119           CONSTRUCTION "承建方",
1120           IS_SAME_ADDRESS "是否共站址",
1121           SAME_ADDRESS_SITES "对应共站址基站ID",
1122           IS_SAME_ADDRESS_CT "共站址运营商"
1123     FROM seq_4g_siteinfo
1124     where sdate = (select max(sdate) from seq_4g_siteinfo)
1125           and is_alive = 1
1126 `
1127 err = sqlQuery2Csv(db, exportSql, filepath)
1128 if err != nil {
1129     logger.Fatal("从数据库导出数据到csv失败", err)
1130 } else {
1131     if isExist(filepath) {
1132         waitFiles = append(waitFiles, filepath)
1133     }
1134 }
1135
1136 // 导出文件-21 5g全量工参数据
1137 fileName = "gcToyxgx_21_5g_seq_siteinfo_" + dateDir + ".csv"
1138 filepath = fileDir + "/" + fileName
1139 exportSql = `
1140     SELECT SDATE "数据日期",

```

```

1141     CITY_CODE "城市编码",
1142     CITY_NAME "城市名称",
1143     DISTRICT_CODE "行政区编码",
1144     DISTRICT_NAME "行政区名称",
1145     NETWORK_NAME "网络类型",
1146     PHYSTATION_ADDRESS "物理站名称",
1147     STATION_NAME "基站名称",
1148     to_char(GNBID) "基站GNBID",
1149     CELL_NAME "小区名称",
1150     CELL_ID "小区id",
1151     to_char('46001' || gnbid * 4096 + cell_id) "小区全球唯一识别码",
1152     LON "小区经度",
1153     LAT "小区纬度",
1154     DIRECTION "方位角",
1155     HEIGHT "天线挂高",
1156     M_DOWNTILT "机械下倾",
1157     E_DOWNTILT "电子下倾",
1158     STATION_TYPE "基站类型",
1159     ISDIGITALINDOOR "是否数字化室分",
1160     DOWN_FREQ "下行频点",
1161     VENDER "厂家",
1162     OWN_SCHOOLYARD "是否校园站",
1163     TOWERADDRESS_CODE "铁塔地址编码",
1164     PROPERTY "产权归属",
1165     SCENE "场景",
1166     IS_SCENESITE "是否场景站",
1167     MARKETING_NETWORK "销售网格",
1168     TERMINALAMOUNT_5G "终端数量5G",
1169     SECTOR_INCOMING "扇区收入",
1170     IS_BUSY "是否超忙",
1171     IS_ALIVE "是否有业务",
1172     IS_ALIVE_UPDATE_TIME "业务更新时间",
1173     CONSTRUCTION "承建方",
1174     IS_SAME_ADDRESS "是否共站址",
1175     SAME_ADDRESS_SITES "对应共站址基站ID",
1176     IS_SAME_ADDRESS_CT "共站址运营商"
1177 FROM seq_5g_siteinfo
1178 where sdate = (select max(sdate) from seq_5g_siteinfo)
1179     and is_alive = 1
1180
1181 err = sqlQuery2Csv(db, exportSql, filepath)
1182 if err != nil {
1183     logger.Fatal("从数据库导出数据到csv失败", err)
1184 } else {
1185     if isExist(filepath) {
1186         waitFiles = append(waitFiles, filepath)
1187     }
1188 }
1189
1190 // logFileName := "C:/Users/ZhangLei/Desktop/Tableau Desktop Pro
v2019.4.1 CRACK.rar"
1191
1192 ftpUploadFile("192.168.0.41:21", "do", "Richr00t", waitFiles,
"/data/bigscreen", dateDir)
1193 // for _, value := range waitFiles {
1194
1195     // ftpUploadFile("127.0.0.1:21", "user", "user", filepath, "./",
dateDir)

```

```

1196     // }
1197
1198 }
1199
1200 // 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; dayDir: 远程地址要创建的目录
1201 func ftpUploadFile(ftpserver string, ftpuser string, pw string, localFiles []string, remotePath string, dayDir string) {
1202     conn, err := ftp.Connect(ftpserver)
1203     if err != nil {
1204         logger.Fatal("Ftp连接失败:" + err.Error())
1205     } else {
1206         logger.Printf("Ftp连接成功")
1207     }
1208
1209     err = conn.Login(ftpuser, pw)
1210     if err != nil {
1211         logger.Fatal("Ftp登录失败:" + err.Error())
1212     } else {
1213         logger.Printf("Ftp登录成功")
1214     }
1215
1216     dir, err := conn.CurrentDir()
1217     if err != nil {
1218         logger.Fatal("打印当前工作目录失败:" + err.Error())
1219     } else {
1220         logger.Printf("当前工作目录:" + dir)
1221     }
1222     //切换到指定远端目录
1223     err = conn.ChangeDir(remotePath)
1224     if err != nil {
1225         logger.Printf("切换目录错误:" + err.Error())
1226     } else {
1227         logger.Printf("切换目录成功:" + remotePath)
1228     }
1229
1230     entries, err := conn.List(remotePath)
1231     if err != nil {
1232         logger.Printf(err.Error())
1233     }
1234     folderIsExists := false
1235     for _, entry := range entries {
1236         if entry.Type == ftp.EntryTypeFolder {
1237             if entry.Name == dayDir {
1238                 folderIsExists = true //说明目录存在
1239                 break
1240             }
1241         }
1242     }
1243     if !folderIsExists {
1244         err = conn.MakeDir(dayDir)
1245         if err != nil {
1246             logger.Printf("创建目录:" + dayDir + ";错误:" + err.Error())
1247         } else {
1248             logger.Printf("创建目录:" + dayDir + " 成功")
1249         }
1250     }

```

```

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

```

```

10
11 # 3) 工参问题+基站经纬度偏离
12 --create table STATS_4G_ERR_FLAG AS
13 with a_2d as
14 (select distinct ERROR_CLASS, a.city_name, a.ENBID, a.cell_id
15  from STATS_4G_ERROR_SITEINFO a
16  where a.is_alive = 1),
17 b_err as
18 (select distinct '基站经纬度偏离大于3000米' ERROR_CLASS,
19  a.city_name,
20  a.ENBID,
21  a.cell_id
22  from seq_4g_siteinfo a, STATS_4G_DEVIATE_SUMMARY_DB b
23  where a.sdate = (select max(sdate) from seq_4g_siteinfo)
24  and a.is_alive = 1
25  and a.city_name = b.city_name
26  and a.enbid = b.enbid)
27 select city_name,
28  ENBID,
29  cell_id,
30  listagg(ERROR_CLASS, ',') within group(order by ENBID) as ERROR_CLASS
31  from (select * from a_2d union all select * from b_err)
32  group by city_name, ENBID, cell_id;
33
34
35 # 4) 5g工参不全不准问题
36 SELECT distinct city_name, gnbid, CELL_ID, ERROR_CLASS
37  FROM STATS_5G_ERROR_SITEINFO
38  where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
39  and is_alive = 1;
40
41
42 # 5) 4g2天无业务
43 select distinct a.city_name,
44  a.ENBID,
45  a.cell_id,
46  '4G小区2天无业务' ERROR_CLASS
47  from seq_4g_siteinfo a, STATS_4g_CELL_FLOW_7d_DAY b
48  where a.sdate = (select max(sdate) from seq_4g_siteinfo)
49  and a.is_alive = 1
50  and b.sdate = (select max(sdate) from STATS_4g_CELL_FLOW_7d_DAY)
51  and a.enbid = b.enbid
52  and a.cell_id = b.cell_id;
53
54
55 # 6) 5g2天无业务
56 select distinct a.city_name,
57  a.gnbid,
58  a.cell_id,
59  '5G小区2天无业务' ERROR_CLASS
60  from seq_5g_siteinfo a, STATS_5g_CELL_FLOW_7d_DAY b
61  where a.sdate = (select max(sdate) from seq_5g_siteinfo)
62  and a.is_alive = 1
63  and b.sdate = (select max(sdate) from STATS_5g_CELL_FLOW_7d_DAY)
64  and a.gnbid = b.gnbid
65  and a.cell_id = b.cell_id;

```

6.4.2 工具源码

```
1 //oracle_2_excel.exe工具的源码:
2 package main
3
4 import (
5     "database/sql"
6     "fmt"
7     "log"
8     "os"
9     "strconv"
10    "strings"
11    "time"
12
13    _ "github.com/mattn/go-oci8"
14    "github.com/xuri/excelize"
15 )
16
17 var logger *log.Logger
18 var file *os.File
19 var err error
20
21 func init() {
22     file, err = os.OpenFile("logger.log",
23 os.O_APPEND|os.O_CREATE|os.O_WRONLY, 666)
24     if err != nil {
25         log.Fatal(err)
26     }
27
28     logger = log.New(file, "", log.LstdFlags)
29     logger.SetPrefix("Test- ") // 设置日志前缀
30     logger.SetFlags(log.Ldate | log.Ltime | log.Lshortfile)
31     /*
32     const (
33         // 数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。
34         // 在所有项目后会有一个冒号: 2009/01/23 01:23:23.123123
35         /a/b/c/d.go:23: message
36         Ldate          = 1 << iota // 日期: 2009/01/23
37         Ltime          // 时间: 01:23:23
38         Lmicroseconds // 微秒分辨率: 01:23:23.123123 (用于
39 增强Ltime位)
40         Llongfile      // 文件全路径名+行号:
41         /a/b/c/d.go:23
42         Lshortfile     // 文件无路径名+行号: d.go:23 (会覆盖
43 掉Llongfile)
44         LstdFlags      = Ldate | Ltime // 标准logger的初始值
45     )
46     */
47 }
48
49 func sqlQuery2Csv(db *sql.DB, sqlStmt string) (map[int][][]byte, []string,
50 error) {
51     rows, err := db.Query(sqlStmt)
52     if err != nil {
53         logger.Fatal("sqlQuery执行失败,err: %s,err sql:%s", err, sqlStmt)
54     }
55     defer rows.Close()
56 }
```

```

50 //返回所有列
51 cols, _ := rows.Columns()
52 //这里表示一行所有列的值, 用[]byte表示
53 result := make(map[int][][]byte, 10)
54
55 //这里表示一行填充数据
56 scans := make([]interface{}, len(cols))
57 //这里scans引用vals, 把数据填充到[]byte里
58
59 // 统计行数
60 i := 0
61 //result := make(map[int]map[string]string)
62
63 for rows.Next() {
64
65     //填充数据
66     vals := make([][]byte, len(cols))
67     for k, _ := range vals {
68         scans[k] = &vals[k]
69     }
70     rows.Scan(scans...)
71     //每行数据
72     //把vals中的数据复制到row中
73     // for _, v := range vals {
74     // // key := cols[k]
75     // //这里把[]byte数据转成string
76     // row = append(row, string(v))
77     // }
78     result[i] = vals
79     i++
80 }
81
82 return result, cols, nil
83 }
84
85 func isExist(slice []string, val string) (string, bool) {
86     for _, item := range slice {
87         if strings.ToUpper(item) == strings.ToUpper(val) {
88             return item, true
89         }
90     }
91     return "", false
92 }
93
94 func IsNum(s string) bool {
95     _, err := strconv.ParseFloat(s, 64)
96     return err == nil
97 }
98
99 func main() {
100     // 数据库部分
101     dateDir := time.Now().Format("20060102") // dateDir :=
time.Now().Format("200601")
102     db, err := sql.Open("oci8", fmt.Sprintf("%s/%s@s", "c##fast491",
"F@st491*321", "192.168.0.64:1521/fast"))
103     if err != nil {
104         logger.Fatalf("oracle登录失败:" + err.Error())
105     }

```

```

106 defer db.Close()
107 logger.Printf("数据库已登录")
108
109 err = db.Ping()
110 if err != nil {
111     logger.Fatalf("oracle链接不可达:" + err.Error())
112 }
113
114 // 创建Excel工作簿
115 file := excelize.NewFile()
116 styleID, err := file.NewStyle(`{"font":{"color": "#777777"}}`)
117 if err != nil {
118     logger.Println(err)
119 }
120
121 // 编号001
122 sheets := file.GetSheetList()
123
124 sname, existFlag := isExist(sheets, "sheet1")
125
126 if existFlag {
127     file.SetSheetName(sname, "4G工参不准不准+经纬度偏离")
128 } else {
129     file.NewSheet("4G工参不准不准+经纬度偏离")
130 }
131
132 streamWriter, err := file.NewStreamWriter("4G工参不准不准+经纬度偏离")
133 if err != nil {
134     logger.Println(err)
135 }
136 if err := streamWriter.SetRow("A1", []interface{}{
137     excelize.Cell{StyleID: styleID, Value: "Data"}}); err != nil {
138     logger.Println(err)
139 }
140
141 exportsql := `
142 with a_2d as
143     (select distinct ERROR_CLASS, a.city_name, a.ENBID, a.cell_id
144      from STATS_4G_ERROR_SITEINFO a
145      where a.is_alive = 1),
146 b_err as
147     (select distinct '基站经纬度偏离大于3000米' ERROR_CLASS,
148      a.city_name,
149      a.ENBID,
150      a.cell_id
151      from seq_4g_siteinfo a, STATS_4G_DEVIATE_SUMMARY_DB b
152      where a.sdate = (select max(sdate) from seq_4g_siteinfo)
153      and a.is_alive = 1
154      and a.city_name = b.city_name
155      and a.enbid = b.enbid)
156 select city_name,
157        ENBID,
158        cell_id,
159        listagg(ERROR_CLASS, ',') within group(order by ENBID) as
ERROR_CLASS
160     from (select * from a_2d union all select * from b_err)
161     group by city_name, ENBID, cell_id
162 `

```



```

163 // sql末尾不能有分号
164
165 data, coln, err := sqlQuery2Csv(db, exportSql)
166 //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。
167 colname := make([]interface{}, len(coln))
168 colLen := len(coln)
169 for i := 0; i < colLen; i++ {
170     colname[i] = coln[i]
171 }
172
173 // logger.Println(data)
174 cell, _ := excelize.CoordinatesToCellName(1, 1)
175 if err := streamWriter.SetRow(cell, colname); err != nil {
176     logger.Println(err)
177 }
178
179 rowLen := len(data)
180 for row := 0; row < rowLen; row++ {
181     v := data[row]
182     //rowLen索引从0开始, 第一行是表头, 因此只能从第二行写。
183     cell, _ := excelize.CoordinatesToCellName(1, row+2)
184     strV := make([]interface{}, len(v))
185     for k1, v1 := range v {
186         cellV := string(v1)
187         i1, err := strconv.Atoi(cellV)
188         if err != nil {
189             strV[k1] = cellV
190         } else {
191             strV[k1] = i1
192         }
193     }
194
195     // logger.Println(strV)
196
197     if err := streamWriter.SetRow(cell, strV); err != nil {
198         logger.Println(err)
199     }
200 }
201
202 if err := streamWriter.Flush(); err != nil {
203     logger.Println(err)
204 }
205 logger.Println("4G工参不准不准+经纬度偏离 写入成功")
206
207 // 编号002
208 sheets = file.GetSheetList()
209
210 sname, existFlag = isExist(sheets, strings.ToUpper("sheet2"))
211
212 if existFlag {
213     file.SetSheetName(sname, "5G工参不全不准")
214 } else {
215     file.NewSheet("5G工参不全不准")
216 }
217
218 streamWriter, err = file.NewStreamWriter("5G工参不全不准")
219 if err != nil {

```

```

220     logger.Println(err)
221 }
222 // if err := streamWriter.SetRow("A1", []interface{}{
223 //     excelize.Cell{StyleID: styleID, Value: "Data"}}); err != nil {
224 //     fmt.Println(err)
225 // }
226
227 exportsql = `
228 SELECT distinct city_name, to_char(gnbid) gnbid, CELL_ID, ERROR_CLASS
229     FROM STATS_5G_ERROR_SITEINFO
230     where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
231     and is_alive = 1
232 `
233 // sql末尾不能有分号
234
235 data, coln, err = sqlQuery2Csv(db, exportsql)
236 //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。
237 colname = make([]interface{}, len(coln))
238 colLen = len(coln)
239 for i := 0; i < colLen; i++ {
240     colname[i] = coln[i]
241 }
242
243 // logger.Println(data)
244 cell, _ = excelize.CoordinatesToCellName(1, 1)
245 if err := streamWriter.SetRow(cell, colname); err != nil {
246     logger.Println(err)
247 }
248
249 rowLen = len(data)
250 for row := 0; row < rowLen; row++ {
251     v := data[row]
252     //rowLen索引从0开始, 第一行是表头, 因此只能从第二行写。
253     cell, _ := excelize.CoordinatesToCellName(1, row+2)
254     strV := make([]interface{}, len(v))
255     for k1, v1 := range v {
256         cellV := string(v1)
257         i1, err := strconv.Atoi(cellV)
258         if err != nil {
259             strV[k1] = cellV
260         } else {
261             strV[k1] = i1
262         }
263     }
264
265     // logger.Println(strV)
266
267     if err := streamWriter.SetRow(cell, strV); err != nil {
268         logger.Println(err)
269     }
270 }
271 // 执行flush将数据写入sheet
272 if err := streamWriter.Flush(); err != nil {
273     logger.Println(err)
274 }
275
276 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)

```

```

334         i1, err := strconv.Atoi(cellV)
335         if err != nil {
336             strV[k1] = cellV
337         } else {
338             strV[k1] = i1
339         }
340     }
341
342     // logger.Println(strV)
343
344     if err := streamWriter.SetRow(cell, strV); err != nil {
345         logger.Println(err)
346     }
347 }
348 // 执行flush将数据写入sheet
349 if err := streamWriter.Flush(); err != nil {
350     logger.Println(err)
351 }
352
353 logger.Println("4G小区2天无业务 写入成功")
354
355 // 编号004
356 sheets = file.GetSheetList()
357
358 sname, existFlag = isExist(sheets, strings.ToUpper("sheet4"))
359
360 if existFlag {
361     file.SetSheetName(sname, "5G小区2天无业务")
362 } else {
363     file.NewSheet("5G小区2天无业务")
364 }
365
366 streamWriter, err = file.NewStreamWriter("5G小区2天无业务")
367 if err != nil {
368     logger.Println(err)
369 }
370 if err := streamWriter.SetRow("A1", []interface{}{
371     excelize.Cell{StyleID: styleID, value: "Data"}}); err != nil {
372     logger.Println(err)
373 }
374
375 exportSql = `
376 select distinct a.city_name,
377             to_char(a.gnbid) gnbid,
378             a.cell_id,
379             '5G小区2天无业务' ERROR_CLASS
380     from seq_5g_siteinfo a, STATS_5g_CELL_FLOW_7d_DAY b
381    where a.sdate = (select max(sdate) from seq_5g_siteinfo)
382          and a.is_alive = 1
383          and b.sdate = (select max(sdate) from STATS_5g_CELL_FLOW_7d_DAY)
384          and a.gnbid = b.gnbid
385          and a.cell_id = b.cell_id
386 `
387 // sql末尾不能有分号
388
389 data, colN, err = sqlQuery2Csv(db, exportSql)
390 //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。

```

```

391 colName = make([]interface{}, len(colN))
392 colLen = len(colN)
393 for i := 0; i < colLen; i++ {
394     colName[i] = colN[i]
395 }
396
397 // logger.Println(data)
398 cell, _ = excelize.CoordinatesToCellName(1, 1)
399 if err := streamWriter.SetRow(cell, colName); err != nil {
400     logger.Println(err)
401 }
402
403 rowLen = len(data)
404 for row := 0; row < rowLen; row++ {
405     v := data[row]
406     //rowLen索引从0开始，第一行是表头，因此只能从第二行写。
407     cell, _ := excelize.CoordinatesToCellName(1, row+2)
408     strV := make([]interface{}, len(v))
409     for k1, v1 := range v {
410         cellV := string(v1)
411         i1, err := strconv.Atoi(cellV)
412         if err != nil {
413             strV[k1] = cellV
414         } else {
415             strV[k1] = i1
416         }
417     }
418
419     // logger.Println(strV)
420
421     if err := streamWriter.SetRow(cell, strV); err != nil {
422         logger.Println(err)
423     }
424 }
425 // 执行flush将数据写入sheet
426 if err := streamWriter.Flush(); err != nil {
427     logger.Println(err)
428 }
429
430 logger.Println("5G小区2天无业务 写入成功")
431
432 // 编号005
433 sheets = file.GetSheetList()
434
435 sname, existFlag = isExist(sheets, strings.ToUpper("sheet5"))
436
437 if existFlag {
438     file.SetSheetName(sname, "4G有业务无工参")
439 } else {
440     file.NewSheet("4G有业务无工参")
441 }
442
443 streamWriter, err = file.NewStreamWriter("4G有业务无工参")
444 if err != nil {
445     logger.Println(err)
446 }
447 if err := streamWriter.SetRow("A1", []interface{}{
448     excelize.Cell{StyleID: styleID, Value: "Data"}}); err != nil {

```

```

449     logger.Println(err)
450 }
451
452 exportSql = `SELECT * FROM STATS_HAS_FLOW_NO_4G_GC
453 `
454 // sql末尾不能有分号
455
456 data, coln, err = sqlQuery2Csv(db, exportSql)
457 //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。
458 colname = make([]interface{}, len(coln))
459 colLen = len(coln)
460 for i := 0; i < colLen; i++ {
461     colname[i] = coln[i]
462 }
463
464 // logger.Println(data)
465 cell, _ = excelize.CoordinatesToCellName(1, 1)
466 if err := streamWriter.SetRow(cell, colname); err != nil {
467     logger.Println(err)
468 }
469
470 rowLen = len(data)
471 for row := 0; row < rowLen; row++ {
472     v := data[row]
473     //rowLen索引从0开始, 第一行是表头, 因此只能从第二行写。
474     cell, _ := excelize.CoordinatesToCellName(1, row+2)
475     strV := make([]interface{}, len(v))
476     for k1, v1 := range v {
477         cellV := string(v1)
478         i1, err := strconv.Atoi(cellV)
479         if err != nil {
480             strV[k1] = cellV
481         } else {
482             strV[k1] = i1
483         }
484     }
485
486     // logger.Println(strV)
487
488     if err := streamWriter.SetRow(cell, strV); err != nil {
489         logger.Println(err)
490     }
491 }
492 // 执行flush将数据写入sheet
493 if err := streamWriter.Flush(); err != nil {
494     logger.Println(err)
495 }
496
497 logger.Println("4G有业务无工参 写入成功")
498
499 // 编号005
500 sheets = file.GetSheetList()
501
502 sname, existFlag = isExist(sheets, strings.ToUpper("sheet6"))
503
504 if existFlag {
505     file.SetSheetName(sname, "5G有业务无工参")

```

```

506     } else {
507         file.NewSheet("5G有业务无工参")
508     }
509
510     streamWriter, err = file.NewStreamWriter("5G有业务无工参")
511     if err != nil {
512         logger.Println(err)
513     }
514     if err := streamWriter.SetRow("A1", []interface{}{
515         excelize.Cell{StyleID: styleID, value: "Data"}}); err != nil {
516         logger.Println(err)
517     }
518
519     exportSql = `SELECT * FROM STATS_HAS_FLOW_NO_5G_GC
520 `
521     // sql末尾不能有分号
522
523     data, coln, err = sqlQuery2Csv(db, exportSql)
524     //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。
525     colname = make([]interface{}, len(coln))
526     colLen = len(coln)
527     for i := 0; i < colLen; i++ {
528         colname[i] = coln[i]
529     }
530
531     // logger.Println(data)
532     cell, _ = excelize.CoordinatesToCellName(1, 1)
533     if err := streamWriter.SetRow(cell, colname); err != nil {
534         logger.Println(err)
535     }
536
537     rowLen = len(data)
538     for row := 0; row < rowLen; row++ {
539         v := data[row]
540         //rowLen索引从0开始, 第一行是表头, 因此只能从第二行写。
541         cell, _ := excelize.CoordinatesToCellName(1, row+2)
542         strV := make([]interface{}, len(v))
543         for k1, v1 := range v {
544             cellV := string(v1)
545             i1, err := strconv.Atoi(cellV)
546             if err != nil {
547                 strV[k1] = cellV
548             } else {
549                 strV[k1] = i1
550             }
551         }
552
553         // logger.Println(strV)
554
555         if err := streamWriter.SetRow(cell, strV); err != nil {
556             logger.Println(err)
557         }
558     }
559     // 执行flush将数据写入sheet
560     if err := streamWriter.Flush(); err != nil {
561         logger.Println(err)
562     }

```

```

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     {
568         logger.Println(err)
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);

```



```

41 procedure STATS_5g_CELL_FLOW_48h_hour(i_sdate varchar2,i_days number);
42 -- 工参不准或错误
43 procedure STATS_4G_ERROR_SITEINFO(i_sdate varchar2 default null);
44 procedure STATS_5G_ERROR_SITEINFO(i_sdate varchar2 default null);
45 -- 校园45G站点计算
46 procedure STATS_4G_COLLEGES_SITEINFO(i_sdate varchar2,outdoor_range
number,indoor_range number);
47 procedure STATS_5G_COLLEGES_SITEINFO(i_sdate varchar2,outdoor_range
number,indoor_range number);
48 -- 4G基站偏离度计算
49 procedure STATS_4G_DEVIATE_SUMMARY_DB(i_sdate varchar2 default null);
50 procedure STATS_4G_DEVIATE_SUMMARY_QL(i_sdate varchar2 default null);
51 procedure STATS_HAS_FLOW_NO_4G_GC(i_sdate varchar2 default null);
52 procedure STATS_HAS_FLOW_NO_5G_GC(i_sdate varchar2 default null);
53 end;
54

```

```

1 CREATE OR REPLACE PACKAGE BODY pkg_basedata_governance_v4 AS
2
3 --5g没有小区级小时粒度数据。
4
5 --1 4g 小区级小时粒度流量
6 /*procedure sdr_flow_4g_cell_hour(i_sdate varchar2) is
7
8 begin
9 delete from sdr_flow_4g_cell_hour where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
10 insert into sdr_flow_4g_cell_hour
11 select sdate,
12 enbid,
13 cell_id,
14 net,
15 LAYER2ID,
16 REGION_NAME,
17 LAYER3ID,
18 COUNTY_NAME,
19 sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
20 sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
21 sum(THROUGHPUT) THROUGHPUT
22 from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
23 TO_DATE('1970-01-01 08:00:00',
24 'YYYY-MM-DD HH24:MI:SS'),
25 'yyyy-mm-dd hh24'),
26 'yyyy-mm-dd hh24')*\ sdate,
27 t.ran_ne_id,
28 gnbid,
29 sitename,
30 cgisai,
31 --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
32 --case when length(cgisai)=12 then substr(cgisai,6,5)
else substr(cgisai,6,4) end enbid,
33 --case when length(cgisai)=12 then
substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
34 to_number((case
35 when length(cgisai) = 12 then

```

```

36         substr(cgisai, 6, 5)
37     else
38         substr(cgisai, 6, 4)
39     end),
40     'xxxxxxx') enbid,
41     to_number((case
42         when length(cgisai) = 12 then
43             substr(cgisai, -2)
44         else
45             substr(cgisai, -4)
46         end),
47     'xxxxxxx') cell_id,
48     \*
49
50 IF (LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))
51
52 IF (LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
53     rat,
54     CASE
55     WHEN rat = 9 THEN
56         '5G'
57     WHEN RAT = 6 THEN
58         '4G'
59     END NET,
60     l4_dw_throughput,
61     l4_ul_throughput,
62     (l4_dw_throughput + l4_ul_throughput) throughput,
63     layer2id,
64     d.region_name,
65     layer3id,
66     h.county_name
67 FROM sdr_flow_cell_source_csv t
68 LEFT JOIN sdr_flow_region_code d
69 ON t.layer2id = d.region_id
70 LEFT JOIN sdr_flow_mapping_ran_ne_id g
71 ON t.ran_ne_id = g.ran_ne_id
72 LEFT JOIN SDR_FLOW_county_code h
73 ON t.layer3id = h.county_id
74 WHERE T.RAT = 6
75     and cgisai is not null
76     and trunc(sdate) = to_date(i_sdate,'yyyymmdd')
77 group by sdate,
78     enbid,
79     cell_id,
80     net,
81     LAYER2ID,
82     REGION_NAME,
83     LAYER3ID,
84     COUNTY_NAME;
85
86 end;*/
87
88 --2 4g 小区级天粒度流量
89 /*procedure sdr_flow_4g_cell_day(i_sdate varchar2) is
90 begin

```

```

91      delete from sdr_flow_4g_cell_day where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
92      insert into sdr_flow_4g_cell_day
93      select trunc(sdate) sdate,
94             enbid,
95             cell_id,
96             net,
97             LAYER2ID,
98             REGION_NAME,
99             LAYER3ID,
100            COUNTY_NAME,
101
102            sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
103            sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
104            sum(THROUGHPUT) THROUGHPUT
105      from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
106                               TO_DATE('1970-01-01 08:00:00',
107                               'YYYY-MM-DD HH24:MI:SS'),
108                               'yyyymmdd hh24'),
109                               'yyyymmdd hh24')*\ sdate,
110             t.ran_ne_id,
111             gnbid,
112             sitename,
113             cgisai,
114             --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
115             --case when length(cgisai)=12 then substr(cgisai,6,5)
else substr(cgisai,6,4) end enbid,
116             --case when length(cgisai)=12 then
substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
117             to_number((case
118                 when length(cgisai) = 12 then
119                     substr(cgisai, 6, 5)
120                 else
121                     substr(cgisai, 6, 4)
122             end),
123             'xxxxxxx') enbid,
124             to_number((case
125                 when length(cgisai) = 12 then
126                     substr(cgisai, -2)
127                 else
128                     substr(cgisai, -4)
129             end),
130             'xxxxxxx') cell_id,
131
132            \*
133
134      IF (LEN(B2)=12, HEX2DEC(MID(B2, 6, 5)), HEX2DEC(MID(B2, 6, 4)))
135
136      IF (LEN(B2)=12, HEX2DEC(RIGHT(B2, 2)), HEX2DEC(RIGHT(B2, 4)))*\
137      rat,
138      CASE
139      WHEN rat = 9 THEN
140      '5G'
141      WHEN RAT = 6 THEN
142      '4G'
143      END NET,
144      14_dw_throughput,
145      14_ul_throughput,

```

```

144         (l4_dw_throughput + l4_ul_throughput) throughput,
145         layer2id,
146         d.region_name,
147         layer3id,
148         h.county_name
149     FROM sdr_flow_cell_source_csv t
150     LEFT JOIN sdr_flow_region_code d
151         ON t.layer2id = d.region_id
152     LEFT JOIN sdr_flow_mapping_ran_ne_id g
153         ON t.ran_ne_id = g.ran_ne_id
154     LEFT JOIN SDR_FLOW_county_code h
155         ON t.layer3id = h.county_id
156     WHERE T.RAT = 6
157         and cgisai is not null
158         and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
159     group by trunc(sdate),
160             enbid,
161             cell_id,
162             net,
163             LAYER2ID,
164             REGION_NAME,
165             LAYER3ID,
166             COUNTY_NAME;
167
168     end;*/
169
170
171
172     --3 4g 基站级小时粒度流量
173     /*procedure sdr_flow_4g_enb_hour(i_sdate varchar2) is
174
175     begin
176         delete from sdr_flow_4g_enb_hour where trunc(sdate) =
177             to_date(i_sdate,'yyyymmdd');
178         insert into sdr_flow_4g_enb_hour
179         select sdate,
180             enbid,
181             net,
182             LAYER2ID,
183             REGION_NAME,
184             LAYER3ID,
185             COUNTY_NAME,
186
187             sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
188             sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
189             sum(THROUGHPUT) THROUGHPUT
190         from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
191             TO_DATE('1970-01-01 08:00:00',
192             'YYYY-MM-DD HH24:MI:SS'),
193             'yyyy-mm-dd hh24'),
194             'yyyy-mm-dd hh24')*\ sdate,
195             t.ran_ne_id,
196             gnbid,
197             sitename,
198             cgisai,
199             --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
200             --case when length(cgisai)=12 then substr(cgisai,6,5)
201             else substr(cgisai,6,4) end enbid,

```

```

200         --case when length(cgisai)=12 then
substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
201         to_number((case
202             when length(cgisai) = 12 then
203                 substr(cgisai, 6, 5)
204             else
205                 substr(cgisai, 6, 4)
206             end),
207             'XXXXXXXX') enbid,
208         to_number((case
209             when length(cgisai) = 12 then
210                 substr(cgisai, -2)
211             else
212                 substr(cgisai, -4)
213             end),
214             'XXXXXXXX') cell_id,
215
216         \*
217
IF(LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))
218
IF(LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
219         rat,
220         CASE
221             WHEN rat = 9 THEN
222                 '5G'
223             WHEN RAT = 6 THEN
224                 '4G'
225         END NET,
226         l4_dw_throughput,
227         l4_ul_throughput,
228         (l4_dw_throughput + l4_ul_throughput) throughput,
229         layer2id,
230         d.region_name,
231         layer3id,
232         h.county_name
233     FROM sdr_flow_cell_source_csv t
234     LEFT JOIN sdr_flow_region_code d
235         ON t.layer2id = d.region_id
236     LEFT JOIN sdr_flow_mapping_ran_ne_id g
237         ON t.ran_ne_id = g.ran_ne_id
238     LEFT JOIN SDR_FLOW_county_code h
239         ON t.layer3id = h.county_id
240     WHERE T.RAT = 6
241         and cgisai is not null
242         and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
243     group by sdate,
244             enbid,
245             net,
246             LAYER2ID,
247             REGION_NAME,
248             LAYER3ID,
249             COUNTY_NAME;
250
251     end;*/
252
253
254

```

```

255 --4 4g 基站级天粒度流量
256 /*procedure sdr_flow_4g_enb_day(i_sdate varchar2) is
257
258 begin
259     delete from sdr_flow_4g_enb_day where trunc(sdate) =
to_date(i_sdate,'yyyymmdd');
260     insert into sdr_flow_4g_enb_day
261     select trunc(sdate) sdate,
262            enbid,
263            net,
264            LAYER2ID,
265            REGION_NAME,
266            LAYER3ID,
267            COUNTY_NAME,
268
269            sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
270            sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
271            sum(THROUGHPUT) THROUGHPUT
272     from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
273                            TO_DATE('1970-01-01 08:00:00',
274                                    'YYYY-MM-DD HH24:MI:SS'),
275                                    'yyyy-mm-dd hh24'),
276                            'yyyy-mm-dd hh24')*\ sdate,
277            t.ran_ne_id,
278            gnbid,
279            sitename,
280            cgisai,
281            --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
282            --case when length(cgisai)=12 then substr(cgisai,6,5)
else substr(cgisai,6,4) end enbid,
283            --case when length(cgisai)=12 then
substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
284            to_number((case
285                        when length(cgisai) = 12 then
286                            substr(cgisai, 6, 5)
287                        else
288                            substr(cgisai, 6, 4)
289                        end),
290            'xxxxxxx') enbid,
291            to_number((case
292                        when length(cgisai) = 12 then
293                            substr(cgisai, -2)
294                        else
295                            substr(cgisai, -4)
296                        end),
297            'xxxxxxx') cell_id,
298
299            \*
300
IF(LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))
301
IF(LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
302            rat,
303            CASE
304                WHEN rat = 9 THEN
305                    '5G'
306                WHEN RAT = 6 THEN
307                    '4G'

```

```

308         END NET,
309         l4_dw_throughput,
310         l4_ul_throughput,
311         (l4_dw_throughput + l4_ul_throughput) throughput,
312         layer2id,
313         d.region_name,
314         layer3id,
315         h.county_name
316     FROM sdr_flow_cell_source_csv t
317     LEFT JOIN sdr_flow_region_code d
318         ON t.layer2id = d.region_id
319     LEFT JOIN sdr_flow_mapping_ran_ne_id g
320         ON t.ran_ne_id = g.ran_ne_id
321     LEFT JOIN SDR_FLOW_county_code h
322         ON t.layer3id = h.county_id
323     WHERE T.RAT = 6
324           and cgisai is not null
325           and trunc(sdate) = to_date(i_sdate,'yyyymmdd')
326     group by trunc(sdate),
327            enbid,
328            net,
329            LAYER2ID,
330            REGION_NAME,
331            LAYER3ID,
332            COUNTY_NAME;
333
334     end;*/
335
336
337     --5 5g 基站级小时粒度流量
338     /*procedure sdr_flow_5g_enb_hour(i_sdate varchar2) is
339
340     begin
341         delete from sdr_flow_5g_enb_hour where trunc(sdate) =
342         to_date(i_sdate,'yyyymmdd');
343         insert into sdr_flow_5g_enb_hour
344         select sdate,
345                RAN_NE_ID,
346                GNBID,
347                SITENAME,
348                net,
349                LAYER2ID,
350                REGION_NAME,
351                LAYER3ID,
352                COUNTY_NAME,
353
354                sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
355                sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
356                sum(THROUGHPUT) THROUGHPUT
357         from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
358                TO_DATE('1970-01-01 08:00:00',
359                'YYYY-MM-DD HH24:MI:SS'),
360                'yyyy-mm-dd hh24'),
361                'yyyy-mm-dd hh24')*\ sdate,
362                t.ran_ne_id,
363                gnbid,
364                sitename,
365                cgisai,

```

```

365         --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
366         --case when length(cgisai)=12 then substr(cgisai,6,5)
else substr(cgisai,6,4) end enbid,
367         --case when length(cgisai)=12 then
substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
368         to_number((case
369             when length(cgisai) = 12 then
370                 substr(cgisai, 6, 5)
371             else
372                 substr(cgisai, 6, 4)
373         end),
374         'xxxxxxxx') enbid,
375         to_number((case
376             when length(cgisai) = 12 then
377                 substr(cgisai, -2)
378             else
379                 substr(cgisai, -4)
380         end),
381         'xxxxxxxx') cell_id,
382
383         \*
384
IF (LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))
385
IF (LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
386         rat,
387         CASE
388             WHEN rat = 9 THEN
389                 '5G'
390             WHEN RAT = 6 THEN
391                 '4G'
392         END NET,
393         l4_dw_throughput,
394         l4_ul_throughput,
395         (l4_dw_throughput + l4_ul_throughput) throughput,
396         layer2id,
397         d.region_name,
398         layer3id,
399         h.county_name
400     FROM sdr_flow_cell_source_csv t
401     LEFT JOIN sdr_flow_region_code d
402         ON t.layer2id = d.region_id
403     LEFT JOIN sdr_flow_mapping_ran_ne_id g
404         ON t.ran_ne_id = g.ran_ne_id
405     LEFT JOIN SDR_FLOW_county_code h
406         ON t.layer3id = h.county_id
407     WHERE T.RAT = 9
408     and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
409     group by sdate,
410             RAN_NE_ID,
411             GNBID,
412             SITENAME,
413             net,
414             LAYER2ID,
415             REGION_NAME,
416             LAYER3ID,
417             COUNTY_NAME;
418

```



```

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) =
427         to_date(i_sdate,'yyyymmdd');
428         insert into sdr_flow_5g_enb_day
429         select trunc(sdate) sdate,
430                RAN_NE_ID,
431                GNBID,
432                SITENAME,
433                net,
434                LAYER2ID,
435                REGION_NAME,
436                LAYER3ID,
437                COUNTY_NAME,
438
439                sum(L4_DW_THROUGHPUT) L4_DW_THROUGHPUT,
440                sum(L4_UL_THROUGHPUT) L4_UL_THROUGHPUT,
441                sum(THROUGHPUT) THROUGHPUT
442         from (SELECT \*to_date(to_char(starttime / (60 * 60 * 24) +
443                                TO_DATE('1970-01-01 08:00:00',
444                                'YYYY-MM-DD HH24:MI:SS'),
445                                'yyyy-mm-dd hh24'),
446                                'yyyy-mm-dd hh24')*\ sdate,
447                t.ran_ne_id,
448                gnbid,
449                sitename,
450                cgisai,
451                --CONV(SUBSTR(cgisai, 6), 16, 10) eci,
452                --case when length(cgisai)=12 then substr(cgisai,6,5)
453                else substr(cgisai,6,4) end enbid,
454                --case when length(cgisai)=12 then
455                substr(cgisai,-2,2) else substr(cgisai,-4,4) end cid,
456                to_number((case
457                            when length(cgisai) = 12 then
458                                substr(cgisai, 6, 5)
459                            else
460                                substr(cgisai, 6, 4)
461                            end),
462                            'XXXXXXXX') enbid,
463                to_number((case
464                            when length(cgisai) = 12 then
465                                substr(cgisai, -2)
466                            else
467                                substr(cgisai, -4)
468                            end),
469                            'XXXXXXXX') cell_id,
470
471                \*
472
473                IF(LEN(B2)=12,HEX2DEC(MID(B2,6,5)),HEX2DEC(MID(B2,6,4)))
474
475                IF(LEN(B2)=12,HEX2DEC(RIGHT(B2,2)),HEX2DEC(RIGHT(B2,4)))*\
476                rat,

```

```

472             CASE
473                 WHEN rat = 9 THEN
474                     '5G'
475                 WHEN RAT = 6 THEN
476                     '4G'
477             END NET,
478             14_dw_throughput,
479             14_ul_throughput,
480             (14_dw_throughput + 14_ul_throughput) throughput,
481             layer2id,
482             d.region_name,
483             layer3id,
484             h.county_name
485         FROM sdr_flow_cell_source_csv t
486         LEFT JOIN sdr_flow_region_code d
487             ON t.layer2id = d.region_id
488         LEFT JOIN sdr_flow_mapping_ran_ne_id g
489             ON t.ran_ne_id = g.ran_ne_id
490         LEFT JOIN SDR_FLOW_county_code h
491             ON t.layer3id = h.county_id
492         WHERE T.RAT = 9
493             and trunc(sdate) = to_date(i_sdate,'yyyymmdd'))
494     group by trunc(sdate),
495             RAN_NE_ID,
496             GNBID,
497             SITENAME,
498             net,
499             LAYER2ID,
500             REGION_NAME,
501             LAYER3ID,
502             COUNTY_NAME;
503
504 end;*/
505 procedure ini_4g_table is
506 begin
507
508     dbms_output.put_line('start reset table ...');
509
510     execute immediate 'truncate table stats_4g_cell_flow_2m_day';
511     execute immediate 'truncate table stats_4g_cell_flow_48h_hour';
512     execute immediate 'truncate table stats_4g_cell_flow_7d_day';
513     execute immediate 'truncate table stats_4g_colleges_siteinfo';
514     execute immediate 'truncate table stats_4g_error_siteinfo';
515     execute immediate 'truncate table stats_4g_siteinfo_deviate';
516     execute immediate 'truncate table stats_4g_site_flow_2m_day';
517     execute immediate 'truncate table stats_4g_site_flow_48h_hour';
518     execute immediate 'truncate table stats_4g_site_flow_7d_day';
519     execute immediate 'truncate table STATS_4g_SITE_LOWFLOW_7d_DAY';
520     execute immediate 'truncate table STATS_4g_CELL_LOWFLOW_7d_DAY';
521
522     dbms_output.put_line('reset table end...');
523
524 end;
525
526 procedure ini_5g_table is
527 begin
528
529     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

```

```

588      --获得小区流量源表中所有地市的最小日期
589      select min(sdate) sdate_min
590      into istart
591      from (select max(sdate) sdate from stats_4g_cell_flow_2m_day
592            union
593            select to_date(max(sdate), 'yyyymmdd') sdate from
sdr_flow_4g_enb_day
594            union
595            select trunc(sysdate) from dual
596            );
597      dbms_output.put_line('cal_4g_data: start date - ' || to_char(istart,
'yyyymmdd'));
598      --获得结果表的最小日期
599      select min(to_date(sdate, 'yyyymmdd')) sdate_max
600      into iend
601      from (select max(sdate) sdate from sdr_flow_4g_cell_day
602            union
603            select max(sdate) sdate from sdr_flow_4g_enb_day
604            union
605            select to_char(sysdate, 'yyyymmdd') from dual
606            );
607      dbms_output.put_line('cal_4g_data: end date - ' || to_char(iend,
'yyyymmdd'));
608      --dbms_output.put_line(istart);
609      --dbms_output.put_line(iend);
610      --计算时间差, 用于实现遍历的时间列表
611      igap := iend - istart;
612      --dbms_output.put_line(igap);
613      if igap >=1 then
614          for v in (select istart + level sdate from dual connect by level
<= igap) loop
615
616              dbms_output.put_line('cal_4g_data: date - ' || to_char(v.sdate,
'yyyymmdd') || ' start process');
617
618              STATS_4g_SITE_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
619              STATS_4g_CELL_FLOW_2M_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
620
621              STATS_4g_SITE_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);
622              STATS_4g_CELL_FLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 1);
623
624              STATS_4g_SITE_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);
625              STATS_4g_CELL_FLOW_48h_hour(to_char(v.sdate, 'yyyymmdd'), 0);
626
627              STATS_4g_SITE_LOWFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
628              STATS_4g_CELL_LOWFLOW_7d_DAY(to_char(v.sdate, 'yyyymmdd'), 6);
629          end loop;
630      end if;
631      -- 执行工参异常检测
632      dbms_output.put_line('check_45site: 0 -
STATS_4G_ERROR_SITEINFO/STATS_4G_COLLEGES_SITEINFO - ' || sdate || ' start
process');
633      select to_char(max(sdate), 'yyyymmdd') into sdate from
seq_4g_siteinfo;
634      STATS_4G_ERROR_SITEINFO(sdate);
635      STATS_4G_COLLEGES_SITEINFO(sdate, 50, 50);
636      -- 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         irstart varchar2(8);
705 begin
706     irstart:='20210720';
707
708     if i_sdate is null then
709         select min(sdate) into irstart
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         irstart := i_sdate;
720     end if;
721
722     dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
irstart=' || irstart);
723     dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
start');
724
725     STATS_4g_SITE_FLOW_2M_DAY(irstart, 6);
726     STATS_4g_CELL_FLOW_2M_DAY(irstart, 6);
727     STATS_4g_SITE_FLOW_7d_DAY(irstart, 1);
728     STATS_4g_CELL_FLOW_7d_DAY(irstart, 1);
729     STATS_4g_SITE_FLOW_48h_hour(irstart, 0);
730     STATS_4g_CELL_FLOW_48h_hour(irstart, 0);
731     STATS_4g_SITE_LOWFLOW_7d_DAY(irstart, 6);
732     STATS_4g_CELL_LOWFLOW_7d_DAY(irstart, 6);
733
734     dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
end');
735
736 end;
737

```

```

738     procedure STATS_5G_OMCFLOW(i_sdate   varchar2 default null) is --5G工
    参中的最新日期
739         istrict varchar2(8);
740     begin
741         istrict:='20210720';
742
743         if i_sdate is null then
744             select to_char(min(sdate), 'yyyymmdd')
745                 into istrict
746                 from (select city, max(sdate) sdate from SDR_FLOW_CELL_THROUGHPUT
    group by city
747                     union
748                     select 'system', trunc(sysdate) from dual);
749         else
750             istrict := i_sdate;
751         end if;
752
753         dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
    istrict=' || istrict);
754         dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
    start');
755
756         sdr_flow_5g_cell_day(istrict);
757         sdr_flow_5g_enb_day(istrict);
758         -----
759         STATS_5g_SITE_FLOW_2M_DAY(istrict, 6);
760         STATS_5g_phySITE_FLOW_2M_DAY(istrict, 6);
761         STATS_5g_CELL_FLOW_2M_DAY(istrict, 6);
762         -----
763         STATS_5g_SITE_FLOW_7d_DAY(istrict, 1);
764         STATS_5g_phySITE_FLOW_7d_DAY(istrict, 1);
765         STATS_5g_CELL_FLOW_7d_DAY(istrict, 1);
766         -----
767         STATS_5g_SITE_FLOW_48h_hour(istrict, 0);
768         STATS_5g_phySITE_FLOW_48h_hour(istrict, 0);
769         STATS_5g_CELL_FLOW_48h_hour(istrict, 0);
770
771         STATS_5g_SITE_LOWFLOW_7d_DAY(istrict, 6);
772         STATS_5g_CELL_LOWFLOW_7d_DAY(istrict, 6);
773         STATS_5g_PHYSITE_LOFLOW_7d_DAY(istrict, 6);
774         dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
    end');
775
776     end;
777
778     --7 5g 小区级天粒度流量
779     -- 20210726 不在使用
780     procedure SDR_FLOW_5G_CELL_DAY(i_sdate varchar2) is
781
782     begin
783         delete from sdr_flow_5g_cell_day where sdate =i_sdate;
784         insert into sdr_flow_5g_cell_day
785             select to_char(SDATE, 'yyyymmdd') SDATE,
786                 GNBID,
787                 CELL_ID,
788                 CITY,
789                 DISTRICT,
790                 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

```

```

1001         to_date(i_sdate, 'yyyymmdd') - idays and
1002         to_date(i_sdate, 'yyyymmdd') --29 时间参数
1003         group by enbid, cell_id
1004         having sum(nvl(THROUGHPUT,0)) > 0
1005     ) b
1006     on a.enbid = b.enbid
1007     and a.cell_id = b.cell_id)
1008 where benbid is null;
1009 commit;
1010
1011
1012 end;
1013
1014
1015
1016
1017 procedure STATS_5g_SITE_FLOW_7d_DAY(i_sdate varchar2, idays number) is
1018
1019 begin
1020     delete from STATS_5g_SITE_FLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1021     insert into STATS_5g_SITE_FLOW_7d_DAY
1022     select SDATE, CITY_CODE, CITY_NAME, GNBID --, BGNBID
1023     from (select to_date(i_sdate, 'yyyymmdd') sdate,
1024             a.CITY_CODE,
1025             a.CITY_NAME,
1026             a.gnbid,
1027             b.gnbid bgnbid
1028     from (SELECT distinct CITY_CODE, CITY_NAME, GNBID
1029     FROM seq_5g_siteinfo where sdate = (select
max(sdate) from seq_5g_siteinfo)) a
1030     left join (SELECT gnbid
1031     FROM SDR_FLOW_CELL_THROUGHPUT
1032     where sdate between
1033     to_date(i_sdate, 'yyyymmdd') - idays and
1034     to_date(i_sdate, 'yyyymmdd') --29 时间参数
1035     group by gnbid
1036     having
sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,0)) > 0
1037     ) b
1038     on a.gnbid = b.gnbid)
1039     where bgnbid is null;
1040     commit;
1041
1042 end;
1043
1044 procedure STATS_5g_CELL_FLOW_7d_DAY(i_sdate varchar2, idays number) is
1045
1046 begin
1047     delete from STATS_5g_CELL_FLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1048     insert into STATS_5g_CELL_FLOW_7d_DAY
1049     select distinct SDATE, CITY_CODE, CITY_NAME, GNBID, CELL_ID--,
BGNBID, BCELL_ID
1050     from (select to_date(i_sdate, 'yyyymmdd') sdate,
1051             a.CITY_CODE,
1052             a.CITY_NAME,
1053             a.GNBID,

```

```

1054         a.cell_id,
1055         b.GNBID bgnbid,
1056         b.cell_id bcell_id
1057     from (SELECT distinct CITY_CODE, CITY_NAME, GNBID, CELL_ID
1058           FROM seq_5g_siteinfo where sdate = (select
max(sdate) from seq_5g_siteinfo)) a
1059     left join (SELECT GNBID, CELL_ID
1060               FROM SDR_FLOW_CELL_THROUGHPUT
1061               where sdate between
1062                     to_date(i_sdate, 'yyyymmdd') - idays and
1063                     to_date(i_sdate, 'yyyymmdd') --29 时间参数
1064               group by GNBID, CELL_ID
1065               having
sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,0)) > 0
1066             ) b
1067     on a.GNBID = b.GNBID
1068     and a.cell_id = b.cell_id)
1069     where bgnbid is null;
1070     commit;
1071
1072 end;
1073
1074
1075 procedure STATS_4g_SITE_LOWFLOW_7d_DAY(i_sdate varchar2, idays number)
is
1076
1077 begin
1078     delete from STATS_4g_SITE_LOWFLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1079     insert into STATS_4g_SITE_LOWFLOW_7d_DAY
1080     select to_date(i_sdate, 'yyyymmdd') sdate,
1081           a.CITY_CODE,
1082           a.CITY_NAME,
1083           a.enbid,
1084           b.THROUGHPUT
1085     from (SELECT distinct CITY_CODE, CITY_NAME, ENBID
1086           FROM seq_4g_siteinfo
1087           where sdate = (select max(sdate) from seq_4g_siteinfo)) a
1088     inner join (SELECT enbid, sum(nvl(THROUGHPUT, 0)) THROUGHPUT
1089               FROM sdr_flow_4g_enb_day
1090               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
1091             /* and */
1092             to_date(sdate, 'yyyymmdd') between
1093             to_date(i_sdate, 'yyyymmdd') - idays and
1094             to_date(i_sdate, 'yyyymmdd') --29 时间参数
1095             group by enbid
1096             having sum(nvl(THROUGHPUT, 0)) > 0 and
sum(nvl(THROUGHPUT, 0)) < 7516192768) b
1097     on a.enbid = b.enbid;
1098     commit;
1099
1100
1101 end;
1102
1103
1104
1105

```

```

1106     procedure STATS_4g_CELL_LOWFLOW_7d_DAY(i_sdate varchar2, idays number)
1107     is
1108     begin
1109         delete from STATS_4g_CELL_LOWFLOW_7d_DAY where trunc(sdate) =
1110         to_date(i_sdate, 'yyyymmdd');
1111         insert into STATS_4g_CELL_LOWFLOW_7d_DAY
1112         select to_date(i_sdate, 'yyyymmdd') sdate,
1113                a.CITY_CODE,
1114                a.CITY_NAME,
1115                a.enbid,
1116                a.cell_id,
1117                b.THROUGHPUT
1118         from (SELECT distinct CITY_CODE, CITY_NAME, ENBID, cell_id
1119                FROM seq_4g_siteinfo
1120                where sdate = (select max(sdate) from seq_4g_siteinfo)) a
1121         inner join (SELECT enbid,
1122                cell_id,
1123                sum(nvl(THROUGHPUT, 0)) THROUGHPUT
1124                FROM sdr_flow_4g_cell_day
1125                where --nvl(THROUGHPUT,0) > 0 and nvl(THROUGHPUT,0)
1126                <2147483648 -- 低业务量: 扇区按<2GB计算, sdr单位为byte, 因此需除以3个1024
1127                /* and*/
1128                to_date(sdate, 'yyyymmdd') between
1129                to_date(i_sdate, 'yyyymmdd') - idays and
1130                to_date(i_sdate, 'yyyymmdd') --29 时间参数
1131                group by enbid, cell_id
1132                having sum(nvl(THROUGHPUT, 0)) > 0 and
1133                sum(nvl(THROUGHPUT, 0)) < 2147483648) b
1134         on a.enbid = b.enbid
1135         and a.cell_id = b.cell_id;
1136     commit;
1137 end;
1138
1139 -- 5G 业务量单位为MByte, 因此换单单位>7GB, 为>7*1024
1140 procedure STATS_5g_SITE_LOWFLOW_7d_DAY(i_sdate varchar2, idays number)
1141 is
1142 begin
1143     delete from STATS_5g_SITE_LOWFLOW_7d_DAY where trunc(sdate) =
1144     to_date(i_sdate, 'yyyymmdd');
1145     insert into STATS_5g_SITE_LOWFLOW_7d_DAY
1146     select to_date(i_sdate, 'yyyymmdd') sdate,
1147            a.CITY_CODE,
1148            a.CITY_NAME,
1149            a.gnbid,
1150            b.THROUGHPUT
1151     from (SELECT distinct CITY_CODE, CITY_NAME, GNBID
1152            FROM seq_5g_siteinfo
1153            where sdate = (select max(sdate) from seq_5g_siteinfo)) a
1154     inner join (SELECT gnbid,
1155            sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) THROUGHPUT
1156            FROM SDR_FLOW_CELL_THROUGHPUT
1157            where sdate between
1158            to_date(i_sdate, 'yyyymmdd') - idays and
1159            to_date(i_sdate, 'yyyymmdd') --29 时间参数
1160            group by gnbid

```

```

1157             having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) >
0 and sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) < 7168) b
1158         on a.gnbid = b.gnbid;
1159         commit;
1160
1161     end;
1162
1163     procedure STATS_5g_CELL_LOWFLOW_7d_DAY(i_sdate varchar2, idays number)
is
1164
1165     begin
1166         delete from STATS_5g_CELL_LOWFLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1167         insert into STATS_5g_CELL_LOWFLOW_7d_DAY
1168         select to_date(i_sdate, 'yyyymmdd') sdate,
1169             a.CITY_CODE,
1170             a.CITY_NAME,
1171             a.GNBID,
1172             a.cell_id,
1173             b.THROUGHPUT
1174         from (SELECT distinct CITY_CODE, CITY_NAME, GNBID, CELL_ID
1175             FROM seq_5g_siteinfo
1176             where sdate = (select max(sdate) from seq_5g_siteinfo)) a
1177         inner join (SELECT GNBID,
1178             CELL_ID,
1179             sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0))
THROUGHPUT
1180             FROM SDR_FLOW_CELL_THROUGHPUT
1181             where sdate between
1182             to_date(i_sdate, 'yyyymmdd') - idays and
1183             to_date(i_sdate, 'yyyymmdd') --29 时间参数
1184             group by GNBID, CELL_ID
1185             having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) >
0 and sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0)) < 2048) b
1186         on a.GNBID = b.GNBID
1187         and a.cell_id = b.cell_id;
1188         commit;
1189
1190     end;
1191
1192
1193     procedure STATS_5g_PHYSITE_FLOW_2M_DAY(i_sdate varchar2, idays number)
is
1194
1195     begin
1196         delete from STATS_5g_PHYSITE_FLOW_2M_DAY where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1197
1198         insert into STATS_5g_PHYSITE_FLOW_2M_DAY
1199         SELECT DISTINCT TO_DATE(i_sdate, 'yyyymmdd') SDATE,
1200             A.CITY_CODE,
1201             A.CITY_NAME,
1202             A.PHYSTATION_ADDRESS,
1203             A.GNBID,
1204             B.THROUGHPUT
1205         FROM (SELECT DISTINCT CITY_CODE, CITY_NAME, PHYSTATION_ADDRESS,
GNBID
1206             FROM SEQ_5G_SITEINFO

```

```

1207         WHERE SDATE = (SELECT MAX(SDATE) FROM SEQ_5G_SITEINFO)) A
1208     INNER JOIN (SELECT D.CITY_NAME,
1209                 D.GNBID,
1210                 D.PHYSTATION_ADDRESS,
1211                 sum(THROUGHPUT) THROUGHPUT
1212                 FROM (select gnbid,
1213                         cell_id,
1214                         SUM(NVL(UL_THROUGHPUT_M +
DL_THROUGHPUT_M, 0)) THROUGHPUT
1215                         from SDR_FLOW_CELL_THROUGHPUT
1216                         where SDATE BETWEEN
1217                               TO_DATE(i_sdate, 'yyyymmdd') - idays AND
1218                               TO_DATE(i_sdate, 'yyyymmdd')
1219                         group by gnbid, cell_id) T,
1220                 (SELECT distinct CITY_NAME,
1221                         PHYSTATION_ADDRESS,
1222                         gnbid,
1223                         cell_id
1224                 FROM SEQ_5G_SITEINFO
1225                 where SDATE =
1226                 (SELECT MAX(SDATE) FROM
SEQ_5G_SITEINFO)) D
1227         WHERE T.GNBID = D.GNBID
1228             AND T.CELL_ID = D.CELL_ID
1229             GROUP BY D.CITY_NAME, D.GNBID, D.PHYSTATION_ADDRESS
1230             HAVING sum(THROUGHPUT) > 0) B
1231     ON A.GNBID = B.GNBID
1232     AND A.CITY_NAME = B.CITY_NAME
1233     AND A.PHYSTATION_ADDRESS = B.PHYSTATION_ADDRESS;
1234     commit;
1235
1236     end;
1237
1238     procedure STATS_5g_PHYSITE_FLOW_7d_DAY(i_sdate varchar2, idays number)
1239     is
1240     begin
1241         delete from STATS_5g_PHYSITE_FLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1242         insert into STATS_5g_PHYSITE_FLOW_7d_DAY
1243         select distinct SDATE, CITY_CODE, CITY_NAME, PHYSTATION_ADDRESS,
GNBID
1244         from (select to_date(i_sdate, 'yyyymmdd') sdate,
1245                 a.CITY_CODE,
1246                 a.CITY_NAME,
1247                 a.PHYSTATION_ADDRESS,
1248                 a.GNBID,
1249                 b.GNBID bgnbid
1250                 from (SELECT distinct CITY_CODE,
1251                         CITY_NAME,
1252                         PHYSTATION_ADDRESS,
1253                         GNBID
1254                 FROM seq_5g_siteinfo
1255                 where sdate = (select max(sdate) from
seq_5g_siteinfo)) a
1256                 left join (SELECT d.city_name, d.GNBID,
d.PHYSTATION_ADDRESS, sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0))
THROUGHPUT

```



```

1257         FROM SDR_FLOW_CELL_THROUGHPUT t,
seq_5g_siteinfo d
1258         where t.sdate between
1259             to_date(i_sdate, 'yyyymmdd') - idays and
1260             to_date(i_sdate, 'yyyymmdd') --29 时间参数
1261         and d.sdate = (select max(sdate) from
seq_5g_siteinfo)
1262         and t.gnbid = d.gnbid
1263         and t.cell_id = d.cell_id
1264         group by d.city_name, d.GNBID,
d.PHYSTATION_ADDRESS
1265         having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,
0)) > 0
1266             ) b
1267         on a.gnbid = b.gnbid
1268         and a.city_name = b.city_name
1269         and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS)
1270     where bgnbid is null;
1271     commit;
1272
1273     end;
1274
1275     procedure STATS_5g_PHYSITE_LOFLOW_7d_DAY(i_sdate varchar2, idays
number) is
1276
1277     begin
1278         delete from STATS_5g_PHYSITE_LOFLOW_7d_DAY where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1279         insert into STATS_5g_PHYSITE_LOFLOW_7d_DAY
1280             select to_date(i_sdate, 'yyyymmdd') sdate,
1281                 a.CITY_CODE,
1282                 a.CITY_NAME,
1283                 a.PHYSTATION_ADDRESS,
1284                 a.GNBID,
1285                 sum(b.THROUGHPUT) THROUGHPUT
1286             from (SELECT distinct CITY_CODE,
1287                 CITY_NAME,
1288                 PHYSTATION_ADDRESS,
1289                 GNBID,
1290                 CELL_ID
1291                 FROM seq_5g_siteinfo
1292                 where sdate = (select max(sdate) from seq_5g_siteinfo)) a
1293             inner join (SELECT GNBID,
1294                 CELL_ID,
1295                 sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M, 0))
THROUGHPUT
1296                 FROM SDR_FLOW_CELL_THROUGHPUT
1297                 where sdate between to_date(i_sdate, 'yyyymmdd') -
idays and to_date(i_sdate, 'yyyymmdd') --29 时间参数
1298                 group by GNBID, CELL_ID) b
1299                 on a.gnbid = b.gnbid
1300                 and a.cell_id = b.cell_id
1301             group by to_date(i_sdate, 'yyyymmdd'),
1302                 CITY_CODE,
1303                 CITY_NAME,
1304                 PHYSTATION_ADDRESS,
1305                 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

```

```

1352     delete from STATS_4g_SITE_FLOW_48h_hour where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1353     insert into STATS_4g_SITE_FLOW_48h_hour
1354     select SDATE, CITY_CODE, CITY_NAME, ENBID --, BENBID
1355     from (select to_date(i_sdate, 'yyyymmdd') sdate,
1356             a.CITY_CODE,
1357             a.CITY_NAME,
1358             a.enbid,
1359             b.enbid benbid
1360     from (SELECT distinct CITY_CODE, CITY_NAME, ENBID
1361     FROM seq_4g_siteinfo where sdate = (select
max(sdate) from seq_4g_siteinfo)) a
1362     left join (SELECT enbid
1363     FROM sdr_flow_4g_enb_day
1364     where to_date(sdate, 'yyyymmdd') between
to_date(i_sdate, 'yyyymmdd') - idays and --47/24
1365     to_date(i_sdate, 'yyyymmdd') --29 时间参数
1366     group by enbid
1367     having sum(nvl(THROUGHPUT, 0))>0
1368     ) b
1369     on a.enbid = b.enbid)
1370     where benbid is null;
1371     commit;
1372
1373
1374     end;
1375
1376
1377
1378
1379     procedure STATS_4g_CELL_FLOW_48h_hour(i_sdate varchar2, idays number)
is
1380
1381     begin
1382     delete from STATS_4g_CELL_FLOW_48h_hour where trunc(sdate) =
trunc(to_date(i_sdate, 'yyyymmdd'));
1383     insert into STATS_4g_CELL_FLOW_48h_hour
1384     select SDATE, CITY_CODE, CITY_NAME, ENBID, CELL_ID--, BENBID,
BCELL_ID
1385     from (select trunc(to_date(i_sdate, 'yyyymmdd')) sdate,
1386             a.CITY_CODE,
1387             a.CITY_NAME,
1388             a.enbid,
1389             a.cell_id,
1390             b.enbid benbid,
1391             b.cell_id bcell_id
1392     from (SELECT distinct CITY_CODE, CITY_NAME, ENBID, cell_id
1393     FROM seq_4g_siteinfo where sdate = (select
max(sdate) from seq_4g_siteinfo)) a
1394     left join (SELECT enbid, cell_id
1395     --FROM sdr_flow_4g_cell_hour
1396     FROM sdr_flow_4g_cell_day
1397     where to_date(sdate, 'yyyymmdd') between
to_date(i_sdate, 'yyyymmdd') - idays and --47/24
1398     to_date(i_sdate, 'yyyymmdd') --29 时间参数
1399     group by enbid, cell_id
1400     having sum(nvl(THROUGHPUT, 0))>0
1401     ) 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)
1414     is
1415     begin
1416         delete from STATS_5g_SITE_FLOW_48h_hour where trunc(sdate) =
1417         to_date(i_sdate, 'yyyymmdd');
1418         insert into STATS_5g_SITE_FLOW_48h_hour
1419         select SDATE, CITY_CODE, CITY_NAME, GNBID --, BGNBID
1420         from (select to_date(i_sdate, 'yyyymmdd') sdate,
1421                 a.CITY_CODE,
1422                 a.CITY_NAME,
1423                 a.gnbid,
1424                 b.gnbid bgnbid
1425         from (SELECT distinct CITY_CODE, CITY_NAME, GNBID
1426              FROM seq_5g_siteinfo where sdate = (select
1427              max(sdate) from seq_5g_siteinfo)) a
1428         left join (SELECT gnbid
1429                  --FROM sdr_flow_5g_enb_hour
1430                  FROM SDR_FLOW_CELL_THROUGHPUT
1431                  where sdate between to_date(i_sdate, 'yyyymmdd')
1432                  - idays and --47/24
1433                          to_date(i_sdate, 'yyyymmdd') --29 时间参数
1434                  group by gnbid
1435                  having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,
1436                                0))>0
1437                          ) b
1438         on a.gnbid = b.gnbid)
1439         where bgnbid is null;
1440     commit;
1441     end;
1442
1443     procedure STATS_5g_CELL_FLOW_48h_hour(i_sdate varchar2, idays number)
1444     is
1445     begin
1446         --5G小区:因为没有5G小区小时级粒度, 暂时使用天粒度代替
1447         delete from STATS_5g_CELL_FLOW_48h_hour where trunc(sdate) =
1448         to_date(i_sdate, 'yyyymmdd');
1449         insert into STATS_5g_CELL_FLOW_48h_hour
1450         select SDATE, CITY_CODE, CITY_NAME, GNBID, CELL_ID--, BGNBID,
1451         BCELL_ID
1452         from (select to_date(i_sdate, 'yyyymmdd') sdate,
1453                 a.CITY_CODE,
1454                 a.CITY_NAME,
1455                 a.GNBID,

```

```

1452         a.cell_id,
1453         b.GNBID bgnbid,
1454         b.cell_id bcell_id
1455     from (SELECT distinct CITY_CODE, CITY_NAME, GNBID, CELL_ID
1456           FROM seq_5g_siteinfo where sdate = (select
max(sdate) from seq_5g_siteinfo)) a
1457     left join (SELECT GNBID, CELL_ID
1458               FROM SDR_FLOW_CELL_THROUGHPUT
1459               where sdate between to_date(i_sdate, 'yyyymmdd')
- idays and --47/24
1460               to_date(i_sdate, 'yyyymmdd') --29 时间参数
1461               group by GNBID, CELL_ID
1462               having sum(nvl(UL_THROUGHPUT_M+DL_THROUGHPUT_M,
0))>0
1463           ) b
1464     on a.GNBID = b.GNBID
1465     and a.cell_id = b.cell_id)
1466     where bgnbid is null;
1467     commit;
1468
1469
1470     end;
1471
1472
1473     procedure STATS_4G_ERROR_SITEINFO(i_sdate varchar2 default null) is
1474         istart varchar2(8);
1475     begin
1476         --delete from STATS_4G_ERROR_SITEINFO where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1477
1478         istart := '20210720';
1479         --默认值就是对最新的工参进行计算
1480         if i_sdate is null then
1481             SELECT to_char(max(sdate), 'yyyymmdd') into istart FROM
seq_4g_siteinfo;
1482         else
1483             istart := i_sdate;
1484         end if;
1485         dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
istart=' || istart);
1486         dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
start');
1487
1488         execute immediate 'truncate table STATS_4G_ERROR_SITEINFO';
1489         insert into STATS_4G_ERROR_SITEINFO
1490             (ERROR_CLASS,
1491              SDATE,
1492              CITY_CODE,
1493              CITY_NAME,
1494              DISTRICT_CODE,
1495              DISTRICT_NAME,
1496              NETWORK_NAME,
1497              PHYSTATION_ADDRESS,
1498              BBU_NAME,
1499              ENBID,
1500              CELL_NAME,
1501              CELL_ID,
1502              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,

```

```

1725         BBU_NAME,
1726         ENBID,
1727         CELL_NAME,
1728         CELL_ID,
1729         CGI,
1730         LON,
1731         LAT,
1732         DIRECTION,
1733         HEIGHT,
1734         M_DOWNTILT,
1735         E_DOWNTILT,
1736         STATION_TYPE,
1737         ISDIGITALINDOOR,
1738         DOWN_FREQ,
1739         VENDER,
1740         OWN_SCHOOLYARD,
1741         TOWERADDRESS_CODE,
1742         PROPERTY,
1743         SCENE,
1744         IS_SCENESITE,
1745         MARKETING_NETWORK,
1746         TERMINALAMOUNT_5G,
1747         SECTOR_INCOMING,
1748         IS_BUSY,
1749         IS_ALIVE,
1750         IS_ALIVE_UPDATE_TIME,
1751         CONSTRUCTION,
1752         IS_SAME_ADDRESS,
1753         SAME_ADDRESS_SITES,
1754         IS_SAME_ADDRESS_CT
1755     from seq_4g_siteinfo t,
1756     --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。
1757         (SELECT enbid      enbidh,
1758             cell_id  cell_idh,
1759             CELL_NAME CELL_NAMEh,
1760             lon      lonh,
1761             lat      lath,
1762             direction directionh
1763         FROM seq_4g_siteinfo
1764         where SDATE = TO_DATE(istart, 'YYYYMMDD')
1765         and CONSTRUCTION like '%联通%'
1766         group by enbid, cell_id, lon, lat, direction, CELL_NAME
1767         having count(*) > 1) h
1768     where t.enbid = h.enbidh
1769         AND T.cell_id = H.cell_idh
1770         AND T.CELL_NAME = H.CELL_NAMEh
1771         AND T.lon = H.lonh
1772         AND T.lat = H.lath
1773         AND T.direction = H.directionh
1774         AND T.SDATE = TO_DATE(istart, 'YYYYMMDD')
1775         and t.CONSTRUCTION like '%联通%'
1776         and t.is_alive=1
1777
1778     UNION all
1779
1780     --暂时过滤掉室分,如校园站能误判
1781     select distinct '物理站相同但宏站小区距离大于2000米' ERROR_CLASS,

```

```

1782         SDATE,
1783         CITY_CODE,
1784         CITY_NAME,
1785         DISTRICT_CODE,
1786         DISTRICT_NAME,
1787         NETWORK_NAME,
1788         PHYSTATION_ADDRESS,
1789         BBU_NAME,
1790         ENBID,
1791         CELL_NAME,
1792         CELL_ID,
1793         CGI,
1794         LON,
1795         LAT,
1796         DIRECTION,
1797         HEIGHT,
1798         M_DOWNTILT,
1799         E_DOWNTILT,
1800         STATION_TYPE,
1801         ISDIGITALINDOOR,
1802         DOWN_FREQ,
1803         VENDER,
1804         OWN_SCHOOLYARD,
1805         TOWERADDRESS_CODE,
1806         PROPERTY,
1807         SCENE,
1808         IS_SCENESITE,
1809         MARKETING_NETWORK,
1810         TERMINALAMOUNT_5G,
1811         SECTOR_INCOMING,
1812         IS_BUSY,
1813         IS_ALIVE,
1814         IS_ALIVE_UPDATE_TIME,
1815         CONSTRUCTION,
1816         IS_SAME_ADDRESS,
1817         SAME_ADDRESS_SITES,
1818         IS_SAME_ADDRESS_CT
1819     from seq_4g_siteinfo s,
1820     --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
1821     (SELECT CITY_NAMEH, PHYSTATION_ADDRESSH, ENBIDH, LONT, LATT
1822     FROM pysite_bg_2000_4g
1823     union
1824     SELECT CITY_NAMEH, PHYSTATION_ADDRESSH, ENBIDH, LONH, LATH
1825     FROM pysite_bg_2000_4g) k
1826     where s.city_name = k.city_nameh
1827     and s.PHYSTATION_ADDRESS = k.PHYSTATION_ADDRESSH
1828     and s.enbid = k.enbidh
1829     and s.CONSTRUCTION like '%联通%'
1830     and s.SDATE = TO_DATE(istart, 'YYYYMMDD')
1831     and s.is_alive=1
1832
1833     union all
1834
1835     SELECT '工参关键字段为空' ERROR_CLASS,
1836         SDATE,
1837         CITY_CODE,
1838         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
1883
1884     or
1885     STATION_TYPE is null or DOWN_FREQ is null or VENDER is
1886     null)
1887           and m.CONSTRUCTION like '%联通%'
1888           and m.is_alive=1;
1889
1890     COMMIT;
1891     dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
1892     end');
1893
1894     end;
1895
1896     procedure STATS_5G_ERROR_SITEINFO(i_sdate varchar2 default null) is
1897         istart varchar2(8);

```

```

1894     begin
1895         --delete from STATS_4G_ERROR_SITEINFO where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1896
1897         istart := '20210720';
1898         --默认值就是对最新的工参进行计算
1899         if i_sdate is null then
1900             SELECT to_char(max(sdate), 'yyyymmdd')
1901                 into istart
1902                 FROM seq_5g_siteinfo;
1903         else
1904             istart := i_sdate;
1905         end if;
1906
1907         dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
1908         istart=' || istart);
1909         dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
1910         start');
1911
1912         -- 5g基站信息为空统计
1913         --delete from stats_5G_SITEINFO_ISNULL where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
1914         execute immediate 'truncate table STATS_5G_ERROR_SITEINFO';
1915         insert into STATS_5G_ERROR_SITEINFO
1916             (ERROR_CLASS,
1917             SDATE,
1918             CITY_CODE,
1919             CITY_NAME,
1920             DISTRICT_CODE,
1921             DISTRICT_NAME,
1922             NETWORK_NAME,
1923             PHYSTATION_ADDRESS,
1924             STATION_NAME,
1925             GNBID,
1926             CELL_NAME,
1927             CELL_ID,
1928             LON,
1929             LAT,
1930             DIRECTION,
1931             HEIGHT,
1932             M_DOWNTILT,
1933             E_DOWNTILT,
1934             STATION_TYPE,
1935             ISDIGITALINDOOR,
1936             DOWN_FREQ,
1937             VENDER,
1938             OWN_SCHOOLYARD,
1939             TOWERADDRESS_CODE,
1940             PROPERTY,
1941             SCENE,
1942             IS_SCENESITE,
1943             MARKETING_NETWORK,
1944             TERMINALAMOUNT_5G,
1945             SECTOR_INCOMING,
1946             IS_BUSY,
1947             IS_ALIVE,
1948             IS_ALIVE_UPDATE_TIME,
1949             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,

```

```

2006         VENDER,
2007         OWN_SCHOOLYARD,
2008         TOWERADDRESS_CODE,
2009         PROPERTY,
2010         SCENE,
2011         IS_SCENESITE,
2012         MARKETING_NETWORK,
2013         TERMINALAMOUNT_5G,
2014         SECTOR_INCOMING,
2015         IS_BUSY,
2016         IS_ALIVE,
2017         IS_ALIVE_UPDATE_TIME,
2018         CONSTRUCTION,
2019         IS_SAME_ADDRESS,
2020         SAME_ADDRESS_SITES,
2021         IS_SAME_ADDRESS_CT
2022     from seq_5g_siteinfo t,
2023     -- 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
2024         (select GNBID      GNBIDh,
2025             CELL_ID      CELL_IDh,
2026             LON          LONh,
2027             LAT          LATH,
2028             DIRECTION    DIRECTIONh
2029         from (SELECT distinct PHYSTATION_ADDRESS,
2030             GNBID,
2031             CELL_ID,
2032             LON,
2033             LAT,
2034             DIRECTION
2035             FROM seq_5g_siteinfo
2036             where SDATE = TO_DATE(istart, 'YYYYMMDD')
2037             and CONSTRUCTION like '%联通%')
2038         group by GNBID, CELL_ID, LON, LAT, DIRECTION
2039         having count(*) > 1) h
2040     where t.gnbid = h.GNBIDh
2041         and t.cell_id = h.CELL_IDh
2042         and t.LON = h.LONh
2043         and t.LAT = h.LATH
2044         and t.DIRECTION = h.DIRECTIONh
2045         AND T.SDATE = TO_DATE(istart, 'YYYYMMDD')
2046         and T.CONSTRUCTION like '%联通%'
2047         and t.is_alive = 1
2048
2049     union all
2050
2051     select '[基站ID-物理站名]-在多个地市或行政区重复' ERROR_CLASS,
2052         SDATE,
2053         CITY_CODE,
2054         CITY_NAME,
2055         DISTRICT_CODE,
2056         DISTRICT_NAME,
2057         NETWORK_NAME,
2058         PHYSTATION_ADDRESS,
2059         STATION_NAME,
2060         GNBID,
2061         CELL_NAME,
2062         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,

```

```

2236             HEIGHT,
2237             M_DOWNTILT,
2238             E_DOWNTILT,
2239             STATION_TYPE,
2240             ISDIGITALINDOOR,
2241             DOWN_FREQ,
2242             VENDER,
2243             OWN_SCHOOLYARD,
2244             TOWERADDRESS_CODE,
2245             PROPERTY,
2246             SCENE,
2247             IS_SCENESITE,
2248             MARKETING_NETWORK,
2249             TERMINALAMOUNT_5G,
2250             SECTOR_INCOMING,
2251             IS_BUSY,
2252             IS_ALIVE,
2253             IS_ALIVE_UPDATE_TIME,
2254             CONSTRUCTION,
2255             IS_SAME_ADDRESS,
2256             SAME_ADDRESS_SITES,
2257             IS_SAME_ADDRESS_CT
2258         from seq_5g_siteinfo s,
2259         --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
2260         (SELECT city_nameh, PHYSTATION_ADDRESSh, gnbidh, lont, latt
2261             FROM pysite_bg_2000_5g
2262             union
2263             SELECT city_nameh, PHYSTATION_ADDRESSh, gnbidh, lonh, lath
2264             FROM pysite_bg_2000_5g) k
2265     where s.city_name = k.city_nameh
2266           and s.PHYSTATION_ADDRESS = k.PHYSTATION_ADDRESSh
2267           and s.gnbid = k.gnbidh
2268           and s.CONSTRUCTION like '%联通%'
2269           and s.sdate=to_date(istart,'yyyymmdd')
2270           and s.is_alive = 1
2271
2272     union all
2273
2274     SELECT '工参关键字段为空' ERROR_CLASS,
2275            SDATE,
2276            CITY_CODE,
2277            CITY_NAME,
2278            DISTRICT_CODE,
2279            DISTRICT_NAME,
2280            NETWORK_NAME,
2281            PHYSTATION_ADDRESS,
2282            STATION_NAME,
2283            GNBID,
2284            CELL_NAME,
2285            CELL_ID,
2286            LON,
2287            LAT,
2288            DIRECTION,
2289            HEIGHT,
2290            M_DOWNTILT,
2291            E_DOWNTILT,
2292            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,

```

```

2460         TOWERADDRESS_CODE,
2461         PROPERTY,
2462         SCENE,
2463         IS_SCENESITE,
2464         MARKETING_NETWORK,
2465         TERMINALAMOUNT_5G,
2466         SECTOR_INCOMING,
2467         IS_BUSY,
2468         IS_ALIVE,
2469         IS_ALIVE_UPDATE_TIME,
2470         CONSTRUCTION,
2471         IS_SAME_ADDRESS,
2472         SAME_ADDRESS_SITES,
2473         IS_SAME_ADDRESS_CT
2474     FROM SEQ_4G_SITEINFO_GEO A, SEQ_CFG_COLLEGES_GEO B
2475     WHERE SDO_WITHIN_DISTANCE(A.GEOM,
2476                               B.GEOM,
2477                               'distance=' || indoor_range || '
unit=m') =
2478         'TRUE'
2479     AND A.CITY_NAME = B.CITY_NAME
2480     and STATION_TYPE = '室内';
2481     commit;
2482 end;
2483
2484
2485 procedure STATS_5G_COLLEGES_SITEINFO(i_sdate      varchar2,
2486                                       outdoor_range number,
2487                                       indoor_range  number) is
2488
2489     begin
2490
2491         dropindex_ifexists('I_SEQ_5G_SITEINFO_GEO');
2492         execute immediate 'truncate table SEQ_5G_SITEINFO_GEO';
2493         --delete from SEQ_5G_SITEINFO_GEO where trunc(sdate) =
to_date(i_sdate, 'yyyymmdd');
2494         insert into SEQ_5G_SITEINFO_GEO
2495             SELECT SDATE,
2496                    CITY_CODE,
2497                    CITY_NAME,
2498                    DISTRICT_CODE,
2499                    DISTRICT_NAME,
2500                    NETWORK_NAME,
2501                    PHYSTATION_ADDRESS,
2502                    STATION_NAME,
2503                    GNBID,
2504                    CELL_NAME,
2505                    CELL_ID,
2506                    LON,
2507                    LAT,
2508                    DIRECTION,
2509                    HEIGHT,
2510                    M_DOWNTILT,
2511                    E_DOWNTILT,
2512                    STATION_TYPE,
2513                    ISDIGITALINDOOR,
2514                    DOWN_FREQ,
2515                    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','GEO');
2541
2542     execute immediate 'create index i_SEQ_5G_SITEINFO_GEO on
2543     SEQ_5G_SITEINFO_GEO(GEOM) indextype is MDSYS.SPATIAL_INDEX
2544     parameters(''layer_gtype=point'')';
2545
2546     execute immediate 'truncate table STATS_5G_COLLEGES_SITEINFO';
2547     insert into STATS_5G_COLLEGES_SITEINFO
2548     SELECT B.COLLEGE_NAME,
2549            SDATE,
2550            CITY_CODE,
2551            a.CITY_NAME,
2552            DISTRICT_CODE,
2553            DISTRICT_NAME,
2554            NETWORK_NAME,
2555            PHYSTATION_ADDRESS,
2556            STATION_NAME,
2557            GNBID,
2558            CELL_NAME,
2559            CELL_ID,
2560            LON,
2561            LAT,
2562            DIRECTION,
2563            HEIGHT,
2564            M_DOWNTILT,
2565            E_DOWNTILT,
2566            STATION_TYPE,
2567            ISDIGITALINDOOR,
2568            DOWN_FREQ,
2569            VENDER,
2570            OWN_SCHOOLYARD,
2571            TOWERADDRESS_CODE,
2572            PROPERTY,
2573            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

```

```

2679         and a.sdate = b.sdate
2680         and a.sdate = to_date(istart,'yyyymmdd')) t
2681     group by SDATE, CITY, ENODEB, SC_ECI),
2682 --commit;
2683 --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
2684 --带avg的为算数平均，未加权
2685 --execute immediate 'truncate table fdd_MDT_ENB_position';
2686 --insert into fdd_MDT_ENB_position
2687 fdd_MDT_ENB_position as (
2688     select a.sdate,
2689           city,
2690           enodeb,
2691           avg(a.lon) lon,
2692           avg(a.lat) lat,
2693           --注释掉绝对平均算法20210816
2694           --avg(a.avg_lon) avg_lon,
2695           --avg(a.avg_lat) avg_lat,
2696           PHYSTATION_ADDRESS
2697     from fdd_MDT_cell_position a, seq_4g_siteinfo b
2698     where a.enodeb = b.enbid
2699           and mod(a.sc_eci, 256) = b.cell_id
2700           and b.sdate = (select max(sdate) from seq_4g_siteinfo)
2701     group by a.sdate, city, enodeb, PHYSTATION_ADDRESS),
2702 --commit;
2703 --关联工参数据，计算小区级偏离距离
2704 --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。
2705 --create table fdd_MDT_cell_position_distance as
2706 --execute immediate 'truncate table fdd_MDT_cell_position_distance';
2707 --insert into fdd_MDT_cell_position_distance
2708 fdd_MDT_cell_position_distance as (
2709     SELECT b.lon MDT_LON,
2710           b.lat MDT_LAT,
2711           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
2712           --注释掉绝对平均算法20210816
2713           --B.avg_lon AVG_MDT_LON,
2714           --B.avg_lat AVG_MDT_LAT,
2715           --math_distance(A.LON, A.LAT, B.avg_lon, B.avg_lat)
    avg_deviate_DISTANCE,
2716     b.CITY,
2717     a.CITY_CODE,
2718     a.CITY_NAME,
2719     a.DISTRICT_CODE,
2720     a.DISTRICT_NAME,
2721     a.NETWORK_NAME,
2722     a.PHYSTATION_ADDRESS,
2723     a.BBU_NAME,
2724     a.ENBID,
2725     a.CELL_NAME,
2726     SC_ECI,
2727     a.CELL_ID,
2728     a.CGI,
2729     a.LON,
2730     a.LAT,
2731     a.DIRECTION,
2732     a.HEIGHT,
2733     a.M_DOWNTILT,

```

```

2734         a.E_DWNTILT,
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_q1
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_deviat 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,

```

```

2891         b.scene,
2892         b.BBU_NAME,
2893         b.ENBID,
2894         b.CELL_NAME,
2895         b.CELL_ID,
2896         b.LON,
2897         b.LAT,
2898         a.LON             enb_lon,
2899         a.LAT             enb_lat
2900     FROM PHYSTATION_LON A, SEQ_4G_SITEINFO B
2901     WHERE B.SDATE = (select max(sdate) from SEQ_4G_SITEINFO)
2902           AND A.ENBID = B.ENBID
2903           AND A.PHYSTATION_ADDRESS = B.PHYSTATION_ADDRESS
2904           and b.station_type = '宏站'),
2905     --SELECT * FROM cell_deviat where enbid=166775
2906     cell_deviat3000 as
2907     (select b.*
2908       from (select distinct CITY_NAME, PHYSTATION_ADDRESS, ENBID
2909             from cell_deviat
2910             where enb_distance > 3000) a,
2911            cell_deviat b
2912     where a.city_name = b.city_name
2913           and a.PHYSTATION_ADDRESS = b.PHYSTATION_ADDRESS
2914           and a.enbid = b.enbid),
2915     --SELECT * FROM cell_deviat3000 where enbid=166775
2916     site_deviat3000 as
2917     (select a.*, math_distance(a.lon, a.lat, b.lon, b.lat)
2918     cell_distance
2919       from cell_deviat3000 a, CELL_LON b
2920     where a.city = b.city
2921           and a.enbid = b.enodeb
2922           and a.cell_id = mod(b.sc_eci, 256))
2923     --select * from site_deviat3000 where enbid=166775
2924     select t2.*
2925       from (select CITY_NAME,
2926                DISTRICT_NAME,
2927                CITY_CODE,
2928                DISTRICT_CODE,
2929                PHYSTATION_ADDRESS,
2930                BBU_NAME,
2931                ENBID,
2932                --LON,
2933                --LAT,
2934                STATION_TYPE,
2935                VENDER,
2936                --SCENE,
2937                first_value(SCENE) over(partition by CITY_NAME,
2938     DISTRICT_NAME, PHYSTATION_ADDRESS, BBU_NAME, ENBID order by cell_DISTANCE
2939     desc) SCENE,
2940                enb_DISTANCE,
2941                ENBID * 256 + cell_id cell_id,
2942                CELL_NAME,
2943                --CELL_ID,
2944                cell_DISTANCE,
2945                Dense_rank() over(partition by CITY_NAME,
2946     DISTRICT_NAME, PHYSTATION_ADDRESS, BBU_NAME, ENBID order by cell_DISTANCE
2947     desc) rk

```

```

2944         from site_deviate3000 a)
2945     pivot (max(cell_id) ECI, max(cell_name) CELL_NAME,
2946           max(cell_DISTANCE) DISTANCE for rk in('1',
2947           '2',
2948           '3',
2949           '4',
2950           '5',
2951           '6',
2952           '7',
2953           '8',
2954           '9')) t2;
2955
2956     commit;
2957
2958     end;
2959
2960     --更新有业务量无工参统计
2961
2962     procedure STATS_HAS_FLOW_NO_4G_GC(i_sdate varchar2 default null) is
2963         istart varchar2(8);
2964         iend varchar2(8);
2965     begin
2966         --delete from STATS_4G_ERROR_SITEINFO where trunc(sdate) =
2967         to_date(i_sdate, 'yyyymmdd');
2968         if i_sdate is null then
2969             select min(sdate) into istart
2970             from (select max(sdate) sdate
2971                 from sdr_flow_4g_cell_day
2972                 union
2973                 select max(sdate) sdate
2974                 from sdr_flow_4g_enb_day
2975                 union
2976                 select to_char(sysdate, 'yyyymmdd')
2977                 from dual);
2978             select to_char(to_date(istart, 'yyyymmdd')-6, 'yyyymmdd') into iend
2979             from dual;
2980         else
2981             select to_char(to_date(i_sdate, 'yyyymmdd')-6, 'yyyymmdd') into iend
2982             from dual;
2983         end if;
2984
2985         dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
istart=' || istart || ', iend=' || iend);
2986         dbms_output.put_line(to_char(sysdate, 'yyyy-mm-dd hh24:MI:ss') || ' :
start');
2987
2988         execute immediate 'truncate table STATS_HAS_FLOW_NO_4G_GC';
2989         insert into STATS_HAS_FLOW_NO_4G_GC
2990         SELECT iend || '-' || istart "数据时间",
2991             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
to_date(istart, 'yyyymmdd')
3131     group by GNBID, CELL_ID) a
3132     left join (select GNBID,
3133             CELL_ID,
3134             max(CITY_name) CITY_name,
3135             max(cell_name) cell_name,
3136             max(CONSTRUCTION) CONSTRUCTION
3137     from seq_5g_siteinfo
3138     where sdate = (select max(sdate) from
seq_5g_siteinfo)
3139     group by GNBID, CELL_ID) b
3140

```

```

3141         on (a.gnbid = b.gnbid and a.cell_id = b.cell_id)
3142         left join (SELECT distinct gnbid,cell_id FROM
STATS_5G_ERROR_SITEINFO) c
3143         on (a.gnbid = c.gnbid and a.cell_id = c.cell_id)
3144         group by a.GNBID, a.CELL_ID);
3145     commit;
3146     execute immediate 'truncate table STATS_5G_ERR_FLAG';
3147     insert into STATS_5G_ERR_FLAG
3148     with a_2d as (
3149     select distinct a.gnbid,a.cell_id
3150     from seq_5g_siteinfo a,STATS_5g_CELL_FLOW_7d_DAY b
3151     where a.sdate = (select max(sdate) from seq_5g_siteinfo)
3152     and a.is_alive = 1
3153     and b.sdate = (select max(sdate) from
STATS_5g_CELL_FLOW_7d_DAY)
3154     and a.gnbid = b.gnbid
3155     and a.cell_id = b.cell_id),
3156     b_err as (
3157     SELECT distinct gnbid, CELL_ID
3158     FROM STATS_5G_ERROR_SITEINFO
3159     where sdate = (select max(sdate) from STATS_5G_ERROR_SITEINFO)
3160     and is_alive = 1)
3161
3162     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
3163     union all
3164     select '5G小区2天无业务', t.* from (select * from a_2d MINUS select
* from b_err) t
3165     union all
3166     select '工参不全不准', t.* from (select * from b_err MINUS select *
from a_2d) t;
3167     commit;
3168
3169     dbms_output.put_line(to_char(sysdate,'yyyy-mm-dd hh24:MI:ss') || ' :
start');
3170     end;
3171 end pkg_basedata_governance_v4;
3172
3173

```

8、课题联系人信息

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

9、资料备份

- 1 目前所有资料都存放于D:\gongcanzhili\02_gczi目录。
- 2 |—01业务量备份
- 3 |—02工参数据
- 4 |—03报表导出-刘子龙
- 5 |—04plsql脚本备份
- 6 |—05其它备份

7 | | └─偏离度聚类备份
8 | | └─06基站偏离度
9 | | └─07软件程序
10 | | └─08接口上传
11 | | └─09工参资源大屏治理_备份
12 | | | └─业务量
13 | | | └─作战地图工参
14 | | | └─华为SEQ业务量数据及字段说明
15 | | | └─基站偏离统计
16 | | | └─外部输出
17 | | | | └─4G有业务量但无工参
18 | | | | └─5G有业务量但无工参
19 | | | └─大屏原型与报告
20 | | | └─工作总结
21 | | | └─工参治理系统设计-初版
22 | | | └─接口规范
23 | | | └─网络需求工单
24 | | | └─表结构说明
25 | | └─10接口程序源码
26 | | | └─oracle_2_esb
27 | | | └─oracle_2_excel