# 函数

## 例子

|  |
| --- |
| create or replace FUNCTION "GET\_VIRTUALDEVICENAME" (s\_id in number,d\_type in number)  RETURN VARCHAR2 AS  ret varchar2(200);  BEGIN  begin  select d.device\_name into ret from t\_device d where d.station\_id = s\_id and d.device\_type=d\_type and d.device\_name like '%虚拟%';  end;  RETURN ret;  END; |

## 例子2

|  |
| --- |
| create or replace FUNCTION "ISEXIST\_PAR2" (  tablename in varchar2,  dt in varchar2,  objtype in varchar2  )  return number  as  ret number:=0;  begin  select count(\*) into ret from user\_objects  where object\_name=tablename and subobject\_name=tablename||'\_P'||dt and object\_type=objtype;  return ret;  end; |

## 例子3

|  |
| --- |
| create or replace FUNCTION "MUL\_MULALARMLEVEL"  (alarmtypeids in varchar2)  RETURN VARCHAR2 AS  ret varchar2(1000);  pos number;  alarmtypeid\_tmp number;  BEGIN  pos := instr(alarmtypeids, ',');  if pos > 0 then  alarmtypeid\_tmp := to\_number(substr(alarmtypeids, 1, pos - 1));  ret := get\_all\_severity(alarmtypeid\_tmp);    ret := ret || ',' || MUL\_MULALARMLEVEL(substr(alarmtypeids, pos + 1));  else  alarmtypeid\_tmp := to\_number(alarmtypeids);  ret := get\_all\_severity(alarmtypeid\_tmp);  end if;  return ret;  END MUL\_MULALARMLEVEL; |

# 过程

## 例子1

|  |
| --- |
| create or replace procedure "ADDTO\_T\_USERSTATUS\_H" is  numq number;  begin  for numq in 5..34 loop  insert into t\_user\_status\_h  select /\*+ parallel(t\_user\_status,8) \*/  user\_id,max(portname),max(accessdevice\_name),max(LOG\_TIME),trunc(sysdate,'hh')-numq/24 ,max(ERROR\_TAG),max(QOE),  sum(MULTICAST\_TIME),sum(VODCAST\_TIME),sum(HD\_MULTICAST\_TIME),sum(hd\_vodcast\_time),  sum(AUTHNUMBERS),sum(AUTHFAILNUMBERS),sum(MULTIREQNUMBERS),sum(MULTIFAILNUMBERS),  sum(VODREQNUMBERS),sum(VODFAILNUMBERS),sum(httpreqnumbers),sum(httpfailnumbers),  decode(sign(sum(AUTHNUMBERS)-0),1,sum(AUTHFAILNUMBERS)/sum(AUTHNUMBERS),null)\*100,  decode(sign(sum(MULTIREQNUMBERS)-0),1,sum(MULTIFAILNUMBERS)/sum(MULTIREQNUMBERS),null)\*100,  decode(sign(sum(VODREQNUMBERS)-0),1,sum(VODFAILNUMBERS)/sum(VODREQNUMBERS),null)\*100,  decode(sign(sum(httpreqnumbers)-0),1,sum(httpfailnumbers)/sum(httpreqnumbers),null)\*100,  decode(sign(sum(MULTIREQNUMBERS)-0),1,sum(MULTI\_RRT\*MULTIREQNUMBERS)/sum(MULTIREQNUMBERS),null),  decode(sign(sum(VODREQNUMBERS)-0),1,sum(VOD\_RRT\*VODREQNUMBERS)/sum(VODREQNUMBERS),null),  null,  sum(multiabendnumbers),sum(vodabendnumbers),sum(multiupabendnumbers),sum(vodupabendnumbers),  sum(hd\_multiabendnumbers),sum(hd\_vodabendnumbers),sum(hd\_multiupabendnumbers),sum(hd\_vodupabendnumbers),  sum(playerror\_numbers),  decode(sign(sum(MULTICAST\_TIME)-0),1,sum(MULTICAST\_TIME\*MULTICAST\_MLR\_TIMERATE)/sum(MULTICAST\_TIME),null),  decode(sign(sum(VODCAST\_TIME)-0),1,sum(VODCAST\_TIME\*VOD\_MLR\_TIMERATE)/sum(VODCAST\_TIME),null),  decode(sign(sum(MULTICAST\_TIME)-0),1,sum(MULTICAST\_TIME\*MULTICAST\_MLR\_TIMERATE\_FEC)/sum(MULTICAST\_TIME),null),  decode(sign(sum(VODCAST\_TIME)-0),1,sum(VODCAST\_TIME\*VOD\_MLR\_TIMERATE\_FEC)/sum(VODCAST\_TIME),null),  decode(sign(sum(HD\_MULTICAST\_TIME)-0),1,sum(HD\_MULTICAST\_TIME\*HD\_MULTICAST\_MLR\_TIMERATE)/sum(HD\_MULTICAST\_TIME),null),  decode(sign(sum(hd\_vodcast\_time)-0),1,sum(hd\_vodcast\_time\*HD\_VOD\_MLR\_TIMERATE)/sum(hd\_vodcast\_time),null) ,  decode(sign(sum(HD\_MULTICAST\_TIME)-0),1,sum(HD\_MULTICAST\_TIME\*HD\_MULTICAST\_MLR\_TIMERATE\_FEC)/sum(HD\_MULTICAST\_TIME),null),  decode(sign(sum(hd\_vodcast\_time)-0),1,sum(hd\_vodcast\_time\*HD\_VOD\_MLR\_TIMERATE\_FEC)/sum(hd\_vodcast\_time),null),  decode(sign(sum(MULTICAST\_TIME)-0),1,sum(MULTICAST\_TIME\*MULTI\_LOW\_STREAMRATE\_TIMERATE)/sum(MULTICAST\_TIME),null),  decode(sign(sum(VODCAST\_TIME)-0),1,sum(VODCAST\_TIME\*VOD\_LOW\_STREAMRATE\_TIMERATE)/sum(VODCAST\_TIME),null),  decode(sign(sum(HD\_MULTICAST\_TIME)-0),1,sum(HD\_MULTICAST\_TIME\*HD\_MULTI\_LOW\_STRRATE\_TIMERATE)/sum(HD\_MULTICAST\_TIME),null),  decode(sign(sum(hd\_vodcast\_time)-0),1,sum(hd\_vodcast\_time\*HD\_VOD\_LOW\_STREAMRATE\_TIMERATE)/sum(hd\_vodcast\_time),null),  sum(BUFFERFECNUM),  0,  null,null,null,null,null,null,  null,null,null,null,null,null,  null,null,null,null,null,null,  null,null,null,null,null,null,  null,null,null,null,null,null,  null,null,null,null,null,null,  max(POP\_NAME),  null,null,null,null,  sum(TOTAL\_MULTI\_MLR\_TIME),  sum(TOTAL\_MULTI\_BITRATE\_TIME),  sum(TOTAL\_VOD\_MLR\_TIME),  sum(TOTAL\_VOD\_BITRATE\_TIME),  decode(sign((sum(MULTICAST\_TIME)+sum(HD\_MULTICAST\_TIME))-0),1,sum(TOTAL\_MULTI\_MLR\_TIME)/(sum(MULTICAST\_TIME)+sum(HD\_MULTICAST\_TIME)),null)\*100,  decode(sign((sum(MULTICAST\_TIME)+sum(HD\_MULTICAST\_TIME))-0),1,sum(TOTAL\_MULTI\_BITRATE\_TIME)/(sum(MULTICAST\_TIME)+sum(HD\_MULTICAST\_TIME)),null)\*100,  decode(sign((sum(VODCAST\_TIME)+sum(hd\_vodcast\_time))-0),1,sum(TOTAL\_VOD\_MLR\_TIME)/(sum(VODCAST\_TIME)+sum(hd\_vodcast\_time)),null)\*100,  decode(sign((sum(VODCAST\_TIME)+sum(hd\_vodcast\_time))-0),1,sum(TOTAL\_VOD\_BITRATE\_TIME)/(sum(VODCAST\_TIME)+sum(hd\_vodcast\_time)),null)\*100,  count(\*),count(case when ERROR\_TAG>0 then 1 end),max(IS4K),max(STATISTICS\_FLAG),max(MAXCPU),avg(AVGCPU),max(MAXMEMORY),avg(AVGMEMORY),max(TOTALFLASH),min(FREEFLASH)  from t\_user\_status  where process\_time>=trunc(sysdate,'hh')-(numq+1)/24  and process\_time<trunc(sysdate,'hh')-numq/24  group by user\_id;  end loop;  end; |

## 例子2

|  |
| --- |
| create or replace PROCEDURE "DEVICE\_CHART\_FAILUSERS" (  v\_begin\_date in varchar2,  v\_end\_date in varchar2,  n\_device\_type in number,  n\_index\_type in number,  c\_device\_faildetail OUT sys\_refcursor,  n\_ret out number  )  AS  BEGIN  declare  devicetype number;  accesstype number;  begin  if n\_device\_type=0 then  devicetype:=5;  else devicetype:=0 ;  end if;  if n\_device\_type=3 then  accesstype:=3;  else accesstype:=10 ;  end if;    if n\_index\_type=1 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(AUTH\_FAILURE\_NUM\_2+AUTH\_FAILURE\_NUM\_1) as total,sum(AUTH\_FAILURE\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;  elsif n\_index\_type=2 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(MULTICAST\_REQ\_NUM\_2+MULTICAST\_REQ\_NUM\_1) as total,sum(MULTICAST\_REQ\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;    elsif n\_index\_type=3 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(VOD\_REQ\_NUM\_2+VOD\_REQ\_NUM\_1) as total,sum(VOD\_REQ\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;  elsif n\_index\_type=4 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(EPG\_REQ\_NUM\_2+EPG\_REQ\_NUM\_1) as total,sum(EPG\_REQ\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;    elsif n\_index\_type=5 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(MULTICAST\_RRT\_NUM\_2+MULTICAST\_RRT\_NUM\_1) as total,sum(MULTICAST\_RRT\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;      elsif n\_index\_type=6 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(VOD\_RRT\_NUM\_2+VOD\_RRT\_NUM\_1) as total,sum(VOD\_RRT\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;  elsif n\_index\_type=7 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(PLAYERROR\_NUM\_2+PLAYERROR\_NUM\_1) as total,sum(PLAYERROR\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;  elsif n\_index\_type=8 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(TOTAL\_MULTI\_MLR\_NUM\_2+TOTAL\_MULTI\_MLR\_NUM\_1) as total,sum(TOTAL\_MULTI\_MLR\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;  elsif n\_index\_type=9 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(TOTAL\_VOD\_MLR\_NUM\_2+TOTAL\_VOD\_MLR\_NUM\_1) as total,sum(TOTAL\_VOD\_MLR\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;    elsif n\_index\_type=10 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(TOTAL\_MULTI\_BITRATE\_NUM\_2+TOTAL\_MULTI\_BITRATE\_NUM\_1) as total,sum(TOTAL\_MULTI\_BITRATE\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;  elsif n\_index\_type=11 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(TOTAL\_VOD\_BITRATE\_NUM\_2+TOTAL\_VOD\_BITRATE\_NUM\_1) as total,sum(TOTAL\_VOD\_BITRATE\_NUM\_2) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;  elsif n\_index\_type=0 then  open c\_device\_faildetail for  /\*+ parallel(t\_device\_status ,8) \*/  select trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi'),  sum(user\_num) as total,sum(error\_num) as fail  from t\_device\_status  where stat\_time>=to\_date(v\_begin\_date,'yyyy-mm-dd hh24:mi:ss')  and stat\_time<to\_date(v\_end\_date,'yyyy-mm-dd hh24:mi:ss')  -- and (device\_type=n\_device\_type or device\_type<devicetype or device\_type>=accesstype)  and device\_type=1  group by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi')  order by trunc(stat\_time-mod(to\_char(stat\_time,'mi'),5)/1440,'mi');  n\_ret:=0;  end if;  end;  END DEVICE\_CHART\_FAILUSERS; |

## 例子3

|  |
| --- |
| create or replace PROCEDURE "GLOBAL\_EVALUATION\_KPI\_DIMEN" (  dimen\_type in number,  index\_type in number,  begin\_date in varchar2,  end\_date in varchar2,  g\_evluation\_kpi\_dimen OUT sys\_refcursor  )  AS  v\_err\_code NUMBER;  v\_err\_msg VARCHAR2(200);  v\_column\_num2 varchar2(200);  v\_column\_fail\_rate varchar2(300);  v\_sql varchar2(4000);  BEGIN  if index\_type=1 then  v\_column\_num2:='AUTH\_FAILURE\_NUM\_2';  v\_column\_fail\_rate:='round(sum(AUTH\_FAILURE\_NUM\_2) /sum(AUTH\_FAILURE\_NUM\_1+AUTH\_FAILURE\_NUM\_2),6)';  elsif index\_type=2 then  v\_column\_num2:='MULTICAST\_REQ\_NUM\_2';  v\_column\_fail\_rate:='round(sum(MULTICAST\_REQ\_NUM\_2) /sum(MULTICAST\_REQ\_NUM\_1+MULTICAST\_REQ\_NUM\_2),6)';  elsif index\_type=3 then  v\_column\_num2:='VOD\_REQ\_NUM\_2';  v\_column\_fail\_rate:='round(sum(VOD\_REQ\_NUM\_2) /sum(VOD\_REQ\_NUM\_1+VOD\_REQ\_NUM\_2),6)';  elsif index\_type=4 then  v\_column\_num2:='EPG\_REQ\_NUM\_2';  v\_column\_fail\_rate:='round(sum(EPG\_REQ\_NUM\_2) /sum(EPG\_REQ\_NUM\_1+EPG\_REQ\_NUM\_2),6)';  elsif index\_type=5 then  v\_column\_num2:='MULTICAST\_RRT\_NUM\_2';  v\_column\_fail\_rate:='round(sum(MULTICAST\_RRT\_NUM\_2) /sum(EPG\_REQ\_NUM\_1+EPG\_REQ\_NUM\_2),6)';  elsif index\_type=6 then  v\_column\_num2:='VOD\_RRT\_NUM\_2';  v\_column\_fail\_rate:='round(sum(VOD\_RRT\_NUM\_2) /sum(VOD\_RRT\_NUM\_1+VOD\_RRT\_NUM\_2),6) ';  elsif index\_type=7 then  v\_column\_num2:='PLAYERROR\_NUM\_2';  v\_column\_fail\_rate:='round(sum(PLAYERROR\_NUM\_2) /sum(PLAYERROR\_NUM\_1+PLAYERROR\_NUM\_2),6) ';  elsif index\_type=8 then  v\_column\_num2:='TOTAL\_MULTI\_MLR\_NUM\_2';  v\_column\_fail\_rate:='round(sum(TOTAL\_MULTI\_MLR\_NUM\_2) /sum(TOTAL\_MULTI\_MLR\_NUM\_1+TOTAL\_MULTI\_MLR\_NUM\_2),6)';  elsif index\_type=9 then  v\_column\_num2:='TOTAL\_VOD\_MLR\_NUM\_2';  v\_column\_fail\_rate:='round(sum(TOTAL\_VOD\_MLR\_NUM\_2) /sum(TOTAL\_VOD\_MLR\_NUM\_1+TOTAL\_VOD\_MLR\_NUM\_2),6)';  elsif index\_type=10 then  v\_column\_num2:='TOTAL\_MULTI\_BITRATE\_NUM\_2';  v\_column\_fail\_rate:='round(sum(TOTAL\_MULTI\_BITRATE\_NUM\_2) /sum(TOTAL\_MULTI\_BITRATE\_NUM\_1+TOTAL\_MULTI\_BITRATE\_NUM\_2),6)';  elsif index\_type=11 then  v\_column\_num2:='TOTAL\_VOD\_BITRATE\_NUM\_2';  v\_column\_fail\_rate:='round(sum(TOTAL\_VOD\_BITRATE\_NUM\_2) /sum(TOTAL\_VOD\_BITRATE\_NUM\_1+TOTAL\_VOD\_BITRATE\_NUM\_2),6) ';  elsif index\_type=23 then  v\_column\_num2:='DELAY\_TIME\_NUM\_2';  v\_column\_fail\_rate:='round(sum(DELAY\_TIME\_NUM\_2) /sum(DELAY\_TIME\_NUM\_1+DELAY\_TIME\_NUM\_2),6) ';  elsif index\_type=101 then  v\_column\_num2:='error\_num';  v\_column\_fail\_rate:='round(((sum(AUTH\_FAILURE\_NUMBER)\*100/(sum(AUTH\_NUMBER)+0.000001))+(sum(VOD\_REQ\_FAIL\_NUMBER)\*100/(sum(VOD\_REQ\_NUMBER)+0.00001)))/2,6) ';  elsif index\_type=102 then  v\_column\_num2:='error\_num';  v\_column\_fail\_rate:='round(sum(PLAYERROR\_NUMBER)\*100/(sum(MULTICAST\_REQ\_NUMBER)+sum(VOD\_REQ\_NUMBER)+0.00001),6)';  elsif index\_type=103 then  v\_column\_num2:='error\_num';  v\_column\_fail\_rate:='round((sum(MULTICAST\_REQ\_FAIL\_NUMBER)+sum(VOD\_REQ\_FAIL\_NUMBER))\*100/(sum(MULTICAST\_REQ\_NUMBER)+sum(VOD\_REQ\_NUMBER)+0.00001),6) ';  end if;  declare  g\_evluation\_query\_kpi varchar2(5000);  g\_evluation\_query\_kpi\_total varchar2(5000);  begin  insert into t\_hourly\_log values(sysdate,' GLOBAL\_EVALUATION\_KPI\_DIMEN:global evalation dimen: ','dimen :'||dimen\_type||',index\_type:'||index\_type);  commit;  if dimen\_type=0 then  v\_sql:='select a.\*,rownum from (  select b.\* from (  select  1,stb\_id,stb\_name,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '|| v\_column\_fail\_rate||' as fail\_rate  from t\_stb\_status\_d  where  stb\_class=1  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stb\_id is not null  group by stb\_id,stb\_name  union  select  3,stb\_id,stb\_name,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '|| v\_column\_fail\_rate||' as fail\_rate  from t\_stb\_status\_d  where  stb\_class=3  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stb\_id is not null  group by stb\_id,stb\_name  union  select  4,device\_id,GET\_DEVICENAME(device\_id),  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '|| v\_column\_fail\_rate||' as fail\_rate  from t\_device\_status\_d  where device\_type=1  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and device\_id is not null  group by device\_id  union  select  5,station\_id,station\_name,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '|| v\_column\_fail\_rate||' as fail\_rate  from t\_station\_status\_d  where  stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and station\_id is not null  group by station\_id,station\_name  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  elsif dimen\_type=1 then  v\_sql:='select a.\*,rownum from (  select b.\* from (  select  1,stb\_id,stb\_name,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '||v\_column\_fail\_rate||' as fail\_rate  from t\_stb\_status\_d  where  stb\_class=1  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stb\_id is not null  group by stb\_id,stb\_name  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  elsif dimen\_type=3 then  v\_sql:='select a.\*,rownum from (  select b.\* from (  select  3,stb\_id,stb\_name,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '||v\_column\_fail\_rate||' as fail\_rate  from t\_stb\_status\_d  where  stb\_class=3  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stb\_id is not null  group by stb\_id,stb\_name  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  elsif dimen\_type=4 then  v\_sql:=' select a.\*,rownum from (  select b.\* from (  select  4,station\_id,station\_name,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '||v\_column\_fail\_rate||' as fail\_rate  from t\_station\_status\_d  where stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and station\_id is not null  group by station\_id,station\_name  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  elsif dimen\_type=5 then  v\_sql:='select a.\*,rownum from (  select b.\* from (  select  5,device\_id,GET\_DEVICENAME(device\_id),  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '||v\_column\_fail\_rate||' as fail\_rate  from t\_device\_status\_d  where device\_type=1  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and device\_id is not null  group by device\_id  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  elsif dimen\_type=55 then  v\_sql:='select a.\*,rownum from (  select b.\* from (  select  55,device\_id,GET\_DEVICENAME(device\_id),  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '||v\_column\_fail\_rate||' as fail\_rate  from t\_device\_status\_d  where device\_type=5  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and device\_id is not null  group by device\_id  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  elsif dimen\_type=66 then  v\_sql:='select a.\*,rownum from (  select b.\* from (  select  66,pop\_id,pop\_name,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '||v\_column\_fail\_rate||' as fail\_rate  from T\_POP\_STATUS\_D  where pop\_type=1  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and pop\_id is not null  group by pop\_id,pop\_name  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  elsif dimen\_type=77 then  v\_sql:='select a.\*,rownum from (  select b.\* from (  select  77,pop\_id,pop\_name,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '||v\_column\_fail\_rate||' as fail\_rate  from T\_POP\_STATUS\_D  where pop\_type=2  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and pop\_id is not null  group by pop\_id,pop\_name  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  elsif dimen\_type=88 then  v\_sql:='select a.\*,rownum from (  select b.\* from (  select  88,pop\_id,pop\_name,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '||v\_column\_fail\_rate||' as fail\_rate  from T\_POP\_STATUS\_D  where pop\_type=3  and stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and pop\_id is not null  group by pop\_id,pop\_name  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  elsif dimen\_type=99 then  v\_sql:='select a.\*,rownum from (  select b.\* from (  select  99,EPGGROUP\_ID,EPGGROUP\_NAME,  sum(user\_num) as total,sum('||v\_column\_num2||') as fail,  '||v\_column\_fail\_rate||' as fail\_rate  from T\_EPGGROUP\_STATUS\_D  where stat\_time>=to\_date('''||begin\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and stat\_time<to\_date('''||end\_date||''',''yyyy-mm-dd hh24:mi:ss'')  and EPGGROUP\_ID is not null  group by EPGGROUP\_ID,EPGGROUP\_NAME  ) b  where total>20 and fail>5  order by b.fail\_rate desc) a  where rownum<=15';  open g\_evluation\_kpi\_dimen for v\_sql;  end if;  commit;  end;  EXCEPTION  WHEN OTHERS THEN  v\_err\_code :=SQLCODE;  v\_err\_msg :=SQLERRM;  INSERT INTO T\_ERROR VALUES(SYSDATE, v\_err\_code, 'GLOBAL\_EVALUATION\_KPI\_DIMEN: ' || v\_err\_msg);  commit;  END GLOBAL\_EVALUATION\_KPI\_DIMEN; |

## 例子4

|  |
| --- |
| create or replace PROCEDURE "P\_DIMEN\_QUERY"(  m\_query\_rate\_total OUT sys\_refcursor,  m\_query\_rate\_station OUT sys\_refcursor,  m\_query\_rate\_stb OUT sys\_refcursor,  m\_query\_rate\_pop OUT sys\_refcursor  )  AS  BEGIN  begin  declare  start\_d date;  end\_d date;  time\_gap number;  begin  select (trunc(sysdate,'dd')-to\_date('20130720 00:00:00','yyyymmdd hh24:mi:ss')) into time\_gap from dual;  select trunc(sysdate-mod(to\_char(sysdate,'mi'),5)/1440,'mi') into end\_d from dual;  select trunc(sysdate-mod(to\_char(sysdate,'mi'),5)/1440,'mi')-5/1440 into start\_d from dual;  --total  open m\_query\_rate\_total for  select  /\*+ parallel(t\_device\_status,8) \*/  sum(USER\_NUM),sum(ERROR\_NUM),  sum(AUTH\_FAILURE\_NUMBER) as authfail,  sum(AUTH\_NUMBER) as authtotal,  round(sum(AUTH\_FAILURE\_NUMBER)\*100/sum(AUTH\_NUMBER),2) as authrate,  sum(AUTH\_FAILURE\_NUM\_1+AUTH\_FAILURE\_NUM\_2) as authuser,  sum(AUTH\_FAILURE\_NUM\_2) as authfailuser,    sum(MULTICAST\_REQ\_FAIL\_NUMBER) as multireqfail,  sum(MULTICAST\_REQ\_NUMBER) as multireqtotal,  round( decode(sign(sum(MULTICAST\_REQ\_NUMBER)-0),1,sum(MULTICAST\_REQ\_FAIL\_NUMBER)/sum(MULTICAST\_REQ\_NUMBER),0)\*100,2) as multireqrate,  sum(MULTICAST\_REQ\_NUM\_1+MULTICAST\_REQ\_NUM\_2) as multirequser,  sum(MULTICAST\_REQ\_NUM\_2) as multireqfailuser,  sum(VOD\_REQ\_FAIL\_NUMBER) as vodreqfail,  sum(VOD\_REQ\_NUMBER) as vodreqtotal,  round( decode(sign(sum(VOD\_REQ\_NUMBER)-0),1,sum(VOD\_REQ\_FAIL\_NUMBER)/sum(VOD\_REQ\_NUMBER),0)\*100,2) as vodreqrate,  sum(VOD\_REQ\_NUM\_1+VOD\_REQ\_NUM\_2) as vodrequser,  sum(VOD\_REQ\_NUM\_2) as vodreqfailuser,  sum(PLAYERROR\_NUMBER) as playerrors,  sum(PLAYERROR\_NUM\_2) as playerroruser,  sum(PLAYERROR\_NUM\_1+PLAYERROR\_NUM\_2) as playuser,  sum(TOTAL\_MULTI\_MLR\_TIME) as multifail,  sum(MULTICAST\_TIME) as multitotal,  round(decode(sign(sum(MULTICAST\_TIME)-0),1,sum(TOTAL\_MULTI\_MLR\_TIME)/sum(MULTICAST\_TIME),0),2) as multimlrrate,  sum(TOTAL\_MULTI\_MLR\_NUM\_1+TOTAL\_MULTI\_MLR\_NUM\_2) as multiuser,  sum(TOTAL\_MULTI\_MLR\_NUM\_2) as multimlruser,  sum(TOTAL\_VOD\_MLR\_TIME) as vodfail,  sum(VODCAST\_TIME) as vodtotal,  round(decode(sign(sum(VODCAST\_TIME)-0),1,sum(TOTAL\_VOD\_MLR\_TIME)/sum(VODCAST\_TIME),0),2) as vodmlrrate,  sum(TOTAL\_VOD\_MLR\_NUM\_1+TOTAL\_VOD\_MLR\_NUM\_2) as voduser,  sum(TOTAL\_VOD\_MLR\_NUM\_2) vodmlruser,  sum(DELAY\_TIME\_TAG) delay\_time\_tag,  sum(ERRORCODE\_TAG) errorcode\_tag  from t\_device\_status  where STAT\_TIME>=start\_d  and STAT\_TIME<end\_d  and device\_type=1;          --station  open m\_query\_rate\_station for  select  /\*+ parallel(t\_device\_status,8) \*/  get\_stationname\_byid(b.station\_id),sum(USER\_NUM),sum(ERROR\_NUM),  sum(AUTH\_FAILURE\_NUMBER) as authfail,  sum(AUTH\_NUMBER) as authtotal,  round(sum(AUTH\_FAILURE\_NUMBER)\*100/sum(AUTH\_NUMBER),2) as authrate,  sum(AUTH\_FAILURE\_NUM\_1+AUTH\_FAILURE\_NUM\_2) as authuser,  sum(AUTH\_FAILURE\_NUM\_2) as authfailuser,    sum(MULTICAST\_REQ\_FAIL\_NUMBER) as multireqfail,  sum(MULTICAST\_REQ\_NUMBER) as multireqtotal,  round( decode(sign(sum(MULTICAST\_REQ\_NUMBER)-0),1,sum(MULTICAST\_REQ\_FAIL\_NUMBER)/sum(MULTICAST\_REQ\_NUMBER),0)\*100,2) as multireqrate,  sum(MULTICAST\_REQ\_NUM\_1+MULTICAST\_REQ\_NUM\_2) as multirequser,  sum(MULTICAST\_REQ\_NUM\_2) as multireqfailuser,  sum(VOD\_REQ\_FAIL\_NUMBER) as vodreqfail,  sum(VOD\_REQ\_NUMBER) as vodreqtotal,  round( decode(sign(sum(VOD\_REQ\_NUMBER)-0),1,sum(VOD\_REQ\_FAIL\_NUMBER)/sum(VOD\_REQ\_NUMBER),0)\*100,2) as vodreqrate,  sum(VOD\_REQ\_NUM\_1+VOD\_REQ\_NUM\_2) as vodrequser,  sum(VOD\_REQ\_NUM\_2) as vodreqfailuser,  sum(PLAYERROR\_NUMBER) as playerrors,  sum(PLAYERROR\_NUM\_2) as playerroruser,  sum(PLAYERROR\_NUM\_1+PLAYERROR\_NUM\_2) as playuser,  sum(TOTAL\_MULTI\_MLR\_TIME) as multifail,  sum(MULTICAST\_TIME) as multitotal,  round(decode(sign(sum(MULTICAST\_TIME)-0),1,sum(TOTAL\_MULTI\_MLR\_TIME)/sum(MULTICAST\_TIME),0),2) as multimlrrate,  sum(TOTAL\_MULTI\_MLR\_NUM\_1+TOTAL\_MULTI\_MLR\_NUM\_2) as multiuser,  sum(TOTAL\_MULTI\_MLR\_NUM\_2) as multimlruser,  sum(TOTAL\_VOD\_MLR\_TIME) as vodfail,  sum(VODCAST\_TIME) as vodtotal,  round(decode(sign(sum(VODCAST\_TIME)-0),1,sum(TOTAL\_VOD\_MLR\_TIME)/sum(VODCAST\_TIME),0),2) as vodmlrrate,  sum(TOTAL\_VOD\_MLR\_NUM\_1+TOTAL\_VOD\_MLR\_NUM\_2) as voduser,  sum(TOTAL\_VOD\_MLR\_NUM\_2) vodmlruser,  sum(DELAY\_TIME\_TAG) delay\_time\_tag,  sum(ERRORCODE\_TAG) errorcode\_tag  from t\_device\_status a  left join t\_device b  on a.device\_id=b.device\_id  where STAT\_TIME>=start\_d  and STAT\_TIME<end\_d  and a.device\_type=1  group by b.station\_id;      --stb enterprise  --station  open m\_query\_rate\_stb for  select  /\*+ parallel(t\_stb\_status,8) \*/  stb\_oui,sum(USER\_NUM),sum(ERROR\_NUM),  sum(AUTH\_FAILURE\_NUMBER) as authfail,  sum(AUTH\_NUMBER) as authtotal,  round( decode(sign(sum(AUTH\_NUMBER)-0),1,sum(AUTH\_FAILURE\_NUMBER)/sum(AUTH\_NUMBER),0)\*100,2) as authrate,  sum(AUTH\_FAILURE\_NUM\_1+AUTH\_FAILURE\_NUM\_2) as authuser,  sum(AUTH\_FAILURE\_NUM\_2) as authfailuser,    sum(MULTICAST\_REQ\_FAIL\_NUMBER) as multireqfail,  sum(MULTICAST\_REQ\_NUMBER) as multireqtotal,  round( decode(sign(sum(MULTICAST\_REQ\_NUMBER)-0),1,sum(MULTICAST\_REQ\_FAIL\_NUMBER)/sum(MULTICAST\_REQ\_NUMBER),0)\*100,2) as multireqrate,  sum(MULTICAST\_REQ\_NUM\_1+MULTICAST\_REQ\_NUM\_2) as multirequser,  sum(MULTICAST\_REQ\_NUM\_2) as multireqfailuser,  sum(VOD\_REQ\_FAIL\_NUMBER) as vodreqfail,  sum(VOD\_REQ\_NUMBER) as vodreqtotal,  round( decode(sign(sum(VOD\_REQ\_NUMBER)-0),1,sum(VOD\_REQ\_FAIL\_NUMBER)/sum(VOD\_REQ\_NUMBER),0)\*100,2) as vodreqrate,  sum(VOD\_REQ\_NUM\_1+VOD\_REQ\_NUM\_2) as vodrequser,  sum(VOD\_REQ\_NUM\_2) as vodreqfailuser,  sum(PLAYERROR\_NUMBER) as playerrors,  sum(PLAYERROR\_NUM\_2) as playerroruser,  sum(PLAYERROR\_NUM\_1+PLAYERROR\_NUM\_2) as playuser,  sum(TOTAL\_MULTI\_MLR\_TIME) as multifail,  sum(MULTICAST\_TIME) as multitotal,  round(decode(sign(sum(MULTICAST\_TIME)-0),1,sum(TOTAL\_MULTI\_MLR\_TIME)/sum(MULTICAST\_TIME),0),2) as multimlrrate,  sum(TOTAL\_MULTI\_MLR\_NUM\_1+TOTAL\_MULTI\_MLR\_NUM\_2) as multiuser,  sum(TOTAL\_MULTI\_MLR\_NUM\_2) as multimlruser,  sum(TOTAL\_VOD\_MLR\_TIME) as vodfail,  sum(VODCAST\_TIME) as vodtotal,  round(decode(sign(sum(VODCAST\_TIME)-0),1,sum(TOTAL\_VOD\_MLR\_TIME)/sum(VODCAST\_TIME),0),2) as vodmlrrate,  sum(TOTAL\_VOD\_MLR\_NUM\_1+TOTAL\_VOD\_MLR\_NUM\_2) as voduser,  sum(TOTAL\_VOD\_MLR\_NUM\_2) vodmlruser,  sum(DELAY\_TIME\_TAG) delay\_time\_tag,  sum(ERRORCODE\_TAG) errorcode\_tag  from t\_stb\_status a  left join t\_stb b  on a.stb\_name=b.stb\_name  where STAT\_TIME>=start\_d  and STAT\_TIME<end\_d  and a.stb\_class=3  group by b.stb\_oui;  -----2017/8/4----新增 pop维度  open m\_query\_rate\_pop for  select  /\*+ parallel(t\_stb\_status,8) \*/  a.pop\_name,sum(USER\_NUM),sum(ERROR\_NUM),  sum(AUTH\_FAILURE\_NUMBER) as authfail,  sum(AUTH\_NUMBER) as authtotal,  round( decode(sign(sum(AUTH\_NUMBER)-0),1,sum(AUTH\_FAILURE\_NUMBER)/sum(AUTH\_NUMBER),0)\*100,2) as authrate,  sum(AUTH\_FAILURE\_NUM\_1+AUTH\_FAILURE\_NUM\_2) as authuser,  sum(AUTH\_FAILURE\_NUM\_2) as authfailuser,    sum(MULTICAST\_REQ\_FAIL\_NUMBER) as multireqfail,  sum(MULTICAST\_REQ\_NUMBER) as multireqtotal,  round( decode(sign(sum(MULTICAST\_REQ\_NUMBER)-0),1,sum(MULTICAST\_REQ\_FAIL\_NUMBER)/sum(MULTICAST\_REQ\_NUMBER),0)\*100,2) as multireqrate,  sum(MULTICAST\_REQ\_NUM\_1+MULTICAST\_REQ\_NUM\_2) as multirequser,  sum(MULTICAST\_REQ\_NUM\_2) as multireqfailuser,  sum(VOD\_REQ\_FAIL\_NUMBER) as vodreqfail,  sum(VOD\_REQ\_NUMBER) as vodreqtotal,  round( decode(sign(sum(VOD\_REQ\_NUMBER)-0),1,sum(VOD\_REQ\_FAIL\_NUMBER)/sum(VOD\_REQ\_NUMBER),0)\*100,2) as vodreqrate,  sum(VOD\_REQ\_NUM\_1+VOD\_REQ\_NUM\_2) as vodrequser,  sum(VOD\_REQ\_NUM\_2) as vodreqfailuser,  sum(PLAYERROR\_NUMBER) as playerrors,  sum(PLAYERROR\_NUM\_2) as playerroruser,  sum(PLAYERROR\_NUM\_1+PLAYERROR\_NUM\_2) as playuser,  sum(TOTAL\_MULTI\_MLR\_TIME) as multifail,  sum(MULTICAST\_TIME) as multitotal,  round(decode(sign(sum(MULTICAST\_TIME)-0),1,sum(TOTAL\_MULTI\_MLR\_TIME)/sum(MULTICAST\_TIME),0),2) as multimlrrate,  sum(TOTAL\_MULTI\_MLR\_NUM\_1+TOTAL\_MULTI\_MLR\_NUM\_2) as multiuser,  sum(TOTAL\_MULTI\_MLR\_NUM\_2) as multimlruser,  sum(TOTAL\_VOD\_MLR\_TIME) as vodfail,  sum(VODCAST\_TIME) as vodtotal,  round(decode(sign(sum(VODCAST\_TIME)-0),1,sum(TOTAL\_VOD\_MLR\_TIME)/sum(VODCAST\_TIME),0),2) as vodmlrrate,  sum(TOTAL\_VOD\_MLR\_NUM\_1+TOTAL\_VOD\_MLR\_NUM\_2) as voduser,  sum(TOTAL\_VOD\_MLR\_NUM\_2) vodmlruser,  sum(DELAY\_TIME\_TAG) delay\_time\_tag,  sum(ERRORCODE\_TAG) errorcode\_tag  from T\_POP\_STATUS a  left join T\_POP\_INFO b  on a.pop\_id=b.pop\_id  where STAT\_TIME>=start\_d  and STAT\_TIME<end\_d  and a.pop\_type=1  group by a.pop\_name;  end;  commit;  end;  END P\_DIMEN\_QUERY; |

# 程序包

## 例子1

|  |
| --- |
| create or replace PACKAGE "DB\_MAITAIN" AS  -- TABLE\_SPACE  FUNCTION checksize ( tb\_size number,unit\_size IN VARCHAR2 ) RETURN number;--检查需要多少个数据文件  FUNCTION re\_size ( tb\_size number,unit\_size IN VARCHAR2 ) RETURN number;--转换为M  procedure P\_CREATE\_TABLESPACE(TBS\_NAME in varchar2, TBS\_SIZE in number, unit\_size in varchar2,PATH in varchar2,ret out number);  --创建表空间、如果大于32G,分成多个数据文件  procedure P\_CREATE\_TABLESPACE(DF\_NAME in varchar2,TBS\_NAME in varchar2, TBS\_SIZE in number, unit\_size in varchar2,PATH in varchar2,ret out number);  --创建表空间、如果大于32G,分成多个数据文件，存储于指定目录  procedure P\_RESIZE\_TABLESPACE(TBS\_NAME in varchar2, TBS\_SIZE in number, unit\_size in varchar2,ret out number);  --创建表空间、如果大于32G,分成多个数据文件,默认压缩属性  procedure P\_CREATE\_COMPROESS\_TABLESPACE(TBS\_NAME in varchar2,  TBS\_SIZE in number,  unit\_size in varchar2,  PATH in varchar2,  ret out number);  --暂不提供  procedure P\_DROP\_TABLESPACE(TBS\_NAME varchar2 ,ret out number);  --删除表空间  procedure P\_OFFLINE\_TABLESPACE(TBS\_NAME varchar2, ret out number);  --设置表空间离线  procedure P\_CHECK\_TABLESPACE(TBS\_NAME varchar2, ret out number);  --DATAFILE  procedure P\_ADD\_DATAFILE(DF\_NAME in varchar2,TBS\_NAME in varchar2, DF\_SIZE in number, unit\_size in varchar2,PATH in varchar2,ret out number);  --在TBS中增加数据文件,文件大小不能大于32000G  procedure P\_RESIZE\_DATAFILE(DF\_NAME in varchar2, DF\_SIZE in number, unit\_size in varchar2,ret out number);  --重新修改数据文件大小  procedure P\_DROP\_DATAFILE (DF\_NAME in varchar2, TBS\_NAME in varchar2, ret out number);  --删除数据文件  procedure P\_OFFLINE\_DATAFILE (DF\_NAME in varchar2, TBS\_NAME in varchar2, ret out number);  --设置数据文件离线  procedure P\_CHECK\_DATAFILE (DF\_NAME in varchar2, TBS\_NAME in varchar2, ret out number);  --检查是否有同名数据文件  --TABLE PARTITION  procedure P\_CREATE\_PARTITION(TABLE\_NAME in varchar2,PT\_NAME in varchar2,TBS\_NAME in varchar2,daterange in date, INIT\_SIZE in number,ret out number);  --在tbs表空间下为table增加新分区，初始大小为INIT\_SIZE,单位M,pt\_name不能超过30字节  procedure P\_DROP\_PARTITION (TABLE\_NAME in varchar2,PT\_NAME in varchar2, ret out number);  --删除分区  procedure P\_CHECK\_PARTITION (TABLE\_NAME in varchar2,PT\_NAME in varchar2, ret out number);  --检查是否有同名分区  --NORMAL TABLE  procedure P\_TABLE\_REMOVEABLE (TABLE\_NAME in varchar2,ret out number);  --设置表行可迁移  procedure P\_RELEASE\_TABLE\_SIZE (TABLE\_NAME in varchar2,ret out number);  --收缩表  --移动分区表的表空间  procedure P\_MOVE\_PARATIONED\_INDEX(TABLE\_NAME in varchar2,INDEX\_NAME IN VARCHAR2 ,ret out number);  END DB\_MAITAIN; |

## 例子2

|  |
| --- |
| create or replace PACKAGE BODY DB\_MAITAIN AS  FUNCTION checksize(tb\_size number, unit\_size IN VARCHAR2) RETURN number as  v\_check number;  begin  case upper(unit\_size)  when 'K' then  select trunc(tb\_size / 1000 / 1000 / 32) + 1  into v\_check  from dual;  return v\_check;  when 'M' then  select trunc(tb\_size / 1000 / 32) + 1 into v\_check from dual;  return v\_check;  when 'G' then  select trunc(tb\_size / 32) + 1 into v\_check from dual;  return v\_check;  else  return 0;  end case;  end;  FUNCTION re\_size(tb\_size number, unit\_size IN VARCHAR2) RETURN number as  v\_check number;  begin  case upper(unit\_size)  when 'K' then  select trunc(tb\_size / 1000) into v\_check from dual;  return v\_check;  when 'M' then  select trunc(tb\_size) into v\_check from dual;  return v\_check;  when 'G' then  select trunc(tb\_size \* 1000) into v\_check from dual;  return v\_check;  else  return 0;  end case;  end;  procedure P\_CREATE\_TABLESPACE(TBS\_NAME in varchar2,  TBS\_SIZE in number,  unit\_size in varchar2,  PATH in varchar2,  ret out number) AS  v\_sql varchar2(4000);  v\_msg varchar2(2000);  v\_code number;  tb\_check number;  df\_num number;  tb\_size number;  df\_check number;  BEGIN  /\* TODO 需要实施 \*/  p\_check\_tablespace(tbs\_name, tb\_check);    if tb\_check = 0 then  select checksize(TBS\_SIZE, unit\_size) into df\_num from dual;  select re\_size(TBS\_SIZE, unit\_size) into tb\_size from dual;  for i in 1 .. (df\_num - 1) loop  p\_check\_tablespace(tbs\_name, tb\_check);  if tb\_check = 0 then  P\_CHECK\_DATAFILE(tbs\_name || '\_' || i, TBS\_NAME, df\_check);  if df\_check = 0 then  v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || i ||  '.dbf'' SIZE 32000M  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO ';  else  v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || i ||  '.dbf'' SIZE 32000M reuse  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO ';  end if;  else  P\_CHECK\_DATAFILE(tbs\_name || '\_' || i, TBS\_NAME, df\_check);  if df\_check = 0 then  v\_sql := 'ALTER TABLESPACE ' || TBS\_NAME ||  ' ADD  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || i || '.dbf'' SIZE 32000M  ';  else  v\_sql := 'ALTER TABLESPACE ' || TBS\_NAME ||  ' ADD  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || i || '.dbf'' SIZE 32000M reuse  ';  end if;    end if;  execute immediate v\_sql;  end loop;  /\* v\_sql:='CREATE TABLESPACE '||TBS\_NAME||'  DATAFILE ''+DATA\_DM01/'||TBS\_NAME||'\_'||df\_num||'.dbf'' SIZE '||(tb\_size-32000\*(df\_num-1)+1)||'M  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO ';\*/  p\_check\_tablespace(tbs\_name, tb\_check);  if tb\_check = 0 then    P\_CHECK\_DATAFILE(tbs\_name || '\_' || df\_num, TBS\_NAME, df\_check);  if df\_check = 0 then  v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || df\_num ||  '.dbf'' SIZE ' || (tb\_size - 32000 \* (df\_num - 1) + 1) || 'M  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO ';  else  v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || df\_num ||  '.dbf'' SIZE ' || (tb\_size - 32000 \* (df\_num - 1) + 1) ||  'M reuse  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO ';  end if;  else    P\_CHECK\_DATAFILE(tbs\_name || '\_' || df\_num, TBS\_NAME, df\_check);  if df\_check = 0 then  v\_sql := 'ALTER TABLESPACE ' || TBS\_NAME ||  ' ADD  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || df\_num ||  '.dbf'' SIZE ' || (tb\_size - 32000 \* (df\_num - 1) + 1) || 'M  ';  else  v\_sql := 'ALTER TABLESPACE ' || TBS\_NAME ||  ' ADD  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || df\_num ||  '.dbf'' SIZE ' || (tb\_size - 32000 \* (df\_num - 1) + 1) ||  'M reuse  ';  end if;  end if;    execute immediate v\_sql;  ret := 0;  else  ret := 1;  end if;  EXCEPTION  WHEN OTHERS THEN  v\_msg := sqlerrm;  v\_code := SQLCODE;  rollback;  INSERT INTO t\_error  VALUES  (sysdate,  v\_code,  'P\_CREATE\_TABLESPACE, TBS\_NAME:' || TBS\_NAME || ',PATH ' || PATH || ',' ||  v\_msg);  commit;  ret := -1;  END P\_CREATE\_TABLESPACE;  procedure P\_CREATE\_TABLESPACE(DF\_NAME in varchar2,  TBS\_NAME in varchar2,  TBS\_SIZE in number,  unit\_size in varchar2,  PATH in varchar2,  ret out number) AS  v\_sql varchar2(4000);  v\_msg varchar2(2000);  v\_code number;  tb\_check number;  df\_num number;  tb\_size number;  BEGIN  /\* TODO 需要实施 \*/  p\_check\_tablespace(tbs\_name, tb\_check);    if tb\_check = 0 then  select checksize(TBS\_SIZE, unit\_size) into df\_num from dual;  select re\_size(TBS\_SIZE, unit\_size) into tb\_size from dual;  for i in 1 .. (df\_num - 1) loop    v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''' || PATH || '/' || TBS\_NAME || '\_' || i ||  '.dbf'' SIZE 32000M  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO ';  execute immediate v\_sql;  end loop;  v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''' || PATH || '/' || TBS\_NAME || '\_' || df\_num ||  '.dbf'' SIZE ' || (tb\_size - 32000 + (df\_num - 1) + 1) || 'M  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO ';  execute immediate v\_sql;  ret := 0;  else  ret := 1;  end if;  EXCEPTION  WHEN OTHERS THEN  v\_msg := sqlerrm;  v\_code := SQLCODE;  rollback;  INSERT INTO t\_error  VALUES  (sysdate,  v\_code,  'P\_CREATE\_TABLESPACE, TBS\_NAME:' || TBS\_NAME || ',PATH ' || PATH || ',' ||  v\_msg);  commit;  ret := -1;  END P\_CREATE\_TABLESPACE;      procedure P\_CREATE\_COMPROESS\_TABLESPACE(TBS\_NAME in varchar2,  TBS\_SIZE in number,  unit\_size in varchar2,  PATH in varchar2,  ret out number) AS  v\_sql varchar2(4000);  v\_msg varchar2(2000);  v\_code number;  tb\_check number;  df\_num number;  tb\_size number;  df\_check number;  BEGIN  /\* TODO 需要实施 \*/  p\_check\_tablespace(tbs\_name, tb\_check);    if tb\_check = 0 then  select checksize(TBS\_SIZE, unit\_size) into df\_num from dual;  select re\_size(TBS\_SIZE, unit\_size) into tb\_size from dual;  for i in 1 .. (df\_num - 1) loop  p\_check\_tablespace(tbs\_name, tb\_check);  if tb\_check = 0 then  P\_CHECK\_DATAFILE(tbs\_name || '\_' || i, TBS\_NAME, df\_check);  if df\_check = 0 then  v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || i ||  '.dbf'' SIZE 32000M  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO DEFAULT COMPRESS FOR QUERY';  else  v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || i ||  '.dbf'' SIZE 32000M reuse  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO DEFAULT COMPRESS FOR QUERY';  end if;  else  P\_CHECK\_DATAFILE(tbs\_name || '\_' || i, TBS\_NAME, df\_check);  if df\_check = 0 then  v\_sql := 'ALTER TABLESPACE ' || TBS\_NAME ||  ' ADD  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || i || '.dbf'' SIZE 32000M  ';  else  v\_sql := 'ALTER TABLESPACE ' || TBS\_NAME ||  ' ADD  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || i || '.dbf'' SIZE 32000M reuse  ';  end if;    end if;  execute immediate v\_sql;  end loop;  /\* v\_sql:='CREATE TABLESPACE '||TBS\_NAME||'  DATAFILE ''+DATA\_DM01/'||TBS\_NAME||'\_'||df\_num||'.dbf'' SIZE '||(tb\_size-32000\*(df\_num-1)+1)||'M  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO ';\*/  p\_check\_tablespace(tbs\_name, tb\_check);  if tb\_check = 0 then    P\_CHECK\_DATAFILE(tbs\_name || '\_' || df\_num, TBS\_NAME, df\_check);  if df\_check = 0 then  v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || df\_num ||  '.dbf'' SIZE ' || (tb\_size - 32000 \* (df\_num - 1) + 1) || 'M  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO DEFAULT COMPRESS FOR QUERY';  else  v\_sql := 'CREATE TABLESPACE ' || TBS\_NAME || '  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || df\_num ||  '.dbf'' SIZE ' || (tb\_size - 32000 \* (df\_num - 1) + 1) ||  'M reuse  EXTENT MANAGEMENT LOCAL  SEGMENT SPACE MANAGEMENT AUTO DEFAULT COMPRESS FOR QUERY';  end if;  else    P\_CHECK\_DATAFILE(tbs\_name || '\_' || df\_num, TBS\_NAME, df\_check);  if df\_check = 0 then  v\_sql := 'ALTER TABLESPACE ' || TBS\_NAME ||  ' ADD  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || df\_num ||  '.dbf'' SIZE ' || (tb\_size - 32000 \* (df\_num - 1) + 1) || 'M  ';  else  v\_sql := 'ALTER TABLESPACE ' || TBS\_NAME ||  ' ADD  DATAFILE ''+DATA\_DM01/' || TBS\_NAME || '\_' || df\_num ||  '.dbf'' SIZE ' || (tb\_size - 32000 \* (df\_num - 1) + 1) ||  'M reuse  ';  end if;  end if;    execute immediate v\_sql;  ret := 0;  else  ret := 1;  end if;  EXCEPTION  WHEN OTHERS THEN  v\_msg := sqlerrm;  v\_code := SQLCODE;  rollback;  INSERT INTO t\_error  VALUES  (sysdate,  v\_code,  'P\_CREATE\_COMPROESS\_TABLESPACE, TBS\_NAME:' || TBS\_NAME || ',PATH ' || PATH || ',' ||  v\_msg);    ret := -1;  commit;  END P\_CREATE\_COMPROESS\_TABLESPACE;  procedure P\_RESIZE\_TABLESPACE(TBS\_NAME in varchar2,  TBS\_SIZE in number,  unit\_size in varchar2,  ret out number) AS  BEGIN  /\* TODO 需要实施 \*/  NULL;  END P\_RESIZE\_TABLESPACE;  procedure P\_DROP\_TABLESPACE(TBS\_NAME varchar2, ret out number) AS  v\_sql varchar2(4000);  v\_msg varchar2(2000);  v\_code number;  tb\_check number;  BEGIN  /\* TODO 需要实施 \*/  p\_check\_tablespace(tbs\_name, tb\_check);  if tb\_check != 0 then    v\_sql := 'DROP TABLESPACE ' || TBS\_NAME || '  INCLUDING CONTENTS and datafiles cascade constraints';  dbms\_output.put\_line(v\_sql);  execute immediate v\_sql;  ret := 0;  else  ret := 1;  end if;    EXCEPTION  WHEN OTHERS THEN  v\_msg := sqlerrm;  v\_code := SQLCODE;  rollback;  INSERT INTO t\_error  VALUES  (sysdate,  v\_code,  'P\_DROP\_TABLESPACE, TBS\_NAME:' || TBS\_NAME || ',' || v\_msg);  ret := -1;  commit;  END P\_DROP\_TABLESPACE;  procedure P\_OFFLINE\_TABLESPACE(TBS\_NAME varchar2, ret out number) AS  v\_sql varchar2(4000);  BEGIN  /\* TODO 需要实施 \*/  v\_sql := ' alter tablespace ' || TBS\_NAME || ' offline normal ';  execute immediate v\_sql;  ret := 0;    END P\_OFFLINE\_TABLESPACE;  procedure P\_CHECK\_TABLESPACE(TBS\_NAME varchar2, ret out number) AS  BEGIN  /\* TODO 需要实施 \*/  select count(\*)  into ret  from dba\_tablespaces  where upper(tablespace\_name) = upper(tbs\_name);  NULL;  END P\_CHECK\_TABLESPACE;  --DATAFILE  procedure P\_ADD\_DATAFILE(DF\_NAME in varchar2,  TBS\_NAME in varchar2,  DF\_SIZE in number,  unit\_size in varchar2,  PATH in varchar2,  ret out number) As  v\_sql varchar2(4000);  v\_msg varchar2(2000);  v\_code number;  df\_check number;  BEGIN  /\* TODO 需要实施 \*/  P\_check\_datafile(PATH || '/' || DF\_NAME, TBS\_NAME, df\_check);  if df\_check = 0 then    v\_sql := ' alter tablespace ' || TBS\_NAME || ' add datafile ''' || PATH || '/' ||  DF\_NAME || ' size ' || DF\_SIZE || unit\_size;  execute immediate v\_sql;  ret := 0;  else  ret := 1;  end if;  EXCEPTION  WHEN OTHERS THEN  v\_msg := sqlerrm;  v\_code := SQLCODE;  rollback;  INSERT INTO t\_error  VALUES  (sysdate,  v\_code,  'P\_ADD\_DATAFILE, TBS\_NAME:' || TBS\_NAME || ',DF\_NAME:' || PATH || '/' ||  DF\_NAME || ',' || v\_msg);  ret := -1;  commit;  END;  --在TBS中增加数据文件  procedure P\_RESIZE\_DATAFILE(DF\_NAME in varchar2,  DF\_SIZE in number,  unit\_size in varchar2,  ret out number) AS  v\_sql varchar2(4000);  BEGIN  /\* TODO 需要实施 \*/  v\_sql := 'ALTER DATABASE  DATAFILE ''' || DF\_NAME || '  RESIZE ' || DF\_SIZE || unit\_size;  execute immediate v\_sql;  ret := 0;    END;  --重新修改数据文件大小  procedure P\_DROP\_DATAFILE(DF\_NAME in varchar2,  TBS\_NAME in varchar2,  ret out number) AS  BEGIN  /\* TODO 需要实施 \*/  NULL;  END;  --删除数据文件  procedure P\_OFFLINE\_DATAFILE(DF\_NAME in varchar2,  TBS\_NAME in varchar2,  ret out number) AS  BEGIN  /\* TODO 需要实施 \*/  NULL;  END;  --设置数据文件离线  procedure P\_CHECK\_DATAFILE(DF\_NAME in varchar2,  TBS\_NAME in varchar2,  ret out number) AS  BEGIN  /\* TODO 需要实施 \*/  select count(\*) into ret from v$datafile where upper(name) = upper(df\_name);    END;  --检查是否有同名数据文件  --TABLE PARTITION  procedure P\_CREATE\_PARTITION(TABLE\_NAME in varchar2,  PT\_NAME in varchar2,  TBS\_NAME in varchar2,  daterange in date,  INIT\_SIZE in number,  ret out number) AS  v\_sql varchar2(4000);  v\_msg varchar2(2000);  v\_code number;  pt\_check number;  BEGIN  /\* TODO 需要实施 \*/  P\_CHECK\_PARTITION(TABLE\_NAME, pt\_name, pt\_check);  if pt\_check = 0 then  v\_sql := 'ALTER TABLE ' || TABLE\_NAME || ' ADD PARTITION ' || pt\_name || '  VALUES LESS THAN (to\_date(' ||  to\_char(daterange, 'YYYYMMDDHH24MI') ||  ',''YYYYMMDDHH24MI'')) TABLESPACE ' || TBS\_NAME || '  STORAGE ( INITIAL 4M NEXT 16M MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0)';  execute immediate v\_sql;  ret := 0;  else  ret := 1;  end if;  EXCEPTION  WHEN OTHERS THEN  v\_msg := sqlerrm;  v\_code := SQLCODE;  rollback;  INSERT INTO t\_error  VALUES  (sysdate,  v\_code,  'P\_CREATE\_PARTITION, TBS\_NAME:' || TBS\_NAME || ',TABLE\_NAME:' ||  TABLE\_NAME || ',UPRANGE:' || to\_char(daterange, 'YYYYMMDDHH24MI') || ',' ||  v\_msg);  ret := -1;  commit;  END;  --在tbs表空间下为table增加新分区，初始大小为INIT\_SIZE,单位M  procedure P\_DROP\_PARTITION(TABLE\_NAME in varchar2,  PT\_NAME in varchar2,  ret out number) AS  v\_sql varchar2(4000);  v\_msg varchar2(2000);  v\_code number;  pt\_check number;  BEGIN  /\* TODO 需要实施 \*/  P\_CHECK\_PARTITION(TABLE\_NAME, pt\_name, pt\_check);  if pt\_check != 0 then  v\_sql := 'ALTER TABLE ' || TABLE\_NAME || ' drop PARTITION ' ||  pt\_name;  execute immediate v\_sql;  ret := 0;  else  ret := 1;  end if;  EXCEPTION  WHEN OTHERS THEN  v\_msg := sqlerrm;  v\_code := SQLCODE;  rollback;  INSERT INTO t\_error  VALUES  (sysdate,  v\_code,  'P\_DROP\_PARTITION, PT\_NAME:' || PT\_NAME || ',TABLE\_NAME:' ||  TABLE\_NAME || ',' || v\_msg);  ret := -1;  commit;  END;  --删除分区  procedure P\_CHECK\_PARTITION(TABLE\_NAME in varchar2,  PT\_NAME in varchar2,  ret out number) AS  v\_table varchar2(50);  BEGIN  /\* TODO 需要实施 \*/  v\_table := TABLE\_NAME;  select count(\*)  into ret  from user\_TAB\_PARTITIONS  where table\_name = v\_table  and partition\_name = pt\_name;  NULL;  END;  --检查是否有同名分区  --NORMAL TABLE  procedure P\_TABLE\_REMOVEABLE(TABLE\_NAME in varchar2, ret out number) AS  v\_sql varchar2(4000);  BEGIN  v\_sql := 'alter table ' || TABLE\_NAME || ' row movement enable ';  execute immediate v\_sql;  ret := 0;  NULL;  END;  procedure P\_RELEASE\_TABLE\_SIZE(TABLE\_NAME in varchar2, ret out number) AS  v\_sql varchar2(4000);  BEGIN  /\* TODO 需要实施 \*/  v\_sql := ' alter table ' || TABLE\_NAME || ' shrink space cascade';  execute immediate v\_sql;  ret := 0;  END;    procedure P\_MOVE\_PARATIONED\_INDEX(TABLE\_NAME in varchar2,INDEX\_NAME IN VARCHAR2 ,ret out number) AS  CURSOR DATA\_CALCULATED  IS  select a.\*,b.tablespace\_name from (select  t.subpartition\_name,t.partition\_name FROM USER\_IND\_SUBPARTITIONS T  WHERE T.INDEX\_NAME=INDEX\_NAME) a  , user\_tab\_partitions b where a.partition\_name=b.partition\_name;  v\_sql varchar2(4000);  BEGIN  FOR IDX IN DATA\_CALCULATED  LOOP  v\_sql:='ALTER INDEX '||INDEX\_NAME||' REBUILD SUBPARTITION '||IDX.SUBPARTITION\_NAME||' TABLESPACE '||IDX.TABLESPACE\_NAME;    execute immediate v\_sql;  END LOOP;  END;  END DB\_MAITAIN; |

# 触发器

|  |
| --- |
| create or replace trigger "TRIG\_IPTVQOSADV\_TO\_IPTVPRO"  after insert or update or delete  on t\_current\_alarm FOR EACH ROW  declare  err\_num number;  err\_msg varchar2(200);  lasttime date;  begin  if inserting then  --all new alarm should be send  insert into t\_alarm\_to\_iptvpro values (SEQ\_ALARM\_TOIPTVPRO.nextval,:NEW.DIMEN\_TYPE,:NEW.DIMEN\_ID,:NEW.DIMEN\_NAME,  :NEW.ABSTRACT,:NEW.ALARM\_TYPE,:NEW.FIRSTOCCURRENCE,:NEW.LASTOCCURRENCE,:NEW.TALLY,:NEW.ACKNOWLEDGED,:NEW.CONFIRMTIME,  :NEW.CONFIRMUSER,:NEW.DELETETIME,:NEW.DELETEUSER,0,0);  elsif updating then  insert into t\_alarm\_to\_iptvpro values (SEQ\_ALARM\_TOIPTVPRO.nextval,:NEW.DIMEN\_TYPE,:NEW.DIMEN\_ID,:NEW.DIMEN\_NAME,  :NEW.ABSTRACT,:NEW.ALARM\_TYPE,:NEW.FIRSTOCCURRENCE,:NEW.LASTOCCURRENCE,:NEW.TALLY,:NEW.ACKNOWLEDGED,:NEW.CONFIRMTIME,  :NEW.CONFIRMUSER,:NEW.DELETETIME,:NEW.DELETEUSER,0,1);  elsif deleting then  --all delete alarm should be send  insert into t\_alarm\_to\_iptvpro values (SEQ\_ALARM\_TOIPTVPRO.nextval,:OLD.DIMEN\_TYPE,:OLD.DIMEN\_ID,:OLD.DIMEN\_NAME,  :OLD.ABSTRACT,:OLD.ALARM\_TYPE,:OLD.FIRSTOCCURRENCE,:OLD.LASTOCCURRENCE,:OLD.TALLY,:OLD.ACKNOWLEDGED,:OLD.CONFIRMTIME,  :OLD.CONFIRMUSER,:OLD.DELETETIME,:OLD.DELETEUSER,0,1);    end if;  Exception  when no\_data\_found then  err\_msg:=substr(sqlerrm,1,900);  insert into t\_errors values(sysdate,0,'sysn alarm to IPTVPRO insert new alarm '||err\_msg);  when others then  err\_num:=sqlcode;  err\_msg:=substr(sqlerrm,1,900);  insert into t\_errors values(sysdate,err\_num,'sysn alarm to IPTVPRO insert new alarm '||err\_msg);  end; |

# 类型

|  |
| --- |
| create or replace TYPE "AD\_QUERYDEVICE\_RESULT" as object(  s\_time varchar2(50),  e\_time varchar2(50),  v\_url varchar2(4000),  host varchar2(50)  )  create or replace TYPE "AD\_QUERY\_RESULT\_ARRAY" as table of ad\_querydevice\_result |