## Oracle创建每天生成AWR报告

**1.创建生成AWR报告的目录 及 赋予用户的权限(使用sys或system账户)**

create or replace directory DIR\_AWRREPORT as 'E:\DataBase\Oracle\admin\orcl\awrreport';

grant read,write on directory DIR\_AWRREPORT to BDCXXK;

grant select on v\_$database to BDCXXK;

grant select on dba\_hist\_snapshot to BDCXXK;

grant execute on sys.dbms\_workload\_repository to BDCXXK;

**2.创建自动生成AWR报告的存储过程(使用监管账户执行)**

CREATE OR REPLACE PROCEDURE auto\_awrreport

AS

*/\**

*用途 ：自动生成AWR报告*

*创建人 ：gegeman*

*创建日期：2017-12-08*

*\*/*

start\_snap number;

end\_snap number;

rpt\_interval number := 2; *--报告间隔(小时)*

start\_time varchar2(14);

end\_time varchar2(14);

awr\_file utl\_file.file\_type;

v\_dbid number;

v\_dbname varchar2(20);

begin

*--确定数据库名称与id*

select dbid,name into v\_dbid,v\_dbname from v$database;

*---查询起始的快照id*

select max(snap\_id),min(snap\_id) into end\_snap,start\_snap from dba\_hist\_snapshot;

if((end\_snap - rpt\_interval) > start\_snap) then

start\_snap := end\_snap - rpt\_interval;

end if;

*---格式化快照时间*

select to\_char(end\_interval\_time-rpt\_interval/24, 'yyyymmddhh24'),to\_char(end\_interval\_time, 'yyyymmddhh24')

into start\_time,end\_time

from dba\_hist\_snapshot

where snap\_id = end\_snap and instance\_number = 1;

*--实例1报告*

awr\_file := utl\_file.fopen('DIR\_AWRREPORT',lower(v\_dbname) || '\_1\_' || start\_time ||'\_'|| end\_time ||'.html', 'a',32767 );

for awr\_info in ( select output from table( dbms\_workload\_repository.awr\_report\_html(v\_dbid,1,start\_snap,end\_snap,0) ) ) loop

UTL\_FILE.put\_line(awr\_file,awr\_info.output);

end loop;

utl\_file.fclose(awr\_file);

end auto\_awrreport;

**3. 授予监控用户执行procedure权限（使用sys或system账户）**

grant execute on BDCXXK.auto\_awrreport to BDCXXK;

**4.创建job，每天间隔两个小时执行（使用监控账户执行）**

declare job1 number;

begin

sys.dbms\_job.submit(job => job1,

what => 'auto\_awrreport;', *--要执行的存储过程名称*

next\_date => sysdate, *--立即执行*

interval => 'sysdate +2/24' *--每次间隔两个时间*

);

end;

**5.job的一些知识**

 删除job:dbms\_job.remove(jobno);

     修改要执行的操作:job:dbms\_job.what(jobno,what);

     修改下次执行时间：dbms\_job.next\_date(job,next\_date);

     修改间隔时间：dbms\_job.interval(job,interval);

     停止job:dbms.broken(job,broken,nextdate);

     启动job:dbms\_job.run(jobno);

 job：输出变量，是此任务在任务队列中的编号；

●    what：执行的任务的名称及其输入参数；

●    next\_date：任务执行的时间, 如果要立即执行则设为SYSDATE

●    interval：任务执行的时间间隔。

每天：trunc(sysdate+1)

每周：trunc(sysdate+7)

每月：trunc(sysdate+30)

每个星期日：next\_day(trunc(sysdate),'星期日')

每天6点：trunc(sysdate+1)+6/24

半个小时：sysdate+30/(24\*60)