oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（16）awrinput.sql

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（16）awrinput.sql

```
#A chunk of common code used for SWRF reports and ADDM.
Rem      This script gets the dbid,eid,filename,etc from the user
Rem      for both components to use.
#

Rem
Rem $Header: awrinput.sql 22-may-2005.14:30:58 mlfeng Exp $
Rem
Rem awrinput.sql
Rem
Rem Copyright (c) 2003, 2005, Oracle. All rights reserved.
Rem
Rem    NAME
Rem      awrinput.sql - <one-line expansion of the name>
Rem
Rem    DESCRIPTION
Rem      A chunk of common code used for SWRF reports and ADDM.
Rem      This script gets the dbid,eid,filename,etc from the user
Rem      for both components to use.
Rem
Rem    NOTES
Rem      This script could leave a few other SQL*Plus substitution and/or
Rem      bind variables defined at the end.
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mlfeng      05/22/05 - remove leading blank from date conversion
Rem    adagarwa    11/22/04 - Move code for obtaining report name to
Rem                           awrinpnm.sql
Rem    pbelknap    10/15/03 - swrf reporting to html in pl/sql module
Rem    veeve       10/02/03 -
Rem    pbelknap    10/02/03 - fixing for text reports
Rem    veeve       10/01/03 - show current instance and
Rem                           give default values for dbid and inst_num
Rem    pbelknap    10/01/03 - Created
Rem


-- The following list of SQL*Plus bind variables will be defined and assigned a value
-- by this SQL*Plus script:
--    variable dbid      number     - Database id
--    variable inst_num  number     - Instance number
--    variable bid       number     - Begin snapshot id
--    variable eid       number     - End snapshot id


clear break compute;
repfooter off;
ttitle off;
btitle off;

set heading on;
set timing off veri off space 1 flush on pause off termout on numwidth 10;
set echo off feedback off pagesize 60 linesize 80 newpage 1 recsep off;
```

```
set trimspool on trimout on define "&" concat "." serveroutput on;
set underline on;

--
-- Request the DB Id and Instance Number, if they are not specified

column instt_num  heading "Inst Num"  format 99999;
column instt_name heading "Instance"  format a12;
column dbb_name   heading "DB Name"   format a12;
column dbbid      heading "DB Id"     format a12 just c;
column host       heading "Host"      format a12;

prompt
prompt
prompt Instances in this Workload Repository schema
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
select distinct
       (case when cd.dbid = wr.dbid and
                  cd.name = wr.db_name and
                  ci.instance_number = wr.instance_number and
                  ci.instance_name   = wr.instance_name
             then '* '
             else '  '
        end) || wr.dbid   dbbid
     , wr.instance_number instt_num
     , wr.db_name         dbb_name
     , wr.instance_name   instt_name
     , wr.host_name       host
  from dba_hist_database_instance wr, v$database cd, v$instance ci;

prompt
prompt Using &&dbid for database Id
prompt Using &&inst_num for instance number


--
--  Set up the binds for dbid and instance_number

variable dbid       number;
variable inst_num   number;
begin
  :dbid      :=  &dbid;
  :inst_num  :=  &inst_num;
end;
/

--
--  Error reporting

whenever sqlerror exit;
variable max_snap_time char(10);
declare

  cursor cidnum is
     select 'X'
       from dba_hist_database_instance
      where instance_number = :inst_num
        and dbid            = :dbid;
```

```
  cursor csnapid is
     select to_char(max(end_interval_time),'dd/mm/yyyy')
       from dba_hist_snapshot
      where instance_number = :inst_num
        and dbid            = :dbid;

  vx     char(1);

begin

  -- Check Database Id/Instance Number is a valid pair
  open cidnum;
  fetch cidnum into vx;
  if cidnum%notfound then
    raise_application_error(-20200,
      'Database/Instance ' || :dbid || '/' || :inst_num ||
      ' does not exist in DBA_HIST_DATABASE_INSTANCE');
  end if;
  close cidnum;

  -- Check Snapshots exist for Database Id/Instance Number
  open csnapid;
  fetch csnapid into :max_snap_time;
  if csnapid%notfound then
    raise_application_error(-20200,
      'No snapshots exist for Database/Instance '||:dbid||'/'||:inst_num);
  end if;
  close csnapid;

end;
/
whenever sqlerror continue;


--
--  Ask how many days of snapshots to display

set termout on;
column instart_fmt noprint;
column inst_name   format a12  heading 'Instance';
column db_name     format a12  heading 'DB Name';
column snap_id     format 99999990 heading 'Snap Id';
column snapdat     format a18  heading 'Snap Started' just c;
column lvl         format 99   heading 'Snap|Level';

prompt
prompt
prompt Specify the number of days of snapshots to choose from
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt Entering the number of days (n) will result in the most recent
prompt (n) days of snapshots being listed.  Pressing <return> without
prompt specifying a number lists all completed snapshots.
prompt
prompt

set heading off;
column num_days new_value num_days noprint;
```

```
select    'Listing '
       || decode( nvl('&&num_days', 3.14)
                , 0    , 'no snapshots'
                , 3.14 , 'all Completed Snapshots'
                , 1    , 'the last day''s Completed Snapshots'
                , 'the last &num_days days of Completed Snapshots')
     , nvl('&&num_days', 3.14)  num_days
  from sys.dual;
set heading on;


--
-- List available snapshots

break on inst_name on db_name on host on instart_fmt skip 1;

ttitle off;

select to_char(s.startup_time,'dd Mon "at" HH24:mi:ss')  instart_fmt
     , di.instance_name                                  inst_name
     , di.db_name                                        db_name
     , s.snap_id                                         snap_id
     , to_char(s.end_interval_time,'dd Mon YYYY HH24:mi') snapdat
     , s.snap_level                                      lvl
  from dba_hist_snapshot s
     , dba_hist_database_instance di
 where s.dbid              = :dbid
   and di.dbid             = :dbid
   and s.instance_number   = :inst_num
   and di.instance_number  = :inst_num
   and di.dbid             = s.dbid
   and di.instance_number  = s.instance_number
   and di.startup_time     = s.startup_time
   and s.end_interval_time >= decode( &num_days
                                , 0   , to_date('31-JAN-9999','DD-MON-YYYY')
                                , 3.14, s.end_interval_time
                                , to_date(:max_snap_time,'dd/mm/yyyy') - (&num_days-1))
 order by db_name, instance_name, snap_id;

clear break;
ttitle off;


--
--  Ask for the snapshots Id's which are to be compared

prompt
prompt
prompt Specify the Begin and End Snapshot Ids
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt Begin Snapshot Id specified: &&begin_snap
prompt
prompt End   Snapshot Id specified: &&end_snap
prompt


--
--  Set up the snapshot-related binds
```

```
variable bid        number;
variable eid        number;
begin
  :bid       :=  &begin_snap;
  :eid       :=  &end_snap;
end;
/

prompt


--
--  Error reporting

whenever sqlerror exit;
declare

  cursor cspid(vspid dba_hist_snapshot.snap_id%type) is
     select end_interval_time
          , startup_time
       from dba_hist_snapshot
      where snap_id         = vspid
        and instance_number = :inst_num
        and dbid            = :dbid;

  bsnapt  dba_hist_snapshot.end_interval_time%type;
  bstart  dba_hist_snapshot.startup_time%type;
  esnapt  dba_hist_snapshot.end_interval_time%type;
  estart  dba_hist_snapshot.startup_time%type;

begin

  -- Check Begin Snapshot id is valid, get corresponding instance startup time
  open cspid(:bid);
  fetch cspid into bsnapt, bstart;
  if cspid%notfound then
    raise_application_error(-20200,
      'Begin Snapshot Id '||:bid||' does not exist for this database/instance');
  end if;
  close cspid;

  -- Check End Snapshot id is valid and get corresponding instance startup time
  open cspid(:eid);
  fetch cspid into esnapt, estart;
  if cspid%notfound then
    raise_application_error(-20200,
      'End Snapshot Id '||:eid||' does not exist for this database/instance');
  end if;
  if esnapt <= bsnapt then
    raise_application_error(-20200,
      'End Snapshot Id '||:eid||' must be greater than Begin Snapshot Id '||:bid);
  end if;
  close cspid;

  -- Check startup time is same for begin and end snapshot ids
  if ( bstart != estart) then
    raise_application_error(-20200,
```

```
      'The instance was shutdown between snapshots '||:bid||' and '||:eid);
  end if;

end;
/
whenever sqlerror continue;


-- Undefine substitution variables
undefine dbid
undefine inst_num
undefine num_days
undefine begin_snap
undefine end_snap
undefine db_name
undefine inst_name
```