oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（17）awrload.sql

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（17）awrload.sql

```
#AWR LOAD
#SQL/Plus script to help users load data into the AWR from
Rem      a dump file.
#

Rem
Rem $Header: awrload.sql 06-apr-2005.18:35:13 mlfeng Exp $
Rem
Rem awrload.sql
Rem
Rem Copyright (c) 2004, 2005, Oracle. All rights reserved.
Rem
Rem    NAME
Rem      awrload.sql - AWR LOAD
Rem
Rem    DESCRIPTION
Rem      SQL/Plus script to help users load data into the AWR from
Rem      a dump file.
Rem
Rem    NOTES
Rem      User must be connected as SYS to run this SQL/Plus script.
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mlfeng      03/01/05 - add disclaimer for support, create staging
Rem    mlfeng      06/01/04 - mlfeng_awr_import_export
Rem    mlfeng      05/17/04 - Created
Rem

--
--  Use the default directory name, file name, schema
-- define directory_name = ’DATA_PUMP_DIR’
-- define file_name      = ’awrdat’
-- define schema_name    = ’AWR_STAGE’
-- define default_tablespace = ''
-- define temporary_tablespace = ''
--

set echo off heading on underline on verify off
set feedback off linesize 80 termout on;
whenever sqlerror exit;

prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt  Disclaimer: This SQL/Plus script should only be called under
prompt              the guidance of Oracle Support.
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt
prompt ~~~~~~~~~~
prompt  AWR LOAD
prompt ~~~~~~~~~~
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt ~  This script will load the AWR data from a dump file. The   ~
```

```
prompt ~  script will prompt users for the following information:    ~
prompt ~     (1) name of directory object                            ~
prompt ~     (2) name of dump file                                   ~
prompt ~     (3) staging schema name to load AWR data into           ~
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

--
-- Ask User for Directory Name
--

prompt
prompt Specify the Directory Name
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~

set heading on;
column dirname format a30 heading 'Directory Name'
column dirpath format a49 heading 'Directory Path'

select directory_name dirname, directory_path dirpath
  from DBA_DIRECTORIES
 order by directory_name;

set termout off;
column dflt_dir new_value dflt_dir noprint;
select ''  dflt_dir from dual;
set termout on;

prompt
prompt Choose a Directory Name from the list above (case-sensitive).
prompt

set heading off;
column directory_name new_value directory_name noprint;
select 'Using the dump directory: ' || nvl('&&directory_name','&dflt_dir')
     , nvl('&&directory_name','&dflt_dir') directory_name
  from sys.dual;



variable dmpdir  varchar2(30);
variable dmppath varchar2(4000)

declare


  cursor dirpath (dirname varchar2) is
    select directory_path
      from dba_directories
      where directory_name = dirname;

begin
  :dmpdir  := '&directory_name';

   /* select the directory path into a variable */
   open dirpath(:dmpdir);

   fetch dirpath into :dmppath;

   if (dirpath%NOTFOUND) then
     RAISE_APPLICATION_ERROR(-20103,
```

```
                             'directory name ''' || :dmpdir ||
                              ''' is invalid', TRUE);
   end if;

   close dirpath;
end;
/



set termout off;
column dflt_name new_value dflt_name noprint;
select ''  dflt_name from dual;
set termout on;

prompt
prompt Specify the Name of the Dump File to Load
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt Please specify the prefix of the dump file (.dmp) to load:
prompt

set heading off;
column file_name new_value file_name noprint;
select 'Loading from the file name: ' || nvl('&&file_name','&dflt_name')
        || '.dmp'
      , nvl('&&file_name','&dflt_name') file_name
  from sys.dual;

variable dmpfile varchar2(30);

begin
  :dmpfile := '&file_name';
end;
/

set termout off;
column dflt_schema new_value dflt_schema noprint;
select 'AWR_STAGE'  dflt_schema from dual;
set termout on;

prompt
prompt Staging Schema to Load AWR Snapshot Data
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt The next step is to create the staging schema
prompt where the AWR snapshot data will be loaded.
prompt After loading the data into the staging schema,
prompt the data will be transferred into the AWR tables
prompt in the SYS schema.
prompt
prompt
prompt The default staging schema name is &dflt_schema..
prompt To use this name, press <return> to continue, otherwise enter
prompt an alternative.
prompt

set heading off;
column schema_name new_value schema_name noprint;
select 'Using the staging schema name: ' || nvl('&&schema_name','&dflt_schema')
      , nvl('&&schema_name','&dflt_schema') schema_name
```

```
  from sys.dual;

variable schname varchar2(30);
variable schcount number;

/* check if schema already exists */
declare
  cursor schemas (schname varchar2) is
    select count(*) schcount
      from dba_users
      where username = schname
      order by username;

begin
  :schname := '&schema_name';

   /* select the directory path into a variable */
   open schemas(:schname);

   fetch schemas into :schcount;

   if (:schcount > 0) then
     RAISE_APPLICATION_ERROR(-20104,
                             'schema name ''' || :schname ||
                              ''' already exists', TRUE);
   end if;

   close schemas;
end;
/

prompt
prompt Choose the Default tablespace for the &schema_name user
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt Choose the &schema_name users's default tablespace.  This is the
prompt tablespace in which the AWR data will be staged.

set heading on
column db_default format a18 heading 'DEFAULT TABLESPACE'
select tablespace_name, contents
     , decode(tablespace_name,'SYSAUX','*') db_default
  from sys.dba_tablespaces
 where tablespace_name <> 'SYSTEM'
   and contents = 'PERMANENT'
   and status = 'ONLINE'
 order by tablespace_name;
set heading off

prompt
prompt Pressing <return> will result in the recommended default
prompt tablespace (identified by *) being used.
prompt

col default_tablespace new_value default_tablespace noprint
select 'Using tablespace '||
       upper(nvl('&&default_tablespace','SYSAUX'))||
       ' as the default tablespace for the &&schema_name.'
     , nvl('&default_tablespace','SYSAUX') default_tablespace
```

```
  from sys.dual;



begin
  if upper('&&default_tablespace') = 'SYSTEM' then
    raise_application_error(-20105, 'Load failed - SYSTEM tablespace ' ||
                                    'specified for DEFAULT tablespace');
  end if;
end;
/

prompt
prompt
prompt Choose the Temporary tablespace for the &&schema_name user
prompt ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
prompt Choose the &schema_name user's temporary tablespace.

set heading on
column db_default format a23 heading 'DEFAULT TEMP TABLESPACE'
select t.tablespace_name, t.contents
     , decode(dp.property_name,'DEFAULT_TEMP_TABLESPACE','*') db_default
  from sys.dba_tablespaces t
     , sys.database_properties dp
 where t.contents           = 'TEMPORARY'
   and t.status             = 'ONLINE'
   and dp.property_name(+)  = 'DEFAULT_TEMP_TABLESPACE'
   and dp.property_value(+) = t.tablespace_name
 order by tablespace_name;

set heading off

prompt
prompt Pressing <return> will result in the database's default temporary
prompt tablespace (identified by *) being used.
prompt

col temporary_tablespace new_value temporary_tablespace noprint
select 'Using tablespace '||
       nvl('&&temporary_tablespace',property_value)||
       ' as the temporary tablespace for &&schema_name.'
     , nvl('&&temporary_tablespace',property_value) temporary_tablespace
  from database_properties
 where property_name='DEFAULT_TEMP_TABLESPACE';

begin
  if upper('&&temporary_tablespace') = 'SYSTEM' then
    raise_application_error(-20106, 'Load failed - SYSTEM tablespace ' ||
                                    'specified for TEMPORARY tablespace');
  end if;
end;
/

set heading off

prompt
prompt
prompt ... Creating &&schema_name user
```

```
create user &&schema_name
  identified by &&schema_name
  default tablespace &&default_tablespace
  temporary tablespace &&temporary_tablespace;

alter user &&schema_name quota unlimited on &&default_tablespace;

prompt

set termout on;

set serveroutput on;
exec dbms_output.enable(500000);
set termout on;

column loc    format a80 newline;
column locend format a80;

declare
  begpos   NUMBER;
  numchar  NUMBER  := 74;

begin
  dbms_output.put_line('|');
  dbms_output.put_line('| ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
  dbms_output.put_line('|  Loading the AWR data from the following  ');
  dbms_output.put_line('|  directory/file:                          ');

  begpos := 1;
  WHILE (begpos <= length(:dmppath)) LOOP
    dbms_output.put_line('|   ' || substr(:dmppath, begpos, numchar));
    begpos := begpos + numchar;
  END LOOP;

  dbms_output.put_line('|   ' || :dmpfile || '.dmp');
  dbms_output.put_line('| ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
  dbms_output.put_line('|');
  dbms_output.put_line('|  *** AWR Load Started ...');
  dbms_output.put_line('|');
  dbms_output.put_line('|  This operation will take a few moments. The ');
  dbms_output.put_line('|  progress of the AWR load operation can be ');
  dbms_output.put_line('|  monitored in the following directory/file: ');

  begpos := 1;
  WHILE (begpos <= length(:dmppath)) LOOP
    dbms_output.put_line('|   ' || substr(:dmppath, begpos, numchar));
    begpos := begpos + numchar;
  END LOOP;

  dbms_output.put_line('|   ' || :dmpfile || '.log');
  dbms_output.put_line('|');
end;
/

whenever sqlerror continue;
set heading off;
set linesize 110 pagesize 50000;
set echo off;
```

```
set feedback off;
set termout on;

begin
  /* call PL/SQL routine to load the data into the staging schema */
  dbms_swrf_internal.awr_load(schname  => :schname,
                              dmpfile  => :dmpfile,
                              dmpdir   => :dmpdir);
end;
/

begin
  /* call PL/SQL routine to move the data into AWR */
  dbms_swrf_internal.move_to_awr(schname => :schname);
end;
/

prompt ... Dropping &&schema_name user

drop user &&schema_name cascade;

prompt
prompt End of AWR Load

undefine directory_name
undefine file_name
undefine schema_name
undefine default_tablespace
undefine temporary_tablespace
```