

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能 (6) addmrpti.sql

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能 (6) addmrpti.sql

```
#SQL*Plus script that prompts for dbid and instance_number
#to run ADDM analysis on a pair on AWR snapshots and
#display the textual ADDM report of the analysis.

Rem
Rem $Header: addmrpti.sql 22-nov-2004.13:49:16 adagarwa Exp $
Rem
Rem addmrpti.sql
Rem
Rem Copyright (c) 2003, 2004, Oracle. All rights reserved.
Rem
Rem NAME
Rem     addmrpti.sql - SQL*Plus script that prompts for dbid and instance_number
Rem                     to run ADDM analysis on a pair on AWR snapshots and
Rem                     display the textual ADDM report of the analysis.
Rem
Rem DESCRIPTION
Rem     This SQL*Plus script can be used to run ADDM on any two AWR snapshots provided
Rem     the two snapshots were taken by the same instance.
Rem     This script can be used to run ADDM on:
Rem         snapshots taken by the current instance,
Rem         snapshots taken by remote instances in a RAC environment, and
Rem         snapshots imported from other databases.
Rem     If you are trying to run ADDM only on snapshots from the current instance,
Rem     addmrpti.sql is easier to use and does not prompt for dbid and instance_number.
Rem
Rem NOTES
Rem     Prompts for a dbid and an instance_number,
Rem     Displays the snapshots taken in the past &num_days,
Rem     Prompts for a pair of AWR snapshots,
Rem     Runs ADDM across those snapshots and
Rem     Displays and spools the textual ADDM report of the analysis.
Rem
Rem     If you want to use this script in an non-interactive fashion,
Rem     do something similar to the following:
Rem
Rem     define dbid          = 1234567890;
Rem     define inst_num     = 1;
Rem     define num_days     = 3;
Rem     define begin_snap   = 10;
Rem     define end_snap     = 11;
Rem     define report_name  = /tmp/addm_report_10_11.txt
Rem     @@?/rdbms/admin/addmrpti
Rem
Rem
Rem MODIFIED (MM/DD/YY)
Rem adagarwa 11/22/04 - obtain report name from awrinpnm.sql
Rem veeve    02/23/04 - show report_name at the end
Rem pbelknap 10/15/03 - swrf reporting to html in pl/sql module
Rem veeve    10/13/03 - more cleanup
Rem pbelknap 10/09/03 - moving parameters to addmrpt, set default vals
Rem pbelknap 10/02/03 - change swrfinput to awrinput
Rem veeve    10/02/03 - created addmrpti.sql
```

```

Rem    veeve        10/01/03 - use swrfinput.sql
Rem    veeve        09/06/03 - veeve_addm_production_1
Rem    veeve        09/04/03 - created addmrpt.sql
Rem

--

-- Specify default values for the report name prefix and extension.
-- These would be ignored, if the variable 'report_name' is defined.
define report_name_prefix    = 'addmrpt_';
define report_name_extension = '.txt';

set heading on echo off feedback off verify off underline on timing off;

--

-- First of all, get the inputs from awrinput.sql.
-- The bind variables :dbid, :inst_num, :bid, :eid
-- These will be defined in awrinput.sql and passed over.
@@awrinput.sql
-- Get the substitution variable &report_name
@@awrinputnm.sql &report_name_prefix &report_name_extension

set pagesize 0;
set heading off echo off feedback off verify off;

variable task_name  varchar2(40);

prompt
prompt
prompt Running the ADDM analysis on the specified pair of snapshots ...
prompt

begin
  declare
    id number;
    name varchar2(100);
    descr varchar2(500);
  BEGIN
    name := '';
    descr := 'ADDM run: snapshots [' || :bid || ', '
              || :eid || '], instance ' || :inst_num
              || ', database id ' || :dbid;

    dbms_advisor.create_task('ADDM', id, name, descr, null);

    :task_name := name;

    -- set time window
    dbms_advisor.set_task_parameter(name, 'START_SNAPSHOT', :bid);
    dbms_advisor.set_task_parameter(name, 'END_SNAPSHOT', :eid);

    -- set instance number
    dbms_advisor.set_task_parameter(name, 'INSTANCE', :inst_num);

    -- set dbid
    dbms_advisor.set_task_parameter(name, 'DB_ID', :dbid);

    -- execute task
    dbms_advisor.execute_task(name);

```

```

    end;
end;
/

prompt
prompt Generating the ADDM report for this analysis ...
prompt
prompt

spool &report_name;

set long 1000000 pagesize 0 longchunksize 1000
column get_clob format a80

select dbms_advisor.get_task_report(:task_name, 'TEXT', 'TYPICAL')
from   sys.dual;

spool off;
prompt
prompt End of Report
prompt Report written to &report_name.

set termout off;

clear columns sql;
tttitle off;
btitle off;
repfooter off;

-- Restore the default SQL*Plus settings
set pagesize 14;
set heading on echo off feedback 6 verify on underline on timing off;
set long 80 longchunksize 80
set termout on

-- Undefine all variables that need to be set before calling this script
undefine dbid
undefine inst_num
undefine num_days
undefine begin_snap
undefine end_snap

-- Undefine report_name created in awrinput.sql
undefine report_name

undefine report_name_prefix
undefine report_name_extension

whenever sqlerror continue;
--
-- End of script file;

```