

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能 (20) awrsqrpi.sql

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能 (20) awrsqrpi.sql

```
#Workload Repository SQL Report Instance
#SQL*Plus command file to report on differences between values
Rem      recorded in two snapshots
Rem
Rem      This script requests the user for the dbid, instance number
Rem      and the sql id, before producing a report for a particular
Rem      sql statement in this instance.
#

Rem
Rem $Header: awrsqrpi.sql 05-jan-2005.14:23:20 adagarwa Exp $
Rem
Rem awrsqrpi.sql
Rem
Rem Copyright (c) 2004, 2005, Oracle. All rights reserved.
Rem
Rem      NAME
Rem      awrsqrpi.sql - Workload Repository SQL Report Instance
Rem
Rem      DESCRIPTION
Rem      SQL*Plus command file to report on differences between values
Rem      recorded in two snapshots
Rem
Rem      This script requests the user for the dbid, instance number
Rem      and the sql id, before producing a report for a particular
Rem      sql statement in this instance.
Rem
Rem      NOTES
Rem      Run as SYSDBA. Generally this script should be invoked by awrsqrpt,
Rem      unless you want to pick a database other than the default.
Rem
Rem      If you want to use this script in an non-interactive fashion,
Rem      without executing the script through awrrpt, then
Rem      do something similar to the following:
Rem
Rem      define inst_num      = 1;
Rem      define num_days     = 3;
Rem      define inst_name    = 'Instance';
Rem      define db_name      = 'Database';
Rem      define dbid         = 4;
Rem      define sql_id       = 'abcdefabcdefa';
Rem      define begin_snap   = 10;
Rem      define end_snap     = 11;
Rem      define report_type  = 'text';
Rem      define report_name  = /tmp/awr_sql_report_10_11.txt
Rem      @@?/rdbms/admin/awrsqrpi.sql
Rem
Rem      MODIFIED      (MM/DD/YY)
Rem      adagarwa      01/05/05 - adagarwa_awr_sql_rpt
Rem      adagarwa      09/08/04 - Created
Rem
set echo off;
```

```

-- *****
-- Customer-customizable report settings
-- Change these variables to run a report on different statistics
-- *****
-- The default number of days of snapshots to list when displaying the
-- list of snapshots to choose the begin and end snapshot Ids from.
--
-- List all snapshots
-- define num_days = '';
--
-- List no (i.e. 0) snapshots
-- define num_days = 0;
--
-- List past 3 day's snapshots
-- define num_days = 3;
--
-- Reports can be printed in text or html, and you must set the report_type
-- in addition to the report_name
--
-- Issue Report in Text Format
-- define report_type='text';
--
-- Issue Report in HTML Format
--define report_type='html';

-- Optionally, set the snapshots for the report. If you do not set them,
-- you will be prompted for the values.
-- define begin_snap = 10;
-- define end_snap = 11;

-- Optionally, set the name for the report itself
-- define report_name = 'awrrpt_1_10_11.html'

-- Set the sqlid to be analyzed
-- define sql_id = 'abcdefabcdefa'

-- *****
-- End customer-customizable settings
-- *****

set veri off;
set feedback off;

variable rpt_options number;

-- Add new options here
-- option settings

--
-- Find out if we are going to print report to html or to text
prompt
prompt Specify the Report Type
prompt ~~~~~
prompt Would you like an HTML report, or a plain text report?

```

```
prompt Enter 'html' for an HTML report, or 'text' for plain text
prompt Defaults to 'html'
```

```
column report_type new_value report_type;
set heading off;
select 'Type Specified: ',lower(nvl('&&report_type','html')) report_type from dual;
set heading on;
```

```
set termout off;
-- Set the extension based on the report_type
column ext new_value ext;
select '.html' ext from dual where lower('&&report_type') <> 'text';
select '.txt' ext from dual where lower('&&report_type') = 'text';
set termout on;
```

```
-- Get the common input
```

```
@@awrinput.sql
```

```
prompt
```

```
-- Get the SQL ID from the user
```

```
prompt
```

```
prompt Specify the SQL Id
```

```
prompt ~~~~~
```

```
prompt SQL ID specified: &sql_id
```

```
-- Assign value to bind variable
```

```
variable sqlid VARCHAR2(13);
```

```
begin
```

```
    :sqlid := '&sql_id';
```

```
end;
```

```
/
```

```
whenever sqlerror exit;
```

```
declare
```

```
cursor csqlid(vsqlid dba_hist_sqlstat.sql_id%type) is
```

```
    select sql_id
```

```
        from dba_hist_sqlstat
```

```
    where snap_id > :bid
```

```
        and snap_id <= :eid
```

```
        and instance_number = :inst_num
```

```
        and dbid = :dbid
```

```
        and sql_id = vsqlid;
```

```
    inpsqlid dba_hist_sqlstat.sql_id%type;
```

```
begin
```

```
-- Check if the sqlid is valid. It must contain an entry in the
```

```
-- DBA_HIST_SQLSTAT table for the specified sqlid
```

```
open csqlid(:sqlid);
```

```
fetch csqlid into inpsqlid;
```

```
if csqlid%notfound then
```

```
    raise_application_error(-20025,
```

```
    'SQL ID '||:sqlid||' does not exist for this database/instance');
```

```
end if;
```

```

end;
/

whenever sqlerror continue;

-- Get the name of the report.
@@awrinpm.sql 'awrsqldrpt_' &&ext

set termout off;
-- set report function name and line size
column fn_name new_value fn_name noprint;
select 'awr_sql_report_text' fn_name from dual where lower('&report_type') = 'text';
select 'awr_sql_report_html' fn_name from dual where lower('&report_type') <> 'text';

column lnsz new_value lnsz noprint;
-- Line size for Text Reports: 120
select '120' lnsz from dual where lower('&report_type') = 'text';
-- Line size for HTML Reports: 1500
select '1500' lnsz from dual where lower('&report_type') <> 'text';

set linesize &lnsz;
set termout on;
spool &report_name;
prompt

select 'WARNING: timed_statistics setting changed between begin/end snaps: TIMINGS ARE INVALID'
from dual
where not exists
  (select null
   from dba_hist_parameter b
   , dba_hist_parameter e
  where b.snap_id      = :bid
    and e.snap_id      = :eid
    and b.dbid         = :dbid
    and e.dbid         = :dbid
    and b.instance_number = :inst_num
    and e.instance_number = :inst_num
    and b.parameter_hash = e.parameter_hash
    and b.parameter_name = 'timed_statistics'
    and b.value         = e.value);

select output from table(dbms_workload_repository.&fn_name( :dbid,
                                                         :inst_num,
                                                         :bid, :eid,
                                                         :sqlid,
                                                         :rpt_options ));

spool off;

prompt Report written to &report_name.

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

```

```
set linesize 78 termout on feedback 6 heading on;
```

```
-- Undefine report name (created in awrinpnm.sql)
```

```
undefine report_name
```

```
-- Undefine sql_id
```

```
undefine sql_id
```

```
undefine report_type
```

```
undefine ext
```

```
undefine fn_name
```

```
undefine lnsz
```

```
undefine NO_OPTIONS
```

```
undefine top_n_events
```

```
undefine num_days
```

```
undefine top_n_sql
```

```
undefine top_pct_sql
```

```
undefine sh_mem_threshold
```

```
undefine top_n_segstat
```

```
whenever sqlerror continue;
```

```
--
```

```
-- End of script file;
```