oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（23）bsln_pkgdef.sql

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（23）bsln_pkgdef.sql

```
#Baseline packages (creation).
#This script defines the packaged procedures and functions required
Rem      for metric baseline support.
#

Rem
Rem $Header: bsln_pkgdef.sql 11-may-2005.13:24:02 jberesni Exp $
Rem
Rem bsln_pkgdef.sql
Rem
Rem Copyright (c) 2004, 2005, Oracle. All rights reserved.
Rem
Rem    NAME
Rem      bsln_pkgdef.sql - Baseline packages (creation).
Rem
Rem    DESCRIPTION
Rem      This script defines the packaged procedures and functions required
Rem      for metric baseline support.
Rem
Rem    NOTES
Rem
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    jberesni    05/11/05 - subinterval_code non-deterministic
Rem    jberesni    03/28/05 -
Rem    jberesni    01/21/05 - refactor
Rem    jberesni    09/23/04 - fix 3910279
Rem    jberesni    08/10/04 - daynight
Rem    jberesni    08/01/04 - compute_all and set_all
Rem    jberesni    07/28/04 - restructure
Rem    jberesni    07/23/04 - misc fixes
Rem    jberesni    07/15/04 - candidate1
Rem    jsoule      05/19/04 - add exceptions
Rem    jsoule      05/18/04 - external constants
Rem    jsoule      05/17/04 - update
Rem    jsoule      05/10/04 - Created
Rem

create or replace
package mgmt_bsln
-----------------------------
-- DB Control deployment 0
-----------------------------
as
   ---------------------------------------------------------------------------
   --
   --    externally visible constants and subtypes
   --
   ---------------------------------------------------------------------------
   K_BSLN_XX constant mgmt_bsln_baselines.subinterval_key%type := 'XX';
   K_BSLN_HX constant mgmt_bsln_baselines.subinterval_key%type := 'HX';
   K_BSLN_XD constant mgmt_bsln_baselines.subinterval_key%type := 'XD';
   K_BSLN_HD constant mgmt_bsln_baselines.subinterval_key%type := 'HD';
```

```
K_BSLN_XW constant mgmt_bsln_baselines.subinterval_key%type := 'XW';
K_BSLN_HW constant mgmt_bsln_baselines.subinterval_key%type := 'HW';
K_BSLN_NW constant mgmt_bsln_baselines.subinterval_key%type := 'NW';
K_BSLN_ND constant mgmt_bsln_baselines.subinterval_key%type := 'ND';
K_BSLN_NX constant mgmt_bsln_baselines.subinterval_key%type := 'NX';

K_DEFAULT_KEY_VALUE constant varchar2(10) := ' ';
K_DEFAULT_NUM_OCCURS constant number := 1;

K_FAIL_ACTION_UNSET constant varchar2(16) := 'UNSET';
K_FAIL_ACTION_PRESERVE constant varchar2(16) := 'PRESERVE';

K_METHOD_SIGLVL constant mgmt_bsln_threshold_parms.threshold_method%type := 'SIGLVL';
K_METHOD_PCTMAX constant mgmt_bsln_threshold_parms.threshold_method%type := 'PCTMAX';

K_SIGLVL_95   constant number := 0.95;
K_SIGLVL_99   constant number := 0.99;
K_SIGLVL_999  constant number := 0.999;
K_SIGLVL_9999 constant number := 0.9999;

K_SOURCE_EM constant mgmt_bsln_datasources.source_type%type := 'EM';
K_SOURCE_DB constant mgmt_bsln_datasources.source_type%type := 'DB';

K_TRUE  constant integer := 1;
K_FALSE constant integer := 0;

K_BSLN_STATIC  constant mgmt_bsln_baselines.type%type := 'S';
K_BSLN_ROLLING constant mgmt_bsln_baselines.type%type := 'R';

K_STATUS_ACTIVE constant mgmt_bsln_baselines.status%type := 'ACTIVE';
K_STATUS_INACTIVE constant mgmt_bsln_baselines.status%type := 'INACTIVE';

-------------------------------------------------------------------------
--
--         package exception declarations
--
-------------------------------------------------------------------------
X_INVALID_BASELINE         constant number := -20101;
X_INVALID_INTERVAL         constant number := -20102;
X_DATASOURCE_NOT_FOUND     constant number := -20103;
X_INVALID_THRESHOLD_METHOD constant number := -20104;
X_INVALID_METRIC           constant number := -20105;
X_BASELINE_NOT_FOUND       constant number := -20106;
X_SOURCE_CONFLICT          constant number := -20107;
X_NOT_SUPPORTED            constant number := -20108;
X_BSLNTHR_ERROR            constant number := -20109;

INVALID_BASELINE           exception;
INVALID_INTERVAL           exception;
DATASOURCE_NOT_FOUND       exception;
INVALID_THRESHOLD_METHOD   exception;
INVALID_METRIC             exception;
BASELINE_NOT_FOUND         exception;
SOURCE_CONFLICT            exception;
NOT_SUPPORTED              exception;
BSLNTHR_ERROR              exception;

PRAGMA EXCEPTION_INIT(INVALID_BASELINE, -20101);
```

```
PRAGMA EXCEPTION_INIT(INVALID_INTERVAL, -20102);
PRAGMA EXCEPTION_INIT(DATASOURCE_NOT_FOUND, -20103);
PRAGMA EXCEPTION_INIT(INVALID_THRESHOLD_METHOD, -20104);
PRAGMA EXCEPTION_INIT(INVALID_METRIC, -20105);
PRAGMA EXCEPTION_INIT(BASELINE_NOT_FOUND, -20106);
PRAGMA EXCEPTION_INIT(SOURCE_CONFLICT, -20107);
PRAGMA EXCEPTION_INIT(NOT_SUPPORTED, -20108);
PRAGMA EXCEPTION_INIT(BSLNTHR_ERROR, -20109);


--------------------------------------------------------------------------------
--
-- package subtypes
--
--------------------------------------------------------------------------------
subtype guid_t is mgmt_bsln_baselines.bsln_guid%type;
subtype subinterval_code_t is mgmt_bsln_statistics.subinterval_code%type;
subtype subinterval_key_t is mgmt_bsln_baselines.subinterval_key%type;
subtype key_value_t  is mgmt_bsln_datasources.key_value%type;
subtype fail_action_t is mgmt_bsln_threshold_parms.fail_action%type;
subtype threshold_method_t is mgmt_bsln_threshold_parms.THRESHOLD_METHOD%TYPE;
subtype param_value_t is mgmt_bsln_threshold_parms.critical_param%type;

-- deployment-specific subtype declaration
subtype alert_threshold_t is varchar2(256);
--------------------------------------------------------------------------------
--
--    utility modules
--
--------------------------------------------------------------------------------

function valid_key (subinterval_key_in subinterval_key_t)
return boolean;

function target_uid
      (target_guid_in  in guid_t)
return guid_t
DETERMINISTIC;

function target_uid
      (dbid_in         in mgmt_bsln_datasources.dbid%type
      ,instance_num_in in mgmt_bsln_datasources.instance_num%type)
return guid_t
DETERMINISTIC;

function this_target_uid
return guid_t;

function metric_uid
      (metric_guid_in in guid_t)
return guid_t
DETERMINISTIC;

function metric_uid
      (metric_id_in in mgmt_bsln_datasources.metric_id%type)
return guid_t
DETERMINISTIC;
```

```
function datasource_guid
      (target_uid_in in guid_t
      ,metric_uid_in in guid_t
      ,key_value_in  in key_value_t := K_DEFAULT_KEY_VALUE)
return guid_t
DETERMINISTIC;

function baseline_guid
      (target_uid_in in guid_t
      ,name_in       in mgmt_bsln_baselines.name%type)
return guid_t
DETERMINISTIC;

function stdhh24 (date_in in date)
return binary_integer;

function subinterval_code
      (subinterval_key_in in subinterval_key_t
      ,time_in            in date)
return subinterval_code_t;

function cached_subinterval_code
      (subinterval_key_in in subinterval_key_t
      ,time_in            in date)
return subinterval_code_t;

function target_source_type (target_uid_in in mgmt_bsln.guid_t)
return varchar2;

function baseline_is_active (bsln_guid_in in guid_t)
return boolean;

function datasource_rec(ds_guid_in in guid_t) RETURN mgmt_bsln_datasources%ROWTYPE;

function baseline_rec(bsln_guid_in in guid_t) RETURN mgmt_bsln_baselines%ROWTYPE;

--------------------------------------------------------------------------------
--
--    administration modules
--
--------------------------------------------------------------------------------
procedure update_moving_window
      (interval_days_in in number
      ,subinterval_key_in in subinterval_key_t
      ,target_uid_in    in guid_t := null
      );

procedure create_baseline_static
      (name_in           in mgmt_bsln_baselines.name%type
      ,interval_begin_in in date
      ,interval_end_in   in date
      ,subinterval_key_in in subinterval_key_t
      ,target_uid_in     in guid_t := null
      );

procedure create_baseline_rolling
      (name_in          in mgmt_bsln_baselines.name%type
      ,subinterval_key_in in subinterval_key_t
```

```
      ,interval_days_in in number
      ,target_uid_in    in guid_t := null
      );

procedure drop_baseline
      (name_in       in mgmt_bsln_baselines.name%type
      ,target_uid_in in guid_t := null
      );

procedure register_datasource
      (target_guid_in  in guid_t
      ,metric_guid_in  in guid_t
      ,key_value_in    in key_value_t := K_DEFAULT_KEY_VALUE
      );

procedure register_datasource
      (dbid_in          in mgmt_bsln_datasources.dbid%type
      ,instance_num_in  in mgmt_bsln_datasources.instance_num%type
      ,metric_id_in     in mgmt_bsln_datasources.metric_id%type
      );

function registered_ds_guid
      (target_guid_in  in guid_t
      ,metric_guid_in  in guid_t
      ,key_value_in    in key_value_t := K_DEFAULT_KEY_VALUE)
return guid_t;

function registered_ds_guid
      (dbid_in  in mgmt_bsln_datasources.dbid%type
      ,instance_num_in  in mgmt_bsln_datasources.instance_num%type
      ,metric_id_in in mgmt_bsln_datasources.metric_id%type)
return guid_t;

procedure deregister_datasource
      (target_guid_in in guid_t
      ,metric_guid_in in guid_t
      ,key_value_in   in key_value_t := K_DEFAULT_KEY_VALUE);

procedure deregister_datasource
      (dbid_in      in mgmt_bsln_datasources.dbid%type
      ,instance_num_in in mgmt_bsln_datasources.instance_num%type
      ,metric_id_in    in mgmt_bsln_datasources.metric_id%type);

procedure activate_baseline
      (name_in        in mgmt_bsln_baselines.name%type
      ,target_uid_in  in guid_t := null
      );

procedure deactivate_baseline
      (name_in  in mgmt_bsln_baselines.name%type
      ,target_uid_in  in  guid_t := null
      );

procedure unset_threshold_parameters
      (bsln_guid_in  in guid_t
      ,ds_guid_in    in guid_t
      );
```

```
   procedure set_threshold_parameters
      (bsln_guid_in        in guid_t
      ,ds_guid_in          in guid_t
      ,threshold_method_in in mgmt_bsln_threshold_parms.threshold_method%type
      ,warning_param_in    in mgmt_bsln_threshold_parms.warning_param%type
      ,critical_param_in   in mgmt_bsln_threshold_parms.critical_param%type
      ,num_occurs_in       in integer := K_DEFAULT_NUM_OCCURS
      ,fail_action_in      in fail_action_t := K_FAIL_ACTION_UNSET
      );

   ------------------------------------------------------------------------------
   --
   --    operational routines
   --
   ------------------------------------------------------------------------------
   procedure set_all_thresholds;
   procedure compute_all_statistics;
   ------------------------------------------------------------------------------
   --
   --    submit and drop jobs to compute and set thresholds
   --
   ------------------------------------------------------------------------------
   procedure submit_bsln_jobs;
   procedure delete_bsln_jobs;

   ------------------------------------------------------------------------------
   --
   --    new enable/disable API
   --
   ------------------------------------------------------------------------------
   procedure enable;
   procedure disable;
   function is_enabled return integer;

   ------------------------------------------------------------------------------
   --
   --    extraction cursor record and ref cursor types
   --
   ------------------------------------------------------------------------------
   type extract_rectype is record
      (datasource_guid  mgmt_bsln.guid_t
      ,bsln_guid        mgmt_bsln.guid_t
      ,subinterval_key  mgmt_bsln_baselines.subinterval_key%TYPE
      ,obs_time         date
      ,obs_value        number
      );

   type extract_cvtype is ref cursor return extract_rectype;
   ------------------------------------------------------------------------------
   --
   --    extract and compute statistics modules
   --
   ------------------------------------------------------------------------------
--   procedure compute_load_stats
--         (compute_date_in in date
--         ,bsln_guid_in in varchar2);

   function extract_compute_stats
```

```
      (extract_cv  in extract_cvtype
      ,compute_date_in  in date := SYSDATE)
return bsln_statistics_set
PIPELINED
CLUSTER extract_cv by (datasource_guid)
PARALLEL_ENABLE
   (PARTITION extract_cv BY HASH(datasource_guid));

function exptail_stats (observation_set_in  bsln_observation_set)
return bsln_statistics_set;

function compute_statistics
      (bsln_name_in      in mgmt_bsln_baselines.name%type
      ,interval_begin_in in date
      ,interval_end_in   in date
      ,subinterval_key_in in subinterval_key_t
      ,target_uid_in     in guid_t := null
      )
return bsln_statistics_set;

procedure load_statistics
      (statistics_set_in in bsln_statistics_set
      ,replace_flag_in in boolean := TRUE);

function data_and_model_OK
      (threshold_method_in in varchar2
      ,threshold_param_in  in  number
      ,sample_count_in  in number
      ,fit_quality_in   in number
      )
return integer;

-------------------------------------------------------------------------
--  record type to pass to new_threshold_value function
-------------------------------------------------------------------------
TYPE THR_rectype is RECORD
   (threshold_method   mgmt_bsln_threshold_parms.threshold_method%TYPE
   ,num_occurrences    mgmt_bsln_threshold_parms.num_occurrences%TYPE
   ,warning_param      mgmt_bsln_threshold_parms.warning_param%TYPE
   ,critical_param     mgmt_bsln_threshold_parms.critical_param%TYPE
   ,fail_action        mgmt_bsln_threshold_parms.fail_action%TYPE
   ,sample_count       number
   ,minval             number
   ,maxval             number
   ,pctile_95          number
   ,pctile_99          number
   ,pctile_999         number
   ,pctile_9999        number
   ,est_fit_quality    number
   ,est_sample_count   number
   );

procedure new_threshold_value
            (THR_rec_in     THR_rectype
            ,param_in       mgmt_bsln_threshold_parms.warning_param%TYPE
            ,value_inout in out alert_threshold_t);

-------------------------------------------------------------------------
```

```
    ------------------------------------------------------------------------
    --
    --    SLPA declarations for Design by Contract support
    --
    ------------------------------------------------------------------------
    ASSERTFAIL     EXCEPTION;
    ASSERTFAIL_C   CONSTANT INTEGER := -20999;
    PRAGMA EXCEPTION_INIT(ASSERTFAIL, -20999);
    PKGNAME_C      CONSTANT VARCHAR2(20) := 'MGMT_BSLN';
    ------------------------------------------------------------------------
end mgmt_bsln;
/
```