

## oracle 10g 研究ORACLE\_HOME rdbms admin 下的脚本的功能 (4) a1001000.sql

oracle 10g 研究ORACLE\_HOME rdbms admin 下的脚本的功能 (4)

#additional ANONYMOUS BLOCK dictionary upgrade.

#Upgrade Oracle RDBMS from 10.1.0 to the new release

Rem

Rem \$Header: a1001000.sql 21-jun-2005.16:48:21 nbhatt Exp \$

Rem

Rem a1001000.sql

Rem

Rem Copyright (c) 1999, 2005, Oracle. All rights reserved.

Rem

Rem NAME

Rem a1001000.sql - additional ANONYMOUS BLOCK dictionary upgrade.

Rem Upgrade Oracle RDBMS from 10.1.0 to the new release

Rem

Rem

Rem DESCRIPTION

Rem Additional upgrade script to be run during the upgrade of an

Rem 10.1.0 database to the new release.

Rem

Rem This script is called from u1001000.sql and a0902000.sql

Rem

Rem Put any anonymous block related changes here.

Rem Any dictionary create, alter, updates and deletes

Rem that must be performed before catalog.sql and catproc.sql go

Rem in c1001000.sql

Rem

Rem The upgrade is performed in the following stages:

Rem STAGE 1: steps to upgrade from 10.1 to 10.2

Rem STAGE 2: upgrade from 10.2 to the new release

Rem

Rem NOTES

Rem \* This script must be run using SQL\*PLUS.

Rem \* You must be connected AS SYSDBA to run this script.

Rem

Rem MODIFIED (MM/DD/YY)

Rem nbhatt 06/21/05 - upgrade must handle queue tables with errors

Rem dsampath 06/22/05 - 4449955 fix upgrade for sqltext signature

Rem dsampath 05/31/05 - change sqltext signature in dictionary tables

Rem kyagoub 04/06/05 - add desc message for new parameter resume\_filter

Rem lkaplan 05/11/05 - unlock\_table\_stats before delete\_table\_stats

Rem xan 03/20/05 - no auto stats collections for AQ tables

Rem qiwang 04/26/05 - logmnr ckpt upgrade: fill sparel column

Rem jmzhang 03/23/05 - move logstdby\$parameters to system ts

Rem gssmith 03/14/05 - Fix for bug 4236538

Rem kyagoub 01/05/05 - add description for two new R2 sqltune task

Rem parameters: COMMIT\_ROWS and LOCAL\_TIME\_LIMIT

Rem araghava 12/09/04 - 3989938: fix partitioned tables that had

Rem retention set at table level

Rem ysarig 12/09/04 - bug 3930796 - comment for ALERT\_QUE

Rem nshodhan 11/23/04 - bug 4020148

Rem weiwang 11/12/04 - fix LRG 1793661

Rem kyagoub 10/25/04 - sqlset: correct bind\_capture to N vs. Y

Rem kyagoub 10/07/04 - add bind\_data to sqlset\_plans table

```

Rem    weiwang      10/12/04 - create deq by condition view
Rem    ushaft       10/13/04 - add description values in wri$_adv%parameters tables
Rem    bpwang       10/07/04 - Bug 3880023: Remove orphaned entries in
Rem                                apply$_dest_obj_ops
Rem    nbhatt       08/30/04 - make aq downgrande idempotent
Rem    pbelknap     08/06/04 - sts changes for force_matching_signature
Rem    qyu          04/28/04 - #3334209: move lob segment for kottd$ and etc
Rem    kyagoub      08/01/04 - add direct_writes to sqlset
Rem    pbelknap     07/13/04 - chk for 904 error
Rem    rramkiss     07/13/04 - lrg-1714684
Rem    pbelknap     06/29/04 - move sqlt block from 'c' script
Rem    nbhatt       06/28/04 - recreate buffer view
Rem    sbalaram     05/25/04 - AQ: recreate base view, alter primary key
Rem    ajadams      06/22/04 - conditionally call dbms_logstdby.set_tablespace
Rem    jawilson     06/01/04 - changes for canonicalized subname in reg$
Rem    rvissapr     06/08/04 - fix bug 3282580 unlimited failed loginattempts
Rem    rvissapr     05/20/04 - dblink encoding - proj 5523
Rem    mtao         05/28/04 - old logstdby sessions: LOG_AUTO_DELETE false
Rem    liwong       02/21/04 - Fast column value evaluation
Rem    rgupta       04/26/04 - move logical standby tables to SYSAUX tblspc
Rem    rburns       03/22/04 - call 10.1 registry procedrue
Rem    mbrey        03/30/04 - CDC change source upgrade
Rem    gssmith      02/18/04 - Adding Access Advisor upgrade items
Rem    rburns       01/16/04 - rburns_add_10_1_updw_scripts
Rem    rburns       01/07/04 - Created
Rem

```

```

Rem =====
Rem BEGIN STAGE 1: upgrade from 10.1.0 to 10.2
Rem =====

```

```

Rem Insert PL/SQL blocks here

```

```

Rem =====
Rem Begin upgrade sqltext signature fields in various dict tables
Rem =====

```

```

DECLARE

```

```

    CURSOR sig_sql_cur IS
        SELECT  signature old_sig,
                flags force_flag
        FROM    sys.sql$;

    sig_rec sig_sql_cur%ROWTYPE;
    sqltext clob;
    new_sig number;

```

```

BEGIN

```

```

    OPEN sig_sql_cur;
    LOOP

```

```

        FETCH sig_sql_cur INTO sig_rec;
        EXIT WHEN sig_sql_cur%NOTFOUND OR sig_sql_cur%NOTFOUND IS NULL;

```

```

    BEGIN

```

```

        select sql_text into sqltext from sys.sql$text

```

```

        where signature = sig_rec.old_sig;

--compute new signature taking care of force flag
new_sig := sys.dbms_sqltune.sqltext_to_signature(sqltext,
                                                sig_rec.force_flag);

--left bit shift new signature by 1 bit to avoid collision with old
--signature
new_sig := new_sig + 18446744073709551616;

--update signature in sys.sql$, sys.sqlprof$, sys.sqlprof$desc,
--sys.sqlprof$attr and sys.sql$text
update sys.sql$
    set signature = new_sig where signature = sig_rec.old_sig;

update sys.sqlprof$
    set signature = new_sig where signature = sig_rec.old_sig;

update sys.sqlprof$desc
    set signature = new_sig where signature = sig_rec.old_sig;

update sys.sqlprof$attr
    set signature = new_sig where signature = sig_rec.old_sig;

update sys.sql$text
    set signature = new_sig where signature = sig_rec.old_sig;

commit;

EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
    --hash collision encountered due to bug in signature generation
    --rollback the transaction
    --delete the profile from all tables, dump the info to alert.log

    new_sig := new_sig - 18446744073709551616;

    sys.dbms_sqltune_internal.test_ksd_trace(
        2, 'Internal error: Mismatch in signature for SQL Profile, ' || '\n' ||
        'Upgrade failed for SQL statement : ' || sqltext || '\n' ||
        'Dropping the SQL profile from the dictionary' || '\n' ||
        'Old Signature: ' || sig_rec.old_sig || ' ' || '\n' ||
        'New Signature: ' || new_sig);

    delete from sys.sql$ where signature = sig_rec.old_sig;

    delete from sys.sqlprof$ where signature = sig_rec.old_sig;

    delete from sys.sqlprof$desc where signature = sig_rec.old_sig;

    delete from sys.sqlprof$attr where signature = sig_rec.old_sig;

    delete from sys.sql$text where signature = sig_rec.old_sig;

    commit;
END;
END LOOP;
CLOSE sig_sql_cur;

```

```

--now revert signature, shift right by 1 bit and compute the hash
OPEN sig_sql_cur;
LOOP

    FETCH sig_sql_cur INTO sig_rec;
    EXIT WHEN sig_sql_cur%NOTFOUND OR sig_sql_cur%NOTFOUND IS NULL;

BEGIN
    new_sig := sig_rec.old_sig - 18446744073709551616;

    update sys.sql$
        set signature = new_sig
        where signature = sig_rec.old_sig;
    commit;

    update sys.sql$
        set nhash = mod(new_sig, 4294967296)
        where signature = new_sig;

    update sys.sqlprof$
        set signature = new_sig
        where signature = sig_rec.old_sig;
    commit;

    update sys.sqlprof$
        set nhash = mod(new_sig, 4294967296)
        where signature = new_sig;

    update sys.sqlprof$desc
        set signature = new_sig
        where signature = sig_rec.old_sig;

    update sys.sqlprof$attr
        set signature = new_sig
        where signature = sig_rec.old_sig;

    update sys.sql$text
        set signature = new_sig
        where signature = sig_rec.old_sig;
    commit;

END;

END LOOP;
CLOSE sig_sql_cur;

END;
/

Rem =====
Rem End upgrade sqltext signature fields in various dict tables
Rem =====

Rem ===== Beginning of STREAMS upgrade =====
DECLARE

```

```

vt sys.re$variable_type_list;
BEGIN
vt := sys.re$variable_type_list(
  sys.re$variable_type('DML', 'SYS.LCR$_ROW_RECORD',
    'SYS.DBMS_STREAMS_INTERNAL.ROW_VARIABLE_VALUE_FUNCTION',
    'SYS.DBMS_STREAMS_INTERNAL.ROW_FAST_EVALUATION_FUNCTION'),
  sys.re$variable_type('DDL', 'SYS.LCR$_DDL_RECORD',
    'SYS.DBMS_STREAMS_INTERNAL.DDL_VARIABLE_VALUE_FUNCTION',
    'SYS.DBMS_STREAMS_INTERNAL.DDL_FAST_EVALUATION_FUNCTION'),
  sys.re$variable_type(NULL, 'SYS.ANYDATA',
    NULL,
    'SYS.DBMS_STREAMS_INTERNAL.ANYDATA_FAST_EVAL_FUNCTION'));

dbms_rule_adm.alter_evaluation_context(
  evaluation_context_name=>'SYS.STREAMS$_EVALUATION_CONTEXT',
  variable_types=>vt);
EXCEPTION WHEN OTHERS THEN
  IF SQLCODE = -24150 THEN
    -- suppress evaluation context does not exist error to minimize
    -- unwanted noise during upgrade.
    NULL;
  ELSE
    RAISE;
  END IF;
END;
/

declare
queue_name          varchar2(128);
canon_qname         varchar2(128);
subscriber_name     varchar2(128);
canon_sname         varchar2(128);
pos                 number;
cursor reg_cur is select subscription_name from sys.reg$ where
  namespace = dbms_aq.namespace_aq for update of subscription_name;
begin
  for reg in reg_cur loop
    pos := INSTR(reg.subscription_name, ':');
    IF pos != 0 THEN
      queue_name := SUBSTR(reg.subscription_name, 1, pos-1);
      subscriber_name := SUBSTR(reg.subscription_name, pos+1);
      dbms_utility.canonicalize(queue_name, canon_qname, 128);
      canon_sname := UPPER(REPLACE(subscriber_name, '"'));
      update sys.reg$ set subscription_name = canon_qname || ':' || '"' ||
        canon_sname || '"' where current of reg_cur;
    ELSE
      dbms_utility.canonicalize(reg.subscription_name, canon_qname, 128);
      update sys.reg$ set subscription_name = canon_qname where current of
        reg_cur;
    END IF;
  end loop;
end;
/

-- Bug 3880023: Some obj#s will not exist (if the table was dropped),
-- delete these rows as they are meaningless in 10.1+
BEGIN
  DELETE FROM sys.apply$_dest_obj_ops WHERE sname IS NULL AND oname IS NULL;

```

```

COMMIT;

END;

/

Rem Bug 4020148
Rem Update logminer session_attr to reflect flags used by 10g code
Rem Only update sessions created by streams capture in 9.2
Rem
Rem Remove KRVX_SESSION_RECORD_GLOBALNAME flag
UPDATE system.logmnr_session$ x
    SET x.session_attr = x.session_attr - 1073741824
    WHERE bitand(x.session_attr, 1073741824) = 1073741824
    AND EXISTS (SELECT c.logmnr_sid
                FROM sys.streams$_capture_process c
                WHERE c.logmnr_sid = x.session#);

COMMIT;

Rem Add KRVX_ATTACH_MULTIPLE flag
UPDATE system.logmnr_session$ x
    SET x.session_attr = x.session_attr + 128
    WHERE bitand(x.session_attr, 128) != 128
    AND EXISTS (SELECT c.logmnr_sid
                FROM sys.streams$_capture_process c
                WHERE c.logmnr_sid = x.session#);

COMMIT;

Rem Bug 4228711
Rem Update logmnr streams checkpoint table to populate spare1 column
Rem Assume no selective pruning has been done if spare1 is NULL
Rem
BEGIN
    UPDATE system.logmnr_restart_ckpt$ a
    SET     a.spare1 = (SELECT NVL(MAX(b.ckpt_scn), 0)
                        FROM   system.logmnr_restart_ckpt$ b
                        WHERE  b.ckpt_scn < a.ckpt_scn
                        and b.session# = a.session#)

    WHERE a.spare1 IS NULL;

    COMMIT;

END;

/

Rem ===== End of STREAMS upgrade =====

Rem=====
Rem Call component registry script for 10.1->10.2 populate
Rem=====

EXECUTE dbms_registry_sys.populate_101;

Rem=====
Rem Begin Advisor Framework upgrade items
Rem=====

Rem
Rem Simple updates
Rem

```

```

update sys.wri$_adv_recommendations
    set flags = 0
    where flags is null;

Rem
Rem Adjust journaling flags.
Rem
Rem We have to do three things here:
Rem     1. Change the datatype of the journal task parameter from a 1
Rem         to a 2. This also gives an indicator to whether this
Rem         upgrade has already been performed against the journal.
Rem     2. Reorder the original number values that went into the
Rem         type column of the journal table. The first 4 values are to
Rem         be reordered.
Rem     3. Change the numeric values in the journal task parameter
Rem         to string keywords to improve readability.
Rem

declare
    dtype binary_integer;
begin
    select datatype into dtype from sys.wri$_adv_def_parameters
        where name = 'JOURNALING';

    if dtype = 1 then
        update sys.wri$_adv_journal
            set type = decode(type, 1, 4, 2, 3, 3, 2, 4, 1, type);

        update sys.wri$_adv_def_parameters
            set value = decode(value, '0', 'UNUSED', '1', 'FATAL', '2', 'ERROR', '3', 'WARNING', '4', 'INFORMATION',
                                '5', 'INFORMATION2', '6', 'INFORMATION3', '7', 'INFORMATION4', '8', 'INFORMATION5',
                                'INFORMATION6'),
            datatype = 2
            where name = 'JOURNALING';

        update sys.wri$_adv_parameters
            set value = decode(value, '0', 'UNUSED', '1', 'FATAL', '2', 'ERROR', '3', 'WARNING', '4', 'INFORMATION',
                                '5', 'INFORMATION2', '6', 'INFORMATION3', '7', 'INFORMATION4', '8', 'INFORMATION5',
                                'INFORMATION6'),
            datatype = 2
            where name = 'JOURNALING';
    end if;
end;
/

Rem
Rem Move new default task parameters to existing tasks.
Rem
Rem This is a 3-level loop.
Rem     1. For each task, fetch its task id and advisor id
Rem     2. Fetch default task parameters
Rem     3. For each default task parameter, we fetch related tasks
Rem     4. For each task, we move a copy of the new default
Rem         task parameter to the task, if it does not already
Rem         exist in the task
Rem

declare

```

```

cursor task_cur IS
    SELECT id,advisor_id FROM sys.wri$_adv_tasks a;

cursor param_cur (id NUMBER) IS
    SELECT *
    FROM sys.wri$_adv_def_parameters a
    WHERE a.advisor_id in (id,0);

l_adv_id binary_integer;
l_task_id binary_integer;
l_cnt binary_integer;
param wri$_adv_def_parameters%ROWTYPE;
begin
    open task_cur;

    loop
        fetch task_cur into l_task_id,l_adv_id;
        exit when task_cur%NOTFOUND;

        open param_cur(l_adv_id);

        loop
            fetch param_cur INTO param;
            EXIT WHEN param_cur%NOTFOUND;

            select count(*) into l_cnt from sys.wri$_adv_parameters
            where name = param.name and task_id = l_task_id;

            if l_cnt = 0 then
                INSERT INTO sys.wri$_adv_parameters
                (task_id,name,datatype,value,flags,description)
                VALUES
                (l_task_id, param.name, param.datatype, param.value,
                param.flags, param.description);
            else
                update sys.wri$_adv_parameters
                set description = param.description,
                    flags = bitand(flags,6) + bitand(param.flags,9)
                where task_id = l_task_id and name = param.name;
            end if;
        end loop;

        close param_cur;
    end loop;

    close task_cur;
end;
/

Rem
Rem Update advisor-specific task parameters
Rem

declare
    l_task_id binary_integer;
    l_name varchar2(30);
    l_value varchar2(4000);
    cursor task_cur IS

```



```

SELECT a.id,b.name,b.value
FROM sys.wri$_adv_tasks a, sys.wri$_adv_parameters b
WHERE a.advisor_id in (2,6,7)
      and a.id = b.task_id
      and b.name in ('ACTION_LIST','MODULE_LIST','USERNAME_LIST',
                    'COMMENTED_FILTER_LIST');

begin
open task_cur;

loop
  fetch task_cur into l_task_id,l_name,l_value;
  exit when task_cur%NOTFOUND;

  if l_name = 'ACTION_LIST' then
    update sys.wri$_adv_parameters
      set value = l_value
      where name = 'VALID_ACTION_LIST'
        and task_id = l_task_id;
  elsif l_name = 'MODULE_LIST' then
    update sys.wri$_adv_parameters
      set value = l_value
      where name = 'VALID_MODULE_LIST'
        and task_id = l_task_id;
  elsif l_name = 'USERNAME_LIST' then
    update sys.wri$_adv_parameters
      set value = l_value
      where name = 'VALID_USERNAME_LIST'
        and task_id = l_task_id;
  elsif l_name = 'COMMENTED_FILTER_LIST' then
    update sys.wri$_adv_parameters
      set value = l_value
      where name = '_INVALID_SQLCOMMENTS_LIST'
        and task_id = l_task_id;
  end if;
end loop;

close task_cur;
end;
/

Rem
Rem add descriptions to advisor default parameters
Rem

update sys.wri$_adv_def_parameters
set   description = 'QSM-03004'
where advisor_id = 0
      and name = 'DAYS_TO_EXPIRE';

update sys.wri$_adv_def_parameters
set   description = 'SMG-00905'
where advisor_id = 0
      and name = 'END_SNAPSHOT';

update sys.wri$_adv_def_parameters
set   description = 'QSM-03011'
where advisor_id = 0
      and name = 'END_TIME';

```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00906'
where  advisor_id = 0
      and name = 'INSTANCE';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03001'
where  advisor_id = 0
      and name = 'JOURNALING';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03015'
where  advisor_id = 0
      and name = 'MODE';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00907'
where  advisor_id = 0
      and name = 'START_SNAPSHOT';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03022'
where  advisor_id = 0
      and name = 'START_TIME';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00900'
where  advisor_id = 0
      and name = 'TARGET_OBJECTS';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03038'
where  advisor_id = 0
      and name = 'TIME_LIMIT';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00901'
where  advisor_id = 1
      and name = 'ANALYSIS_TYPE';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00902'
where  advisor_id = 1
      and name = 'DBIO_EXPECTED';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00903'
where  advisor_id = 1
      and name = 'DB_ELAPSED_TIME';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00904'
where  advisor_id = 1
      and name = 'DB_ID';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00900'
```

```

where advisor_id = 1
and name = 'HISTORY_TABLE' ;

update sys.wri$_adv_def_parameters
set description = 'SMG-00900'
where advisor_id = 1
and name = 'SCOPE_TYPE' ;

update sys.wri$_adv_def_parameters
set description = 'SMG-00900'
where advisor_id = 1
and name = 'SCOPE_VALUE' ;

update sys.wri$_adv_def_parameters
set description = 'QSM-03002'
where advisor_id = 2
and name = 'ACTION_LIST' ;

update sys.wri$_adv_def_parameters
set description = 'QSM-03051'
where advisor_id = 2
and name = 'ADJUSTED_SCALEUP_GREEN_THRESH' ;

update sys.wri$_adv_def_parameters
set description = 'QSM-03052'
where advisor_id = 2
and name = 'ADJUSTED_SCALEUP_RED_THRESH' ;

update sys.wri$_adv_def_parameters
set description = 'QSM-03039'
where advisor_id = 2
and name = 'COMMENTED_FILTER_LIST' ;

update sys.wri$_adv_def_parameters
set description = 'QSM-03003'
where advisor_id = 2
and name = 'CREATION_COST' ;

update sys.wri$_adv_def_parameters
set description = 'QSM-03043'
where advisor_id = 2
and name = 'DEF_DATA_SOURCE' ;

update sys.wri$_adv_def_parameters
set description = 'QSM-03042'
where advisor_id = 2
and name = 'DEF_EM_TEMPLATE' ;

update sys.wri$_adv_def_parameters
set description = 'QSM-03005'
where advisor_id = 2
and name = 'DEF_INDEX_OWNER' ;

update sys.wri$_adv_def_parameters
set description = 'QSM-03006'
where advisor_id = 2
and name = 'DEF_INDEX_TABLESPACE' ;

```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03007'
where  advisor_id = 2
      and name = 'DEF_MVIEW_OWNER' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03008'
where  advisor_id = 2
      and name = 'DEF_MVIEW_TABLESPACE' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03009'
where  advisor_id = 2
      and name = 'DEF_MVLOG_TABLESPACE' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03045'
where  advisor_id = 2
      and name = 'DISABLE_FILTERS' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03010'
where  advisor_id = 2
      and name = 'DML_VOLATILITY' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00900'
where  advisor_id = 2
      and name = 'EM_DATA' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03012'
where  advisor_id = 2
      and name = 'EVALUATION_ONLY' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03013'
where  advisor_id = 2
      and name = 'EXECUTION_TYPE' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03040'
where  advisor_id = 2
      and name = 'FAST_REFRESH' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03041'
where  advisor_id = 2
      and name = 'IMPLEMENT_EXIT_ON_ERROR' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03014'
where  advisor_id = 2
      and name = 'INDEX_NAME_TEMPLATE' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03029'
where  advisor_id = 2
```

```

and name = 'INVALID_ACTION_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03031'
where advisor_id = 2
and name = 'INVALID_MODULE_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03035'
where advisor_id = 2
and name = 'INVALID_SQLSTRING_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03026'
where advisor_id = 2
and name = 'INVALID_TABLE_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03033'
where advisor_id = 2
and name = 'INVALID_USERNAME_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03016'
where advisor_id = 2
and name = 'MODULE_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03017'
where advisor_id = 2
and name = 'MVIEW_NAME_TEMPLATE';

update sys.wri$_adv_def_parameters
set description = 'QSM-03028'
where advisor_id = 2
and name = 'ORDER_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03049'
where advisor_id = 2
and name = 'OVERALL_SCALEUP_GREEN_THRESH';

update sys.wri$_adv_def_parameters
set description = 'QSM-03050'
where advisor_id = 2
and name = 'OVERALL_SCALEUP_RED_THRESH';

update sys.wri$_adv_def_parameters
set description = 'QSM-03048'
where advisor_id = 2
and name = 'RECOMMEND_MV_EXACT_TEXT_MATCH';

update sys.wri$_adv_def_parameters
set description = 'QSM-03019'
where advisor_id = 2
and name = 'REFRESH_MODE';

update sys.wri$_adv_def_parameters

```

```

set    description = 'QSM-03040'
where  advisor_id = 2
      and name = 'REFRESH_TIME';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03020'
where  advisor_id = 2
      and name = 'REPORT_DATE_FORMAT';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03040'
where  advisor_id = 2
      and name = 'REPORT_SECTIONS';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03046'
where  advisor_id = 2
      and name = 'SHOW_RETAINS';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03021'
where  advisor_id = 2
      and name = 'SQL_LIMIT';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03023'
where  advisor_id = 2
      and name = 'STORAGE_CHANGE';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03047'
where  advisor_id = 2
      and name = 'STORAGE_MODE';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03024'
where  advisor_id = 2
      and name = 'USERNAME_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03028'
where  advisor_id = 2
      and name = 'VALID_ACTION_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03030'
where  advisor_id = 2
      and name = 'VALID_MODULE_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03034'
where  advisor_id = 2
      and name = 'VALID_SQLSTRING_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03025'
where  advisor_id = 2
      and name = 'VALID_TABLE_LIST';

```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03032'
where  advisor_id = 2
      and name = 'VALID_USERNAME_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03027'
where  advisor_id = 2
      and name = 'WORKLOAD_SCOPE';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03033'
where  advisor_id = 2
      and name = '_INVALID_USERNAME_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00900'
where  advisor_id = 3
      and name = 'BEGIN_TIME';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00916'
where  advisor_id = 3
      and name = 'BEGIN_TIME_SEC';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00917'
where  advisor_id = 3
      and name = 'END_TIME_SEC';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00918'
where  advisor_id = 5
      and name = 'AUTOTASK_ID';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00919'
where  advisor_id = 5
      and name = 'AUTO_TASK';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00900'
where  advisor_id = 5
      and name = 'CONSIDER_SHRINK';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00900'
where  advisor_id = 5
      and name = 'HISTORY_LEVEL';
```

```
update sys.wri$_adv_def_parameters
set    description = 'SMG-00920'
where  advisor_id = 5
      and name = 'RECOMMEND_ALL';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03002'
```

```

where advisor_id = 6
and name = 'ACTION_LIST';

update sys.wri$adv_def_parameters
set description = 'QSM-03039'
where advisor_id = 6
and name = 'COMMENTED_FILTER_LIST';

update sys.wri$adv_def_parameters
set description = 'QSM-03043'
where advisor_id = 6
and name = 'DEF_DATA_SOURCE';

update sys.wri$adv_def_parameters
set description = 'QSM-03042'
where advisor_id = 6
and name = 'DEF_EM_TEMPLATE';

update sys.wri$adv_def_parameters
set description = 'QSM-03045'
where advisor_id = 6
and name = 'DISABLE_FILTERS';

update sys.wri$adv_def_parameters
set description = 'QSM-03029'
where advisor_id = 6
and name = 'INVALID_ACTION_LIST';

update sys.wri$adv_def_parameters
set description = 'QSM-03031'
where advisor_id = 6
and name = 'INVALID_MODULE_LIST';

update sys.wri$adv_def_parameters
set description = 'QSM-03035'
where advisor_id = 6
and name = 'INVALID_SQLSTRING_LIST';

update sys.wri$adv_def_parameters
set description = 'QSM-03026'
where advisor_id = 6
and name = 'INVALID_TABLE_LIST';

update sys.wri$adv_def_parameters
set description = 'QSM-03033'
where advisor_id = 6
and name = 'INVALID_USERNAME_LIST';

update sys.wri$adv_def_parameters
set description = 'QSM-03018'
where advisor_id = 6
and name = 'MODULE_LIST';

update sys.wri$adv_def_parameters
set description = 'QSM-03037'
where advisor_id = 6
and name = 'ORDER_LIST';

```



```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03020'
where  advisor_id = 6
      and name = 'REPORT_DATE_FORMAT';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03040'
where  advisor_id = 6
      and name = 'REPORT_SECTIONS';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03036'
where  advisor_id = 6
      and name = 'SQL_LIMIT';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03024'
where  advisor_id = 6
      and name = 'USERNAME_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03028'
where  advisor_id = 6
      and name = 'VALID_ACTION_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03030'
where  advisor_id = 6
      and name = 'VALID_MODULE_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03034'
where  advisor_id = 6
      and name = 'VALID_SQLSTRING_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03025'
where  advisor_id = 6
      and name = 'VALID_TABLE_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03032'
where  advisor_id = 6
      and name = 'VALID_USERNAME_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03033'
where  advisor_id = 6
      and name = '_INVALID_USERNAME_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03002'
where  advisor_id = 7
      and name = 'ACTION_LIST';
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03039'
where  advisor_id = 7
```

```
and name = 'COMMENTED_FILTER_LIST';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03003'  
where  advisor_id = 7  
and    name = 'CREATION_COST';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03043'  
where  advisor_id = 7  
and    name = 'DEF_DATA_SOURCE';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03042'  
where  advisor_id = 7  
and    name = 'DEF_EM_TEMPLATE';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03005'  
where  advisor_id = 7  
and    name = 'DEF_INDEX_OWNER';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03006'  
where  advisor_id = 7  
and    name = 'DEF_INDEX_TABLESPACE';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03007'  
where  advisor_id = 7  
and    name = 'DEF_MVIEW_OWNER';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03008'  
where  advisor_id = 7  
and    name = 'DEF_MVIEW_TABLESPACE';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03009'  
where  advisor_id = 7  
and    name = 'DEF_MVLOG_TABLESPACE';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03045'  
where  advisor_id = 7  
and    name = 'DISABLE_FILTERS';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'QSM-03010'  
where  advisor_id = 7  
and    name = 'DML_VOLATILITY';
```

```
update sys.wri$_adv_def_parameters  
set    description = 'SMG-00900'  
where  advisor_id = 7  
and    name = 'EM_DATA';
```

```
update sys.wri$_adv_def_parameters
```

```

set    description = 'QSM-03012'
where  advisor_id = 7
      and  name = 'EVALUATION_ONLY';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03013'
where  advisor_id = 7
      and  name = 'EXECUTION_TYPE';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03040'
where  advisor_id = 7
      and  name = 'FAST_REFRESH';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03041'
where  advisor_id = 7
      and  name = 'IMPLEMENT_EXIT_ON_ERROR';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03014'
where  advisor_id = 7
      and  name = 'INDEX_NAME_TEMPLATE';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03029'
where  advisor_id = 7
      and  name = 'INVALID_ACTION_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03031'
where  advisor_id = 7
      and  name = 'INVALID_MODULE_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03034'
where  advisor_id = 7
      and  name = 'INVALID_SQLSTRING_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03026'
where  advisor_id = 7
      and  name = 'INVALID_TABLE_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03033'
where  advisor_id = 7
      and  name = 'INVALID_USERNAME_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03016'
where  advisor_id = 7
      and  name = 'MODULE_LIST';

update sys.wri$_adv_def_parameters
set    description = 'QSM-03017'
where  advisor_id = 7
      and  name = 'MVIEW_NAME_TEMPLATE';

```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03028'
where  advisor_id = 7
      and name = 'ORDER_LIST' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03048'
where  advisor_id = 7
      and name = 'RECOMMEND_MV_EXACT_TEXT_MATCH' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03019'
where  advisor_id = 7
      and name = 'REFRESH_MODE' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03040'
where  advisor_id = 7
      and name = 'REFRESH_TIME' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03020'
where  advisor_id = 7
      and name = 'REPORT_DATE_FORMAT' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03040'
where  advisor_id = 7
      and name = 'REPORT_SECTIONS' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03046'
where  advisor_id = 7
      and name = 'SHOW_RETAINS' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03021'
where  advisor_id = 7
      and name = 'SQL_LIMIT' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03023'
where  advisor_id = 7
      and name = 'STORAGE_CHANGE' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03047'
where  advisor_id = 7
      and name = 'STORAGE_MODE' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03024'
where  advisor_id = 7
      and name = 'USERNAME_LIST' ;
```

```
update sys.wri$_adv_def_parameters
set    description = 'QSM-03028'
```

```

where advisor_id = 7
and name = 'VALID_ACTION_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03030'
where advisor_id = 7
and name = 'VALID_MODULE_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03034'
where advisor_id = 7
and name = 'VALID_SQLSTRING_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03025'
where advisor_id = 7
and name = 'VALID_TABLE_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03032'
where advisor_id = 7
and name = 'VALID_USERNAME_LIST';

update sys.wri$_adv_def_parameters
set description = 'QSM-03027'
where advisor_id = 7
and name = 'WORKLOAD_SCOPE';

update sys.wri$_adv_def_parameters
set description = 'QSM-03033'
where advisor_id = 7
and name = '_INVALID_USERNAME_LIST';

```

commit;

Rem

Rem add descriptions to current task parameters

Rem

```

update sys.wri$_adv_parameters p
set description =
(select max(dp.description)
from sys.wri$_adv_tasks t, sys.wri$_adv_def_parameters dp
where t.id = p.task_id
and dp.advisor_id = t.advisor_id
and dp.name = p.name)
where description is null;

update sys.wri$_adv_parameters p
set description =
(select max(dp.description)
from sys.wri$_adv_def_parameters dp
where dp.advisor_id = 0
and dp.name = p.name)
where description is null;

```

Rem

```

Rem Changes in advisor definitions: make sqltune task resumable
Rem
update wri$_adv_definitions
    set property = 7
    where id = 4;

commit;

Rem=====
Rem End Advisor Framework upgrade items
Rem=====

Rem=====
Rem Begin SQL Tuning Set upgrade items
Rem=====

-- Change to the new SQL tuning set schema. This is done in two steps:
-- 1. Change parsing_schema_id to parsing_schema_name in
--    wri$_sqlset_statements
-- 2. Split the wri$_sqlset_statements into its component tables.
--    Also create the new version of the binds table

-- 1. Add parsing schema name and populate it
ALTER TABLE wri$_sqlset_statements_10gr1 ADD (parsing_schema_name VARCHAR2(30));

BEGIN
    EXECUTE IMMEDIATE
        'UPDATE wri$_sqlset_statements_10gr1 st SET parsing_schema_name = ' ||
        '(select username from dba_users u where u.user_id = st.parsing_schema_id)';
EXCEPTION
    WHEN OTHERS THEN
        -- Invalid table, column error (in the case of a re-run)
        IF (SQLCODE = -942 OR SQLCODE = -904) THEN
            NULL;
        ELSE
            RAISE;
        END IF;
END;
/

-- 2. Split the statements table into statistics, statements, and mask

-- We do the actual split in a PL/SQL block to avoid doing it twice
-- We have to perform most SQL with EXECUTE IMMEDIATE because otherwise
-- it would not parse after the upgrade
DECLARE
    already_r2    NUMBER;

    -- For iterating over the statements table, to compute the force match
    -- signature
    CURSOR stmt_cur IS
        SELECT s.sql_id, st.sql_text
        FROM   wri$_sqlset_statements s, wrh$_sqltext st
        WHERE  s.sql_id = st.sql_id;

    sql_id          VARCHAR2(13);
    sql_text        CLOB;
    force_matching_signature NUMBER;

```

BEGIN

already\_r2 := 0;

-- Migration already done? Check for a statements\_R1 table

```
select DECODE(count(*),
               0, 1,
               0)
```

into already\_r2

from dba\_tables

where owner = 'SYS' and table\_name = 'WRI\$\_SQLSET\_STATEMENTS\_10GR1';

IF (already\_r2 = 0) THEN

-- Split the existing statements table into statements,stats,mask,plans

-- Enter a 0 for the signature. We compute it in the next step

EXECUTE IMMEDIATE

```
'INSERT /*+ APPEND */ INTO wri$_sqlset_statements ' ||
'SELECT wri$_sqlset_stmt_id_seq.NEXTVAL id, s.sqlset_id, s.sql_id, 0, ' ||
'      s.parsing_schema_name, s.module, s.action, s.command_type ' ||
'FROM   wri$_sqlset_statements_10gr1 s, wrh$_sqltext st ' ||
'WHERE  s.sql_id = st.sql_id';
```

commit;

-- Compute the force matching signature (we must loop because of the

-- non-sql boolean type)

OPEN stmt\_cur;

LOOP

FETCH stmt\_cur INTO sql\_id, sql\_text;

EXIT WHEN stmt\_cur%NOTFOUND;

force\_matching\_signature :=

dbms\_sqltune\_internal.i\_sqltext\_to\_signature(sql\_text, 1);

EXECUTE IMMEDIATE

```
'UPDATE wri$_sqlset_statements ' ||
'  SET force_matching_signature = :1 WHERE sql_id = :2 ' ||
USING force_matching_signature, sql_id;
```

END LOOP;

CLOSE stmt\_cur;

-- Insert zero values for delta columns and null values for load time

-- columns

EXECUTE IMMEDIATE

```
'INSERT /*+ APPEND */ INTO wri$_sqlset_statistics ' ||
'SELECT st.id stmt_id, 0 plan_hash_value, elapsed_time, 0, ' ||
'      cpu_time, 0, buffer_gets, 0, disk_reads, 0, ' ||
'      0, 0, rows_processed, 0, ' ||
'      fetches, 0, executions, 0, end_of_fetch_count, optimizer_cost, ' ||
'      NULL, NULL, stat_period, active_stat_period ' ||
'FROM   wri$_sqlset_statements_10gr1 st_tmp, ' ||
'      wri$_sqlset_statements st ' ||
'WHERE  st.sqlset_id = st_tmp.sqlset_id AND st.sql_id = st_tmp.sql_id';
```

EXECUTE IMMEDIATE

```
'INSERT /*+ APPEND */ INTO wri$_sqlset_mask ' ||
'SELECT st.id stmt_id, 0 plan_hash_value, priority, NULL other ' ||
'FROM   wri$_sqlset_statements_10gr1 st_tmp, ' ||
```

```

        '      wri$_sqlset_statements st '
    'WHERE st.sqlset_id = st_tmp.sqlset_id AND st.sql_id = st_tmp.sql_id';

-- Add a planhash of zero to every statement in the _plans table
EXECUTE IMMEDIATE
    'INSERT /*+ APPEND */ INTO wri$_sqlset_plans '
    'SELECT st.id stmt_id, 0 plan_hash_value, st_tmp.parsing_schema_name, '
    '      NULL bind_data, st_tmp.optimizer_env, '
    '      NULL plan_timestamp, nvl2(b_tmp.position, ''N'', NULL) capture '
    'FROM wri$_sqlset_statements_10gr1 st_tmp, '
    '      wri$_sqlset_statements st, '
    '      wri$_sqlset_binds_10gr1 b_tmp '
    'WHERE st.sqlset_id = st_tmp.sqlset_id AND '
    '      st.sql_id = st_tmp.sql_id AND '
    '      st_tmp.sqlset_id = b_tmp.sqlset_id(+) AND '
    '      st_tmp.sql_id = b_tmp.sql_id(+) AND b_tmp.position(+) = 1';

-- Copy from the old binds table into the new one
EXECUTE IMMEDIATE
    'INSERT /*+ APPEND */ INTO wri$_sqlset_binds '
    'SELECT st.id stmt_id, 0 plan_hash_value, position, value '
    'FROM wri$_sqlset_statements st, '
    '      wri$_sqlset_binds_10gr1 binds '
    'WHERE binds.sqlset_id = st.sqlset_id AND '
    '      binds.sql_id = st.sql_id';

commit;

-- Drop the old statements table and the old binds table
EXECUTE IMMEDIATE 'DROP TABLE wri$_sqlset_statements_10gr1';
EXECUTE IMMEDIATE 'DROP TABLE wri$_sqlset_binds_10gr1';

END IF;
END;
/

Rem=====
Rem End SQL Tuning Set upgrade items
Rem=====

Rem=====
Rem Begin Server Generated Alerts upgrade items
Rem=====

-- Alter alert queue table and alert queue to add comment
BEGIN
    dbms_aqadm.alter_queue_table(
        queue_table => 'SYS.ALERT_QT',
        comment => 'Server Generated Alert Queue Table');
    dbms_aqadm.alter_queue(
        queue_name => 'SYS.ALERT_QUE',
        comment => 'Server Generated Alert Queue');

    commit;
EXCEPTION
    WHEN OTHERS THEN
        rollback;
END;
```



/

```
Rem=====
Rem End Server Generated Alerts upgrade items
Rem=====
```

```
Rem
Rem Begin CDC changes here
Rem
Rem hotlog&sync has type changes, autolog are all user defined to set bit
BEGIN
```

```
    UPDATE cdc_change_sources$
       SET source_type = 4
       WHERE source_name = 'HOTLOG_SOURCE' ;
```

```
    UPDATE cdc_change_sources$
       SET source_type = 8
       WHERE source_name = 'SYNC_SOURCE' ;
```

```
    UPDATE cdc_change_sources$
       SET source_type = source_type + 128
       WHERE source_type = 2;
```

```
    COMMIT;
END;
```

/

```
Rem
Rem End CDC changes
Rem
```

```
Rem=====
Rem Begin Logical Standby upgrade items
Rem=====
```

```
Rem
Rem Complete Logical Standby upgrade with migration of metadata to SYSAUX
Rem
DECLARE
```

```
    tablespacename VARCHAR2(32) := null;
```

```
BEGIN
```

```
    BEGIN
```

```
        select s.name into tablespacename
        from obj$ o, ts$ s, user$ u, tab$ t
        where s.ts# = t.ts# and o.obj# = t.obj# and
              o.owner# = u.user# and u.name = 'SYSTEM' and
              o.name = 'LOGSTDBY$EVENTS' and rownum = 1;
```

```
    EXCEPTION
```

```
        WHEN OTHERS THEN
```

```
            tablespacename := 'SYSAUX' ;
```

```
    END;
```

```
    IF 'SYSTEM' = tablespacename OR 'SYSAUX' = tablespacename THEN
```

```
        dbms_logstdby.set_tablespace('SYSAUX');
```

```
    END IF;
```

```
END;
```

/

```

Rem
Rem Always move logstdby$parameters to system tablespace
Rem
BEGIN
    execute immediate 'ALTER TABLE SYSTEM.LOGSTDBY$PARAMETERS ' ||
        'MOVE TABLESPACE SYSTEM';
END;
/

Rem
Rem Alter LOGSTDBY$EVENTS table
Rem
BEGIN
    execute immediate 'ALTER TABLE SYSTEM.LOGSTDBY$EVENTS ' ||
        'MODIFY LOB (full_event) (PCTVERSION 0)';
END;
/

Rem
Rem Turn OFF LOG_AUTO_DELETE for older Logical Standby sessions
Rem
begin
    delete from system.logstdby$parameters where name = 'LOG_AUTO_DELETE';

    insert into system.logstdby$parameters (name, value)
        (select 'LOG_AUTO_DELETE', 'FALSE' from dual
         where
             (select count(*) from system.logmnr_session$
              where client#=2) > 0);

    commit;
end;
/

Rem=====
Rem End Logical Standby upgrade items
Rem=====

Rem=====
Rem Begin moving lob to enable storage in row for kottd$, kotad$, kottb$
Rem and kotmd$ tables in the db that is upgraded from 8.0
Rem=====

DECLARE
    lob_property NUMBER;
    index_name    VARCHAR2(30);

BEGIN
    SELECT bitand(1.property, 2) INTO lob_property
    FROM obj$ o, lob$ l WHERE o.obj#=l.obj# AND o.name='KOTTD$';

    IF (lob_property != 2) THEN
    BEGIN
        execute immediate 'ALTER TABLE KOTTD$ MOVE
            LOB(sys_nc_rowinfo$) STORE AS (ENABLE STORAGE IN ROW)';

        SELECT o.name INTO index_name FROM obj$ o, ind$ i, obj$ b WHERE
            b.obj#=i.bo# AND o.obj#=i.obj# AND i.type#=1 AND b.name='KOTTD$';
    
```

```

        execute immediate 'ALTER INDEX ' || index_name || ' REBUILD';
END;
END IF;

SELECT bitand(1.property, 2) INTO lob_property
FROM obj$ o, lob$ l WHERE o.obj#=l.obj# AND o.name='KOTAD$';

IF (lob_property != 2) THEN
BEGIN
    execute immediate 'ALTER TABLE KOTAD$ MOVE
        LOB(sys_nc_rowinfo$) STORE AS (ENABLE STORAGE IN ROW)';

    SELECT o.name INTO index_name FROM obj$ o, ind$ i, obj$ b WHERE
        b.obj#=i.bo# AND o.obj#=i.obj# AND i.type#=1 AND b.name='KOTAD$';

    execute immediate 'ALTER INDEX ' || index_name || ' REBUILD';
END;
END IF;

SELECT bitand(1.property, 2) INTO lob_property
FROM obj$ o, lob$ l WHERE o.obj#=l.obj# AND o.name='KOTTB$';

IF (lob_property != 2) THEN
BEGIN
    execute immediate 'ALTER TABLE KOTTB$ MOVE
        LOB(sys_nc_rowinfo$) STORE AS (ENABLE STORAGE IN ROW)';

    SELECT o.name INTO index_name FROM obj$ o, ind$ i, obj$ b WHERE
        b.obj#=i.bo# AND o.obj#=i.obj# AND i.type#=1 AND b.name='KOTTB$';
    execute immediate 'ALTER INDEX ' || index_name || ' REBUILD';
END;
END IF;

SELECT bitand(1.property, 2) INTO lob_property
FROM obj$ o, lob$ l WHERE o.obj#=l.obj# AND o.name='KOTMD$';

IF (lob_property != 2) THEN
BEGIN
    execute immediate 'ALTER TABLE KOTMD$ MOVE
        LOB(sys_nc_rowinfo$) STORE AS (ENABLE STORAGE IN ROW)';

    SELECT o.name INTO index_name FROM obj$ o, ind$ i, obj$ b WHERE
        b.obj#=i.bo# AND o.obj#=i.obj# AND i.type#=1 AND b.name='KOTMD$';

    execute immediate 'ALTER INDEX ' || index_name || ' REBUILD';
END;
END IF;

END;
/

alter system flush shared_pool;

Rem=====
Rem End moving lob to enable storage in row for kottd$, kotad$, kottb$
Rem and kotmd$ tables in the db that is upgraded from 8.0
Rem=====

```

```
Rem=====
Rem Begin Advanced Queuing upgrade items
Rem=====
```

```
DECLARE
```

```
CURSOR buf_cur IS
SELECT qt.schema, qt.name, qt.flags
  FROM system.aq$queue_tables qt
 WHERE EXISTS (SELECT q.name
                FROM system.aq$queues q
                WHERE q.table_objno = qt.objno
                AND (bitand(q.properties, 512) = 512));
alt_stmt1 VARCHAR2(300);
alt_stmt2 VARCHAR2(300);
```

```
tab_w_errors exception;
pragma          exception_init(tab_w_errors, -4063);
```

```
BEGIN
```

```
FOR buf_rec IN buf_cur LOOP
  BEGIN
    alt_stmt1 := 'ALTER TABLE ' ||
                buf_rec.schema || '.' || 'AQ$_' || buf_rec.name || '_P' ||
                ' DROP PRIMARY KEY';
    EXECUTE IMMEDIATE alt_stmt1;

    alt_stmt2 := 'ALTER TABLE ' ||
                buf_rec.schema || '.' || 'AQ$_' || buf_rec.name || '_P' ||
                ' ADD PRIMARY KEY (q_name, msgid)';
    EXECUTE IMMEDIATE alt_stmt2;
  EXCEPTION
    WHEN tab_w_errors THEN
      dbms_system.ksdwrt(dbms_system.alert_file,
        'upgrade of queue table:' || buf_rec.schema || '.' || buf_rec.name ||
        ' to 10.2 failed when altering constraints. ' ||
        ' Queue table had errors, recreate the queue table');
  END;
```

```
-- No automatic stats collection for AQ spill tables
```

```
DBMS_STATS.UNLOCK_TABLE_STATS(buf_rec.schema, 'AQ$_' || buf_rec.name || '_P');
DBMS_STATS.DELETE_TABLE_STATS(buf_rec.schema, 'AQ$_' || buf_rec.name || '_P');
DBMS_STATS.LOCK_TABLE_STATS(buf_rec.schema, 'AQ$_' || buf_rec.name || '_P');
```

```
END LOOP;
END;
/
```

```
DECLARE
```

```
CURSOR qt_cur IS
SELECT qt.schema, qt.name, qt.flags
  FROM system.aq$queue_tables qt;
qt_w_errors exception;
pragma          exception_init(qt_w_errors, -24203);
tab_w_errors exception;
pragma          exception_init(tab_w_errors, -4063);
```

BEGIN

FOR qt\_rec IN qt\_cur LOOP

BEGIN

-- for multiconsumer newstyle, recreate, for scq newstyle, first create

IF (bitand(qt\_rec.flags, 1) = 1 and bitand(qt\_rec.flags, 8) = 8) THEN

dbms\_aqadm\_sys.drop\_buffer\_view(qt\_rec.schema, qt\_rec.name);

END IF;

-- ignore if view was already created

IF (bitand(qt\_rec.flags, 8) = 8) THEN

dbms\_aqadm\_sys.create\_buffer\_view(qt\_rec.schema, qt\_rec.name, TRUE);

END IF;

IF (bitand(qt\_rec.flags, 1) = 1 and bitand(qt\_rec.flags, 8) = 8) THEN

sys.dbms\_prvtaqim.create\_base\_view(  
qt\_rec.schema, qt\_rec.name, qt\_rec.flags);

ELSE

sys.dbms\_aqadm\_sys.create\_base\_view(  
qt\_rec.schema, qt\_rec.name, qt\_rec.flags);

END IF;

EXCEPTION

WHEN qt\_w\_errors OR tab\_w\_errors THEN

dbms\_system.ksdwrt(dbms\_system.alert\_file,  
'upgrade of queue table:' || qt\_rec.schema || '.' || qt\_rec.name ||  
' to 10.2 failed when recreating views.' ||  
' Queue table had errors, recreate the queue table');

END;

-- No automatic stats collection for AQ tables

DBMS\_STATS.UNLOCK\_TABLE\_STATS(qt\_rec.schema, qt\_rec.name);

DBMS\_STATS.DELETE\_TABLE\_STATS(qt\_rec.schema, qt\_rec.name);

DBMS\_STATS.LOCK\_TABLE\_STATS(qt\_rec.schema, qt\_rec.name);

END LOOP;

END;

/

Rem=====

Rem End Advanced Queuing upgrade items

Rem=====

Rem =====

Rem Begin Changes link\$

Rem =====

Rem =====

Rem Upgrade should just re-execute the create dblink clause to get  
Rem encoded version.

Rem =====

Execute dbms\_dblink.upgrade;

Rem =====

Rem End of link\$ Changes

Rem =====

```

Rem =====
Rem =====
Rem Upgrade sets failed_login_attempts = 10
Rem          if it is UNLIMITED for DEFAULT profile
Rem =====

DECLARE
    prec DBA_PROFILES%ROWTYPE;
BEGIN
    SELECT * INTO prec FROM DBA_PROFILES
    WHERE profile = 'DEFAULT' AND resource_name = 'FAILED_LOGIN_ATTEMPTS';

    IF prec.LIMIT = 'UNLIMITED' THEN
        EXECUTE IMMEDIATE
            'ALTER PROFILE default LIMIT failed_login_attempts 10';
    END IF;
END;
/

Rem =====
Rem End of DEFAULT profile changes
Rem =====

Rem =====
Rem Begin (bug 3989938): fix partitioned tables that had retention set
Rem =====

DECLARE
    c          varchar2(200);
    CURSOR c_ret IS
        select u.name OWNER, o.name TABLE_NAME, c.name COLUMN_NAME
        from sys.obj$ o, sys.col$ c, sys.partlob$ l, sys.user$ u
        where o.owner# = u.user# and
              o.obj# = c.obj# and
              c.obj# = l.tabobj# and
              c.intcol# = l.intcol# and
              bitand(c.property, 32768) != 32768 and
              bitand(l.defflags, 32) = 32;
BEGIN
    FOR r_ret in c_ret LOOP
        BEGIN
            c := 'ALTER TABLE "' || r_ret.owner || '"."' || r_ret.table_name || '"';
            c := c || ' MODIFY LOB("' || r_ret.column_name || '" ) (retention)';
            EXECUTE IMMEDIATE c;
        END;
    END LOOP;
END;
/

Rem =====
Rem End (bug 3989938)
Rem =====

Rem
Rem END STAGE 1: upgrade from 10.1.0 to 10.2

```

Rem =====

Rem =====

Rem BEGIN STAGE 2: invoke script for subsequent release

Rem =====

Rem

Rem For the next release we have to add a file here called

REM @@a1002000

Rem =====

Rem END STAGE 2: invoke script for subsequent release

Rem =====

Rem \*\*\*\*\*

Rem END a1001000.sql

Rem \*\*\*\*\*