oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（28）c0902000.sql

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（28）c0902000.sql

```
#upgrade Oracle RDBMS from 9.2.0 to the new release
#Put any dictionary related changes here (ie-create, alter,
Rem      update,...).  DO NOT put PL/SQL modules in this script.
Rem      If you must upgrade using PL/SQL, put the module in a0902000.sql
Rem      as catalog.sql and catproc.sql will be run before a0902000.sql
Rem      is invoked.
Rem
Rem      This script is called from u0902000.sql and c0900010.sql
Rem
Rem      This script performs the upgrade in the following stages:
Rem        STAGE 1: upgrade from 9.2.0 to 10.1
Rem        STAGE 2: upgrade from 10.1 to the next release
#

Rem
Rem $Header: c0902000.sql 02-may-2005.13:37:08 mtakahar Exp $
Rem
Rem c0902000.sql
Rem
Rem Copyright (c) 1999, 2005, Oracle. All rights reserved.
Rem
Rem    NAME
Rem      c0902000.sql - upgrade Oracle RDBMS from 9.2.0 to the new release
Rem
Rem    DESCRIPTION
Rem      Put any dictionary related changes here (ie-create, alter,
Rem      update,...).  DO NOT put PL/SQL modules in this script.
Rem      If you must upgrade using PL/SQL, put the module in a0902000.sql
Rem      as catalog.sql and catproc.sql will be run before a0902000.sql
Rem      is invoked.
Rem
Rem      This script is called from u0902000.sql and c0900010.sql
Rem
Rem      This script performs the upgrade in the following stages:
Rem        STAGE 1: upgrade from 9.2.0 to 10.1
Rem        STAGE 2: upgrade from 10.1 to the next 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    mtakahar    04/26/05 - lrg1847073 don't mark dubious stats if reupgrade
Rem    rburns      05/06/05 - limit undo
Rem    amanikut    04/27/05 - reupgrade test
Rem    rburns      09/02/04 - remove serveroutput
Rem    ycao        08/23/04 - bug 3841411: move LE lob$ logic from catproc
Rem    ciyer       07/24/04 - invalidate views to pick up new dependeny
Rem    arithikr    08/11/04 - 3798917 - delete DEFAULT_TBS_TYPE from props$
Rem    rburns      07/15/04 - remove dbms_output compiles
Rem    mtakahar    04/20/04 - #(3272499) flag potentially dubious stats
Rem    jciminsk    03/04/04 - move grid to c1001000.sql
Rem    jciminsk    02/06/04 - merge from RDBMS_MAIN_SOLARIS_040203
```

```
Rem    jciminsk    12/12/03 - merge from RDBMS_MAIN_SOLARIS_031209
Rem    ksurlake    08/26/03 - modify primary key for aq$_propagation_status
Rem    lchidamb    08/29/03 - add retry_count, retry_time
Rem    lchidamb    08/14/03 - add director quiesce operations table
Rem    rvenkate    08/13/03 - increase network_name size
Rem    lchidamb    08/13/03 - add node/service policy tables
Rem    ksurlake    08/18/03 - Change primary key for aq$_propagation_status
Rem    lchidamb    07/29/03 - add director escalation table
Rem    elu         07/25/03 - modify dir$database_attributes
Rem    jstamos     07/29/03 - add director columns
Rem    rvenkate    07/16/03 - add services to aq$_queues
Rem    elu         07/16/03 - add db priority for grid
Rem    ckantarj    07/03/03 - add TAF characteristics to service$
Rem    jstamos     06/30/03 - add director state
Rem    rburns      01/07/04 - add calls to 10.1 scripts
Rem    nireland    01/07/04 - Fix ts# for indices on temp tables. #3238525
Rem    rvissapr    12/11/03 - bug 3275411 - seq$ will have 32chars in audit$
Rem    xan         12/15/03 - bug fix: 3320404
Rem    rburns      12/09/03 - bug 3306397 - fix spare6
Rem    qyu         12/04/03 - #3048174: fix charset in col#
Rem    gmulagun    12/03/03 - #3294084: Upgrade user$.audit$ column
Rem    ksurlake    11/05/03 - Bug 2867252: Upgrade for reg_info and reg$
Rem    ksurlake    11/03/03 - Bug 2867252: Upgrade for aq$_srvntfn_msg
rem    mtakahar    10/22/03 - add storage parameters to stats storage objects
Rem    vraja       10/21/03 - rename FLASHBACK ANY TRANSACTION to SELECT ANY
Rem                           TRANSACTION
Rem    nireland    10/21/03 - Fix LRG with dba_procedures
Rem    mtyulene    09/29/03 - change cache_stats_1$ and cache_stats_0$,
Rem    zqiu        10/02/03 - new column in aw_prop$
Rem    gkulkarn    09/27/03 - bug fix: 3140873
Rem    nireland    10/08/03 - Revoke public grant on dba_procedures
Rem    dvoss       10/03/03 - clean up timeseries objects
Rem    zqiu        10/02/03 - new column in aw_prop$
Rem    gkulkarn    09/27/03 - bug fix: 3140873
Rem    jawilson    08/04/03 - Add timezone column to aq$_queue_tables
Rem    qyu         09/19/03 - #3138892: fix ts# for lob in temp tables
Rem    gmulagun    09/11/03 - change type of audit PROCESS# column
Rem    ksurlake    08/27/03 - Add ack column to aq$_replay_info
Rem    rburns      08/28/03 - cleanup
Rem    lchidamb    08/29/03 - add retry_time, retry_count
Rem    araghava    09/04/03 - (3127926): use more efficient sql to update
Rem                           partitioning tables
Rem    dsemler     08/06/03 - add system service entry
Rem    gviswana    07/03/03 - Move view invalidation to utlip.sql
Rem    lchidamb    08/14/03 - add director quiesce operations table
Rem    rvenkate    08/13/03 - increase network_name size
Rem    lchidamb    08/13/03 - add node/service policy tables
Rem    lchidamb    07/29/03 - add director escalation table
Rem    elu         07/25/03 - modify dir$database_attributes
Rem    jstamos     07/29/03 - add director columns
Rem    rvenkate    07/16/03 - add services to aq$_queues
Rem    elu         07/16/03 - add db priority for grid
Rem    ckantarj    07/03/03 - add TAF characteristics to service$
Rem    jstamos     06/30/03 - add director state
Rem    alakshmi    07/24/03 - add cascade option
Rem    clei        07/15/03 - synonym policies no longer attached to base obj
Rem    gssmith     07/14/03 - Add upgrade for Summary Advisor
Rem    mramache    06/23/03 - sql profiles
```

```
Rem    liwong      06/18/03 - dml_handlers for virtual objects
Rem    lkaplan     06/04/03 - add convert_long_to_lob_chunk
Rem    liwong      06/01/03 - Add apply$_virtual_obj_cons
Rem    mdevin      05/01/03 - Upgrade to table smon_scn_time
Rem    sbalaram    06/01/03 - add streams$_dest_objs
Rem    rburns      05/28/03 - fix lcr row_record for 9201
Rem    weiwang     05/14/03 - remove the extra IOT column for rule_set$
Rem    jnesheiw    05/29/03 - enlarge object column in logstdby$scn
Rem    raguzman    05/23/03 - set logical standby bit in TAB$
Rem    sichandr    05/23/03 - privs for multi level nested tables
Rem    lchidamb    05/09/03 - add director objects
Rem    rvissapr    05/20/03 - bug 2944537 - add exempt identity policy
Rem    lchidamb    05/09/03 - add director objects
Rem    elu         05/07/03 - add start_scn to streams$_apply_milestone
Rem    krajaman    05/01/03 - Upgrade fixes
Rem    rburns      04/25/03 - revise timestamp
Rem    jstamos     04/24/03 - add director upgrade
Rem    nshodhan    04/22/03 - bug-2897618
Rem    tbgraves    04/22/03 - merge SVRMGMT
Rem    skaluska    04/15/03 - transparent session migration
Rem    nshodhan    04/07/03 - add constructor for lcr$_row_unit
Rem    nshodhan    04/03/03 - add lcr$_row_unit
Rem    gmulagun    04/06/03 - bug 2822534: rename tran_id to xid
Rem    rburns      03/20/03 - drop 07 view
Rem    narora      03/19/03 - bug 2842797: default value of fetchlwm_scn
Rem    zqiu        03/10/03 - more columns for aw_obj$
Rem    vraja       02/10/03 - add FLASHBACK ANY TRANSACTION priv
Rem    gviswana    02/03/03 - Invalidate views to pick up new dependency model
Rem    srtata      02/07/03 - change DDL and DML stmts on aud
Rem    mxiao       01/30/03 - change to MATERIALIZED VIEW in AUDIT_ACTIONS
Rem    narora      01/13/03 - add fetchlwm_scn to apply_milestone
Rem    alakshmi    01/20/03 - streams$_capture_process.version varchar2(30=>64)
Rem    jwwarner    01/27/03 - upgrade xmlgenformattype
Rem    nbhatt      01/28/03 - lrg 1295018
Rem    clei        01/15/03 - change rls_sc$
Rem    pabingha    01/17/03 - CDC subscription description length
Rem    rburns      01/14/03 - fix registry version
Rem    weiwang     01/15/03 - invalidate dependents of rule set
Rem    htran       01/14/03 - i_streams_message_consumers only on streams_name
Rem    svivian     01/09/03 - back out Plan Stability changes
Rem    tkeefe      01/08/03 - bug 2734166: Eliminate multiple inserts
Rem                           into proxy_info$
Rem    twtong      01/10/03 - fix bug-2677089
Rem    rburns      12/06/02 - add namespace to registry table
Rem    lbarton     12/31/02 - modify metascript, metascriptfilter
Rem    svivian     01/06/03 - outline temporary tables
Rem    akalra      12/04/02 - add columns to smon_scn_time
Rem    mbrey       12/19/02 - adding columns to cdc_change_tables$
Rem    raguzman    12/19/02 - add dbid column to fga_log$ and aud$
Rem    nfolkert    12/24/02 - remove invalidation of summaries
Rem    tbgraves    12/10/02 - initial tablespace sizes for SYSTEM/SYSAUX
Rem                           remove CATALOG registry timestamp
Rem    zqiu        12/03/02 - add OLAP Service system table indice
Rem    mmorsi      12/11/02 - Fix for Bug 2707312
Rem    gclaborn    12/12/02 - Add column parse_attr to metaxslparam$
Rem    rvissapr    12/16/02 - bug 2594538
Rem    jwwarner    12/11/02 - drop old xmlconcat fcn
Rem    sslim       11/22/02 - lrg 1112873: logical standby support
```

```
Rem    zqiu        11/20/02 - modify OLAP Service system tables
Rem    htran       11/15/02 - expand some Streams columns
Rem    akalra      11/26/02 - remove indexes on smon_scn_time
Rem    gmulagun    11/20/02 - add lsqlbind clob column
Rem    alakshmi    11/08/02 - add streams$_capture_process.version
Rem    alakshmi    11/04/02 - MVDD de-coupling during upgrade
Rem    sagrawal    11/12/02 - lrg fix
Rem    pabingha    11/12/02 - CDC generate sub. name
Rem    sagrawal    10/11/02 - PL/SQL warnings
Rem    jgalanes    11/06/02 - Add expimp_tts_ct$ table for 2383871
Rem    rburns      11/05/02 - move ncomp_ddl creation
Rem    mvemulap    10/14/02 - add ncomp_dll
Rem    akalra      11/04/02 - add indexes on smon_scn_time
Rem    mtyulene    10/21/02 - add tab_stats$, ind_stats$
Rem    mmorsi      10/11/02 - adding type mgr upgrade for binary float/double
Rem    rramkiss    10/30/02 - Add new scheduler privileges
Rem    liwong      10/23/02 - Add status_change_time
Rem    nmanappa    10/21/02 - populating padding bytes of audit$ column
Rem    dsemler     10/15/02 - service object
Rem    apadmana    10/18/02 - Add table streams$_message_rules
Rem    apadmana    10/14/02 - Sysaux: Streams
Rem    mmorsi      10/11/02 - adding type mgr upgrade for binary float/double
Rem    masubram    10/06/02 - add new online redefinition table
Rem    schakkap    10/03/02 - tab_stats$, fixed_obj$
Rem    zqiu        10/09/02 - more OLAP Service system tables
Rem    asundqui    10/07/02 - new Resource Manager parameters
Rem    dcassine    10/03/02 - add start & end date to streams$_apply_process
Rem    dcassine    10/01/02 - add start & end date to streams$_capture_process
Rem    vmarwah     10/04/02 - Undrop Tables: Record CON# in recyclebin$.
Rem    rburns      10/05/02 - fix snap alter
Rem    apadmana    09/30/02 - add table streams$_privileged_user
Rem    yhu         10/08/02 - upgrade for ODCIEnv
Rem    rburns      09/25/02 - drop OLAP_SRF_T body
Rem    tchorma     10/01/02 - Add new column to operator$
Rem    masubram    09/18/02 - new dictionary tables for online redef
Rem    mdilman     09/17/02 - insert DEFAULT_TBS_TYPE to props$
Rem    twtong      09/26/02 - add alias_txt to snap$
Rem    btao        09/20/02 - grant privileges to system for access advisor
Rem    gmulagun    09/16/02 - enhance fga_log$ and aud$ trails
Rem    kdias       09/13/02 - add advisor priv
Rem    lbarton     09/18/02 - add metapathmap$
Rem    elu         09/10/02 - add negative rule sets for streams
Rem    vmarwah     09/04/02 - Undrop Tables: Record BaseObj and Object to purge
Rem    rburns      08/30/02 - invalidate MVs on all upgrades
Rem    clei        09/03/02 - add ANALYZE ANY DICTIONARY privilege
Rem    yhu         09/17/02 - upgrade for domain index array insert
Rem    tkeefe      09/12/02 - Move proxy_data$ and proxy_role_data$ out of
Rem                           bootstrap region
Rem    mtakahar    08/26/02 - remove monitoring bit in tab$
Rem    twtong      09/10/02 - extend sum and sumdep
Rem    cluu        09/06/02 - drop obsolete mts views
Rem    wnorcott    08/28/02 - remove hard tabs from liwong txn
Rem    wnorcott    08/26/02 - fix syntax error
Rem    liwong      08/22/02 - Capture extra attributes
Rem    mxiao       08/18/02 - add columns to mlog$, cdc_change_tables$
Rem    wnorcott    08/15/02 - ADD cdc changes
Rem    clei        08/12/02 - add security relevant columns metadata
Rem    hsbedi      08/20/02 - external table upgrade
```

```
Rem    twtong      08/14/02 - extend sum$ to support rewrite equivalence
Rem    vmarwah     08/08/02 - Undrop Tables: modify RecycleBin$ schema.
Rem    esoyleme    08/02/02 - modify the correct column in ps$
Rem    rburns      08/02/02 - fix SQL statement
Rem    nshodhan    08/01/02 - streams$_capture_process changes
Rem    weiwang     07/31/02 - add rules engine upgrade script
Rem    rburns      07/30/02 - drop ODCI types
Rem    nshodhan    07/24/02 - downstream capture
Rem    dcassine    07/26/02 - add precommit handler to streams$_apply_process
Rem    rburns      07/19/02 - add timestamps
Rem    twtong      07/29/02 - add sumqb
Rem    alakshmi    07/02/02 - Handle upgrades for LCR types
Rem    vmarwah     07/19/02 - Undrop Table: Create RecycleBin$ updates
Rem    pabingha    08/07/02 - CDC change source/set changes
Rem    zqiu        07/16/02 - add OLAP Service related catalog changes
Rem    rburns      07/03/02 - move sysauth updates
Rem    rvissapr    06/21/02 - add additional index on fga
Rem    rvissapr    06/20/02 - fga dml and multi column support
Rem    mxiao       06/17/02 - rename privileges from snapshot to mat view
Rem    rburns      06/05/02 - move dependency deletes
Rem    twtong      06/19/02 - invalidate dim object after upgrade
Rem    twtong      06/17/02 - add attname to dimattr
Rem    vmarwah     05/08/02 - Undrop Tables: Creating RECYCLEBIN$ table.
Rem    dcwang      05/23/02 - move system privileges on any rules.
Rem    rburns      05/06/02 - remove v$mls_parameters
Rem    araghava    04/30/02 - upgrade partitioning metadata.
Rem    asundqui    05/03/02 - consumer group mapping interface
Rem    sbedarka    04/16/02 - #(2264056) add index on obj# to various part$
Rem    twtong      04/01/02 - fix alter suminline
Rem    dcwang      04/10/02 - add import full database and export full database
Rem    twtong      03/22/02 - add text to suminline
Rem    lbarton     03/20/02 - metadata API 10.1 dictionary changes
Rem    yuli        03/19/02 - drop v$compatibility and v$compatseg
Rem    rburns      03/17/02 - rburns_10i_updown_scripts
Rem    rburns      02/12/02 - Created

Rem========================================================================
Rem BEGIN STAGE 1: upgrade from 9.2.0 to 10.1
Rem========================================================================
alter system flush shared_pool;
alter system flush shared_pool;
alter system flush shared_pool;

Rem Patch all synonyms to have dependency on the next fellow.
Rem May have to re-work this approach.

alter session set events '22299 trace name context forever, level 1';

declare

  CURSOR alter1(objectno number) IS
  SELECT o.obj#,
    CASE
      WHEN u.name = 'PUBLIC'
        THEN 'ALTER PUBLIC SYNONYM "' || o.name || '" COMPILE'
      ELSE
       'ALTER SYNONYM' || ' "' || u.name || '"."' || o.name || '" '
        || 'COMPILE '
```

```
      END
   FROM obj$ o, user$ u WHERE o.type#=5 AND o.linkname is NULL AND
   u.user# = o.owner#  AND o.obj# > objectno order by obj#;

  ddl_statement varchar2(1000);
  my_err    number;
  objnum    number;

begin

  objnum := 0;

  OPEN alter1(objnum);

  LOOP
    BEGIN
      FETCH alter1 INTO objnum, ddl_statement;
       EXIT WHEN alter1%NOTFOUND;
    EXCEPTION
      WHEN OTHERS THEN
        my_err := SQLCODE;
        IF my_err = -1555 THEN -- snapshot too old, re-execute fetch query
          CLOSE alter1;
          OPEN  alter1(objnum);
          GOTO continue;
        ELSE
          RAISE;
        END IF;
    END;

    BEGIN
      -- Issue the Alter synonym compile statement
      EXECUTE IMMEDIATE ddl_statement;
    EXCEPTION
      WHEN OTHERS THEN
      null; -- ignore, and proceed.
    END;

<<continue>>
    null;

  END LOOP;

  CLOSE alter1;

end;
/

alter session set events '22299 trace name context off';

select count(*), status from obj$ where type#=5 group by status;

select obj#, owner#, name, linkname from obj$ o where type#=5
and status=6 and not exists
(select * from dependency$ d where d.d_obj# = o.obj#);

alter system flush shared_pool;
alter system flush shared_pool;
```

```
alter system flush shared_pool;

Rem
Rem Invalidate all views so that their dependences will be updated with
Rem the new synonym dependency model
Rem
update obj$ set status = 6
  where type# = 4
        and status not in (5,6)
        and ((subname is null) or (subname <> 'DBMS_DBUPGRADE_BABY'))
        and linkname is null;

commit;

alter system flush shared_pool;

Rem Remove entries from sys.duc$ - rebuilt for 10.1 by catalog and catproc
delete from duc$;

Rem=========================================================================
Rem Rename system privileges here
Rem=========================================================================

update SYSTEM_PRIVILEGE_MAP set name = 'CREATE MATERIALIZED VIEW'
  where privilege = -172;
update SYSTEM_PRIVILEGE_MAP set name = 'CREATE ANY MATERIALIZED VIEW'
  where privilege = -173;
update SYSTEM_PRIVILEGE_MAP set name = 'ALTER ANY MATERIALIZED VIEW'
  where privilege = -174;
update SYSTEM_PRIVILEGE_MAP set name = 'DROP ANY MATERIALIZED VIEW'
  where privilege = -175;

Rem=========================================================================
Rem Add new system privileges here
Rem=========================================================================

insert into SYSTEM_PRIVILEGE_MAP  values (-255, 'EXPORT FULL DATABASE', 1);
insert into SYSTEM_PRIVILEGE_MAP  values (-256, 'IMPORT FULL DATABASE', 1);

insert into SYSTEM_PRIVILEGE_MAP  values (-257, 'CREATE RULE', 1);
insert into SYSTEM_PRIVILEGE_MAP  values (-258, 'CREATE ANY RULE', 1);
insert into SYSTEM_PRIVILEGE_MAP  values (-259, 'ALTER ANY RULE', 1);
insert into SYSTEM_PRIVILEGE_MAP  values (-260, 'DROP ANY RULE', 1);
insert into SYSTEM_PRIVILEGE_MAP  values (-261, 'EXECUTE ANY RULE', 1);
insert into SYSTEM_PRIVILEGE_MAP  values (-262, 'ANALYZE ANY DICTIONARY', 0);
insert into SYSTEM_PRIVILEGE_MAP  values (-263, 'ADVISOR', 0);

insert into SYSTEM_PRIVILEGE_MAP  values (-264, 'CREATE JOB', 0);
insert into SYSTEM_PRIVILEGE_MAP  values (-265, 'CREATE ANY JOB', 0);
insert into SYSTEM_PRIVILEGE_MAP  values (-266, 'EXECUTE ANY PROGRAM', 0);
insert into SYSTEM_PRIVILEGE_MAP  values (-267, 'EXECUTE ANY CLASS', 0);
insert into SYSTEM_PRIVILEGE_MAP  values (-268, 'MANAGE SCHEDULER', 0);
insert into SYSTEM_PRIVILEGE_MAP  values (-269, 'SELECT ANY TRANSACTION',0);

delete from SYSTEM_PRIVILEGE_MAP where privilege in (-64, -65, -66, -67, -68);

Rem Move "Any rule" system privileges
update sysauth$ set privilege# = -257 where privilege# = -64;
```

```
update sysauth$ set privilege# = -258 where privilege# = -65;
update sysauth$ set privilege# = -259 where privilege# = -66;
update sysauth$ set privilege# = -260 where privilege# = -67;
update sysauth$ set privilege# = -261 where privilege# = -68;

grant all privileges, analyze any dictionary to dba with admin option;
grant create table to system;
grant create snapshot to system;
grant select any table to system;
grant global query rewrite to system;

Rem=========================================================================
Rem Rename audit options here
Rem=========================================================================

update STMT_AUDIT_OPTION_MAP set name = 'CREATE MATERIALIZED VIEW'
  where option# = 172;
update STMT_AUDIT_OPTION_MAP set name = 'CREATE ANY MATERIALIZED VIEW'
  where option# = 173;
update STMT_AUDIT_OPTION_MAP set name = 'ALTER ANY MATERIALIZED VIEW'
  where option# = 174;
update STMT_AUDIT_OPTION_MAP set name = 'DROP ANY MATERIALIZED VIEW'
  where option# = 175;

alter table audit_actions modify (name varchar2(28));
update audit_actions set name = 'CREATE MATERIALIZED VIEW LOG'
  where action = 71;
update audit_actions set name = 'ALTER MATERIALIZED VIEW LOG'
  where action = 72;
update audit_actions set name = 'DROP MATERIALIZED VIEW LOG'
  where action = 73;
update audit_actions set name = 'CREATE MATERIALIZED VIEW'
  where action = 74;
update audit_actions set name = 'ALTER MATERIALIZED VIEW'
  where action = 75;
update audit_actions set name = 'DROP MATERIALIZED VIEW'
  where action = 76;

Rem=========================================================================
Rem Add new audit options here
Rem=========================================================================

insert into STMT_AUDIT_OPTION_MAP values ( 255, 'EXPORT FULL DATABASE', 0);
insert into STMT_AUDIT_OPTION_MAP values ( 256, 'IMPORT FULL DATABASE', 0);
insert into STMT_AUDIT_OPTION_MAP values ( 262, 'ANALYZE ANY DICTIONARY', 0);
insert into STMT_AUDIT_OPTION_MAP values ( 263, 'ADVISOR', 0);
insert into STMT_AUDIT_OPTION_MAP values ( 264, 'CREATE JOB', 0);
insert into STMT_AUDIT_OPTION_MAP values ( 265, 'CREATE ANY JOB', 0);
insert into STMT_AUDIT_OPTION_MAP values ( 266, 'EXECUTE ANY PROGRAM', 0);
insert into STMT_AUDIT_OPTION_MAP values ( 267, 'EXECUTE ANY CLASS', 0);
insert into STMT_AUDIT_OPTION_MAP values ( 268, 'MANAGE SCHEDULER', 0);
insert into STMT_AUDIT_OPTION_MAP values ( 269, 'SELECT ANY TRANSACTION',0);

Rem=========================================================================
Rem Drop views removed from last release here
Rem remove obsolete dependencies for any fixed views in i0902000.sql
Rem=========================================================================
```

```
drop view v_$mls_parameters;
drop public synonym v$mls_parameters;
drop view gv_$mls_parameters;
drop public synonym gv$mls_parameters;
drop view V_$COMPATIBILITY;
drop public synonym V$COMPATIBILITY;
drop view GV_$COMPATIBILITY;
drop public synonym GV$COMPATIBILITY;

drop view V_$COMPATSEG;
drop public synonym V$COMPATSEG;
drop view GV_$COMPATSEG;
drop public synonym GV$COMPATSEG;

-- from catexp7.sql - has dependency on v$compatibility
drop view IMP7UEC;

drop view v_$mts;
drop public synonym v$mts;
drop view gv_$mts;
drop public synonym gv$mts;

Rem=========================================================================
Rem Drop packages removed from last release here
Rem=========================================================================



Rem=========================================================================
Rem Add changes to sql.bsq dictionary tables here
Rem=========================================================================

Rem Repair any existing invalid values in spare6
update tab$ set spare6=NULL where to_char(spare6) = '00-000-00';
commit;

rem table used to store the dropped objects which are still not purged
create table recyclebin$
(
  obj#                number not null,           /* original object number */
  owner#              number not null,                /* owner user number */
  original_name       varchar2(32),                /* Original Object Name */
  operation           number not null,            /* Operation carried out */
                                                              /* 0 -> DROP */
                                          /* 1 -> TRUNCATE (not supported) */
  type#               number not null,          /* object type (see KQD.H) */
  ts#                 number,                         /* tablespace number */
  file#               number,                /* segment header file number */
  block#              number,               /* segment header block number */
  droptime            date,                /* time when object was dropped */
  dropscn             number,           /* SCN of Tx which caused the drop */
  partition_name      varchar2(32),       /* Name of the partition dropped */
                                                        /* NULL otherwise */
  flags               number,               /* flags for undrop processing */
  related             number not null,    /* obj one level up in heirarchy */
  bo                  number not null,                      /* base object */
  purgeobj            number not null,   /* obj to purge when purging this */
  base_ts#            number,            /* Base objects Tablespace number */
  base_owner#         number,                 /* Base objects owner number */
```

```
  space                 number,       /* number of blocks used by the object */
  con#                  number,       /* con#, if index is due to constraint */
  spare1                number,
  spare2                number,
  spare3                number
)
/
create index recyclebin$_obj on recyclebin$(obj#)
/
create index recyclebin$_ts on recyclebin$(ts#)
/
create index recyclebin$_owner on recyclebin$(owner#)
/

Rem add text column to suminline$
ALTER TABLE suminline$ ADD (text long);
Rem
Rem Metadata API changes
Rem
alter table metaview$ modify (xmltag null);
alter table metaview$ modify (udt null);
alter table metaview$ modify (schema null);
alter table metaview$ modify (viewname null);
alter table metaxslparam$ add (properties number default 0 not null);
alter table metaxslparam$ add (parse_attr varchar2(2000));

Rem Add columns to mlog$, cdc_change_tables$ and set the values
Rem in a0902000.sql
ALTER TABLE mlog$ ADD (oldest_seq DATE);
ALTER TABLE cdc_change_tables$ ADD (mvl_oldest_seq NUMBER);
ALTER TABLE cdc_change_tables$ ADD (mvl_oldest_seq_time DATE);

Rem
Rem  Dictionary tables for heterogeneous object types in Metadata API
Rem
create table metascript$                  /* scripts for heterogeneous types */
( htype         varchar2(30) not null,   /* root heterogeneous objtype */
  ptype         varchar2(30) not null, /* parent heterogeneous objtype */
  seq#                number not null,                    /* sequence number */
  rseq#         number not null,        /* sequence number of reference type */
  ltype                 varchar2(30) not null,            /* leaf object name */
  properties    number not null,                    /*leaf type's properties */
                           /* 0x0001 =     1 = leaf is heterogeneous object */
  model                 varchar2(30) not null,            /* model properties */
  version      number not null      /* decimal RDBMS version: eg, 0802010000 */
)
/
create unique index i_metascript1$ on metascript$(ptype,seq#,model,version)
/
create unique index i_metascript2$ on metascript$(model,htype,seq#,version)
/
create table metascriptfilter$              /* filters for steps in a script */
( htype         varchar2(30) not null,   /* root heterogeneous objtype */
  ptype         varchar2(30) not null, /* parent heterogeneous objtype */
  seq#                number not null,                    /* sequence number */
  ltype         varchar2(30) not null,             /* leaf object name */
  filter        varchar2(30) not null,                       /*  filter name */
  pfilter        varchar2(30),                        /* parent filter name */
```

```
  vcval         varchar2(2000),                         /* filter text value */
  bval           number,                               /* filter boolean value */
  nval           number,                               /* filter numeric value */
  properties    number default 0 not null,              /* filter properties */
  model                varchar2(30) not null             /* model properties */
)
/
rem
rem (these indexes intentionally not unique)
rem
create index i_metascriptfilter1$ on metascriptfilter$(model,htype,seq#)
/
create index i_metascriptfilter2$ on metascriptfilter$(model,ptype,seq#)
/
create table metanametrans$    /* path names for heterogeneous objtype nodes */
( name                 varchar2(200) not null,                 /* path name  */
  htype                varchar2(30) not null,  /* root heterogeneous objtype */
  ptype                varchar2(30) not null,    /* immediate parent objtype */
  seq#                number not null,           /* sequence number in ptype */
  properties    number not null,                   /* path name's properties */
  /* 0x0001 =     1 = this is the fully qualified path name */
  model                varchar2(30) not null,           /* model properties  */
  descrip       varchar2(2000)             /* description of the object type */
)
/
create index i_metanametrans1$ on metanametrans$(model,htype,name)
/
create index i_metanametrans2$ on metanametrans$(model,ptype,seq#)
/
create table metapathmap$  /* het objtypes containing objs named by pathname */
( name         varchar2(200) not null,                          /* path name */
  htype        varchar2(30) not null,               /* heterogeneous objtype */
  model        varchar2(30) not null,                           /* model name */
  version      number not null      /* decimal RDBMS version: eg, 0802010000 */
)
/
create index i_metapathmap$ on metapathmap$(name,htype,model)
/

Rem Partitoning metadata

create index i_tabpart_obj$ on tabpart$(obj#);
create index i_indpart_obj$ on indpart$(obj#);
create index i_indsubpart_obj$ on indsubpart$(obj#);

merge /*+ use_hash (tp0) */ into tabpart$ tp0 using
  (select /*+ use_hash (tp) */
     10 * row_number() over (partition by bo# order by part#) part#, obj#
   from   tabpart$ tp
   where bo# in (select obj# from partobj$ po where parttype != 2)) tp1
on (tp1.obj# = tp0.obj#)
when matched then
update set tp0.part# = tp1.part#
when not matched then
insert (obj#) values (null);

merge /*+ use_hash (ip0) */ into indpart$ ip0 using
  (select /*+ use_hash (ip) */
```

```
     10 * row_number() over (partition by bo# order by part#) part#, obj#
   from   indpart$ ip
   where bo# in (select obj# from partobj$ po where parttype != 2)) ip1
on (ip1.obj# = ip0.obj#)
when matched then
update set ip0.part# = ip1.part#
when not matched then
insert (obj#) values (null);

merge /*+ use_hash (tcp0) */ into tabcompart$ tcp0 using
  (select 10 * row_number() over (partition by bo# order by part#) part#, obj#
   from   tabcompart$ tcp) tcp1
on (tcp1.obj# = tcp0.obj#)
when matched then
update set tcp0.part# = tcp1.part#
when not matched then
insert (obj#) values (null);

merge /*+ use_hash (icp0) */ into indcompart$ icp0 using
  (select 10 * row_number() over (partition by bo# order by part#) part#, obj#
   from   indcompart$ icp) icp1
on (icp1.obj# = icp0.obj#)
when matched then
update set icp0.part# = icp1.part#
when not matched then
insert (obj#) values (null);

merge /*+ use_hash (tsp0) */ into tabsubpart$ tsp0 using
  (select /*+ use_hash (tsp) */
     10 * row_number() over (partition by pobj# order by subpart#) subpart#,
     obj#
   from   tabsubpart$ tsp
   where pobj# in (select tcp.obj# from tabcompart$ tcp, partobj$ po
                   where  tcp.bo# = po.obj# and mod(po.spare2, 256) = 4)) tsp1
on (tsp1.obj# = tsp0.obj#)
when matched then
update set tsp0.subpart# = tsp1.subpart#
when not matched then
insert (obj#) values (null);

merge /*+ use_hash (isp0) */ into indsubpart$ isp0 using
  (select /*+ use_hash (isp) */
     10 * row_number() over (partition by pobj# order by subpart#) subpart#,
     obj#
  from   indsubpart$ isp
  where pobj# in (select icp.obj# from indcompart$ icp, partobj$ po
                  where  icp.bo# = po.obj# and mod(po.spare2, 256) = 4)) isp1
on (isp1.obj# = isp0.obj#)
when matched then
update set isp0.subpart# = isp1.subpart#
when not matched then
insert (obj#) values (null);

Rem the following 2 updated must be run after the above updates
Rem since they depend on values updated above.

update lobcomppart$ lcp set part# =
  (select part# from tabcompart$ tcp
```

```
  where lcp.tabpartobj# = tcp.obj#);

update lobfrag$ lf set frag# =
  (select part# from tabpart$ tp
  where lf.tabfragobj# = tp.obj#
  union
  select subpart# from tabsubpart$ tsp
  where lf.tabfragobj# = tsp.obj#);

Rem End partitioning metadata

Rem add attribute name to table dimattr$
alter table dimattr$ add (attname varchar2(30));

Rem invalidate all dimension objects after upgrade
UPDATE obj$ SET status = 5 WHERE type# = 43
/
commit
/

Rem set ts# to 2147483647 for lob in temp tables
update lob$ set ts# = 2147483647 where bitand(property, 8) = 8
/
commit
/

Rem set charsetid and charsetform to 0 for the virtual column added by
Rem the function index if it is not of char type
update col$ set charsetid = 0, charsetform = 0 where
  bitand(property, 65576) = 65576 and type# not in (1, 8, 96, 112)
/
commit
/

Rem synonym specific policies,group,context associated with the parent synonyms
update rls$ set obj# = ptype where ptype is not null;
update rls_grp$ set obj# = synid  where synid is not null;
update rls_ctx$ set obj# = synid where synid is not null;
commit;


Rem VPD metatdata for security relevant columns
create table rls_sc$                      /* RLS secrurity relevant columns */
(
  obj#            NUMBER NOT NULL,                  /* parent object number */
  gname           VARCHAR2(30) NOT NULL,            /* name of policy group */
  pname           VARCHAR2(30) NOT NULL,                  /* name of policy */
  intcol#         NUMBER                      /* security relevant column ID */
)
/
create index i_rls_sc on rls_sc$(obj#, gname, pname)
/

Rem      -----     START FGA (FINE GRAIN AUDIT) META DATA ------

ALTER TABLE fga$ MODIFY (ptxt VARCHAR2(4000) NULL)
/
```

```
ALTER TABLE fga$ ADD (stmt_type  NUMBER default 1)
/

CREATE UNIQUE INDEX i_fgap ON fga$(obj#, pname)
/

Rem Upgrade all existing policies to  SELECT type (1)

UPDATE fga$
SET stmt_type = 1
WHERE stmt_type IS NULL;

ALTER TABLE fga$ MODIFY(stmt_type  NUMBER  default 1 NOT NULL )
/

CREATE TABLE fgacol$
(
  obj#            NUMBER NOT NULL,                    /* parent object number */
  pname           VARCHAR2(30) NOT NULL,                    /* name of policy */
  intcol#         NUMBER NOT NULL                            /* column number */
)
/

TRUNCATE TABLE fgacol$
/

CREATE UNIQUE INDEX i_fgacol ON fgacol$(obj#, pname, intcol#)
/

Rem copy relevant column information into new table

INSERT INTO fgacol$(obj# , pname, intcol#)
SELECT f.obj# , f.pname , c.col#
FROM col$ c, fga$ f
WHERE f.pcol IS NOT NULL AND f.obj# = c.obj# AND c.name = f.pcol;

REm FGA audit Trail

ALTER TABLE fga_log$ ADD (stmt_type NUMBER)
/

UPDATE fga_log$
SET stmt_type = 1
where stmt_type is NULL;

commit;

Rem      -----      End FGA metadata  ------

Rem   Begin update fga_log$  and aud$ for enhancing audit trails

Rem  Add new fine grained audit columns
ALTER TABLE fga_log$ ADD
(
    ntimestamp#           TIMESTAMP,
    proxy$sid             NUMBER,
    user$guid             VARCHAR2(32),
    instance#             NUMBER,
```

```
    process#              VARCHAR2(16),
    xid                   RAW(8),
    auditid               VARCHAR2(64),
    statement             NUMBER,
    entryid               NUMBER,
    dbid                  NUMBER,
    lsqlbind              CLOB
)
/

/* populate  ntimestamp# column */
UPDATE fga_log$
    SET  ntimestamp# = SYS_EXTRACT_UTC (
                                CAST(timestamp# AS TIMESTAMP WITH TIME ZONE)
                       )
    WHERE ntimestamp# IS NULL
/

ALTER TABLE fga_log$ MODIFY (timestamp#        DATE   NULL)
/



Rem  Add new columns to regular audit trail
Rem  AUD$ table could exist either in SYSTEM schema or in SYS schema
Rem  depending on whether the db is OLS (Oracle Label Security) enabled
Rem  or not. So, we should generate appropriate "Alter Table" statement.
DECLARE
  sql_stmt      VARCHAR2(500);
  schema_name   VARCHAR2(10);
BEGIN
  -- find out in which schema AUD$ table exists.
  SELECT u.name INTO schema_name FROM obj$ o, user$ u
         WHERE o.name = 'AUD$' AND o.type#=2 AND o.owner# = u.user#
               AND u.name IN ('SYS', 'SYSTEM');

  -- construct Alter Table statement and execute it
  sql_stmt := 'ALTER TABLE ' || schema_name || '.AUD$ ADD (' ||
                ' ntimestamp#           TIMESTAMP,'     ||
                ' proxy$sid             NUMBER,'        ||
                ' user$guid             VARCHAR2(32),'  ||
                ' instance#             NUMBER,'        ||
                ' process#              VARCHAR2(16),'  ||
                ' xid                   RAW(8),'        ||
                ' auditid               VARCHAR2(64),'  ||
                ' scn                   NUMBER,'        ||
                ' dbid                  NUMBER,'        ||
                ' sqlbind               CLOB,'          ||
                ' sqltext               CLOB'           ||
                ')';
  EXECUTE IMMEDIATE sql_stmt;
END;
/

/* populate ntimestamp# column */
UPDATE aud$
    SET  ntimestamp# = SYS_EXTRACT_UTC (
                                CAST(timestamp# AS TIMESTAMP WITH TIME ZONE)
                       )
```

```
    WHERE ntimestamp# IS NULL
/

DECLARE
  sql_stmt      VARCHAR2(500);
  schema_name   VARCHAR2(10);
BEGIN
  -- find out in which schema AUD$ table exists.
  SELECT u.name INTO schema_name FROM obj$ o, user$ u
      WHERE o.name = 'AUD$' AND o.type#=2 AND o.owner# = u.user#
            AND u.name IN ('SYS', 'SYSTEM');

  -- construct Alter Table statement and execute it
  sql_stmt := 'ALTER TABLE ' || schema_name || '.AUD$ MODIFY (' ||
              ' timestamp#           DATE  NULL'     ||
              ')';
  EXECUTE IMMEDIATE sql_stmt;
END;
/

Rem   End update fga_log$  and aud$ for enhancing audit trails

Rem=========================================================================
Rem BEGIN audit$ column value change
Rem Populate the 8 padding bytes of audit$ column with '-'
Rem=========================================================================

alter system flush shared_pool;
update tab$       set audit$ = substr(audit$, 1, 32) || '------';
update user$      set audit$ = substr(audit$, 1, 32) || '------';
update seq$       set audit$ = substr(audit$, 1, 32) ;
update view$      set audit$ = substr(audit$, 1, 32) || '------';
update procedure$ set audit$ = substr(audit$, 1, 32) || '------';
update dir$       set audit$ = substr(audit$, 1, 32) || '------';
update type_misc$ set audit$ = substr(audit$, 1, 32) || '------';
update library$   set audit$ = substr(audit$, 1, 32) || '------';
commit;

Rem=========================================================================
Rem END audit$ column value change
Rem=========================================================================

Rem Begin changes to OLAP Service catalog tables

alter table aw$ add (
 version number default null,            /* aw storage version */
 oids    number(10) default null,        /* object id page space */
 objs    number(10) default null,        /* object storage page space */
 dict    raw(8) default null             /* aw dictionary object */
);

alter table ps$ modify (psgen number(10));

alter table ps$ add (
 gelrec number default null,             /* generation erase list */
 maprec number default null              /* map record */
);
```

```
alter sequence psindex_seq$ cache 1000;

drop type body OLAP_SRF_T;

create table aw_obj$  /* Analytical Workspace Object table */
(awseq# number,                     /* aw sequence number */
oid number(20),                     /* object number */
objname varchar2(256),              /* object name, ref NAMESIZE in xsobj.c */
gen# number(10),                    /* generation number */
objtype number(4),                  /* object type */
partname varchar2(256),             /* partition name */
objdef blob,                        /* object definition */
objvalue blob,                      /* object value */
compcode blob)                      /* compiled code body */
lob(objdef) store as (enable storage in row)
lob(objvalue) store as (enable storage in row)
lob(compcode) store as (enable storage in row)
tablespace sysaux ;
create unique index i_aw_obj$ on aw_obj$ (awseq#, oid, gen#) tablespace sysaux;

create table aw_prop$ /* Analytical Workspace Property table */
(awseq# number,                         /* aw sequence number */
oid number(20),                         /* object number */
objname varchar2(256),                  /* object name */
gen# number(10),                        /* generation number */
propname varchar2(256),                 /* property name */
proptype number,                        /* property type */
propval blob)                           /* property value */
lob(propval) store as (enable storage in row)
tablespace sysaux ;
create index i_aw_prop$ on aw_prop$ (awseq#, oid) tablespace sysaux;

Rem End changes to OLAP Service catalog tables

Rem Begin changes to external table catalog tables

DROP  TYPE  ODCIExtTableInfo force;
ALTER TABLE external_tab$ ADD (PROPERTY number DEFAULT '1');
ALTER TABLE external_tab$ modify (PROPERTY number not null);

Rem End changes to external table catalog tables

Rem Begin streams changes.

Rem Begin AQ changes
Rem Set the status of all queues to invalid (workaround for 92 bug 2760010)
Rem

UPDATE obj$ SET status = 5 WHERE type# = 24
/
commit
/

DECLARE
  timezone  varchar2(64);
  stmt      varchar2(200);
BEGIN
  timezone := DBTIMEZONE;
```

```
  stmt := 'ALTER TABLE system.aq$_queue_tables ADD (TIMEZONE VARCHAR2(64) ' ||
          'DEFAULT ''' || timezone || ''')';
  EXECUTE IMMEDIATE stmt;
END;
/

alter system flush shared_pool;
alter system flush shared_pool;
alter system flush shared_pool;

Rem End AQ changes

Rem move into SYSAUX
alter table streams$_apply_progress move tablespace SYSAUX;
alter table apply$_error move tablespace SYSAUX;
alter index streams$_apply_error_unq rebuild tablespace SYSAUX;

Rem
Rem Streams capture process table
Rem TODO: migrate predumpscn value to first_scn
Rem
ALTER TABLE streams$_capture_process ADD
(
  use_dblink      number,            /* use dblink from downstream to src db */
  first_scn       number, /* initially predump scn, eventually the earliest  */
                          /* scn from which capture process can restart from */
  source_dbname   varchar2(128),                /* global name of source db */
  negative_ruleset_owner varchar2(30),           /* negative rule set owner */
  negative_ruleset_name  varchar2(30),            /* negative rule set name */
  start_date             date,                  /* captures from start date */
  end_date               date,                   /* captures up to end_date */
  error_number           number,             /* error number reported if any */
  error_message          varchar2(4000),             /* explanation of error */
  status_change_time     date,    /* the date that the status column changed */
  version                varchar2(64),             /* capture version number */
  spare4                 number,                                   /* unused */
  spare5                 number,                                   /* unused */
  spare6                 number,                                   /* unused */
  spare7                 varchar2(1000)                            /* unused */
);

Rem Set flags bit (KNLCAPF_NEED_DECOUPLE) to indicate that MVDD needs to
Rem be de-coupled from LogMiner dictionary during capture start-up.
UPDATE streams$_capture_process
  SET flags=DECODE(bitand(flags, 16), 16, flags, flags+16);
COMMIT;

Rem set capture version number to 9.2.
UPDATE streams$_capture_process
  SET version='9.2.0.0.0'
  WHERE version IS NULL;
COMMIT;

Rem add precommit_handler column to streams$_apply_process
Rem add negative rule set owner and name to streams$_apply_process
ALTER TABLE streams$_apply_process ADD
 (precommit_handler varchar2(98) default NULL,
  negative_ruleset_owner varchar2(30),            /* negative rule set owner */
```

```
  negative_ruleset_name  varchar2(30),             /* negative rule set name */
  start_date             date default NULL,         /* apply txn start limit */
  end_date               date default NULL,           /* apply txn end limit */
  error_number           number,             /* error number reported if any */
  error_message          varchar2(4000),             /* explanation of error */
  status_change_time     date     /* the date that the status column changed */
);

Rem add negative rule set owner and name to streams$_propagation_process
ALTER TABLE streams$_propagation_process ADD (
  negative_ruleset_schema    varchar2(30),        /* negative rule set owner */
  negative_ruleset           varchar2(30)          /* negative rule set name */
);

Rem add start_scn to streams$_apply_milestone
ALTER TABLE streams$_apply_milestone ADD (
  start_scn     number
);

Rem add and_condition to streams$_rules
ALTER TABLE streams$_rules ADD (
  and_condition varchar2(4000)
);

create table streams$_extra_attrs
(
  process#           number not null,                   /* capture_process # */
  name               varchar2(30) not null,                /* attribute name */
  include            varchar2(30),             /* the attribute is included? */
  flag               number,   /* 0x01 = row_attribute, 0x02 = ddl_attribute */
  spare1             number,
  spare2             varchar2(1000)
);

create unique index i_streams_extra_attrs1 on
  streams$_extra_attrs (process#, name);

rem keeps track of the streams privileges granted to a user
create table streams$_privileged_user
(
  user# number not null,     /* user number, this mapping is for user$.user# */
  privs number not null             /* the privileges granted (bit vector) : */
                                    /*   0x1 is streams administrator        */
)
/
create unique index i_streams_privileged_user1
 on streams$_privileged_user(user#)
/

rem populated by dbms_streams_adm.add_message_rule
create table streams$_message_rules
(
  streams_name    varchar2(30) not null,            /* name of apply/dequeue */
  streams_type    number not null, /* propagation(2), apply (3), dequeue (4) */
  msg_type_owner  varchar2(30),                        /* message type owner */
  msg_type_name   varchar2(30),                         /* message type name */
  msg_rule_var    varchar2(30),                     /* message rule variable */
  rule_owner      varchar2(30) not null,                        /* rule owner */
```

```
  rule_name       varchar2(30) not null,                        /* rule name */
  rule_condition  varchar2(4000),             /* text of the rule condition */
  spare1          number,
  spare2          number,
  spare3          number,
  spare4          varchar2(30),
  spare5          varchar2(128)
)
/
create unique index i_streams_message_rules
 on streams$_message_rules(streams_name, streams_type, rule_owner, rule_name)
/

rem consumers of user-enqueued messages
create table streams$_message_consumers
(
  streams_name    varchar2(30) not null,                  /* name of dequeue */
  queue_oid       raw(16)      not null,              /* AQ queue identifier */
  queue_owner     varchar2(30) not null,                       /* queue owner */
  queue_name      varchar2(30) not null,                        /* queue name */
  rset_owner      varchar2(30),                              /* rule set owner */
  rset_name       varchar2(30),                               /* rule set name */
  neg_rset_owner  varchar2(30),                   /* negative rule set owner */
  neg_rset_name   varchar2(30),                    /* negative rule set name */
  spare1          number,
  spare2          number,
  spare3          number,
  spare4          varchar2(30),
  spare5          varchar2(128)
)
/
create unique index i_streams_message_consumers
 on streams$_message_consumers(streams_name)
/

-- expand columns holding procedure names so that canonicalized names will fit
alter table sys.streams$_apply_process modify (message_handler varchar2(98));
alter table sys.streams$_apply_process modify (ddl_handler varchar2(98));
alter table sys.apply$_dest_obj_ops modify (user_apply_procedure varchar2(98));

-- allow dml handlers for virtual objects
alter table sys.apply$_dest_obj_ops add
(sname       varchar2(30),
 oname       varchar2(30),
 apply_name  varchar2(30));

create table apply$_virtual_obj_cons
(
  owner          varchar2(30)  not null,              /* source object owner */
  name           varchar2(30)  not null,               /* source object name */
  powner         varchar2(30)  not null,          /* source parent obj owner */
  pname          varchar2(30)  not null,           /* source parent obj name */
  spare1         number,
  spare2         number,
  spare3         varchar2(30),
  spare4         varchar2(4000)
);
```

```
create unique index i_apply_virtual_obj_cons on
  apply$_virtual_obj_cons (owner, name, powner, pname);

create table sys.apply$_constraint_columns
(
  owner                varchar2(30) not null,   -- object owner
  name                 varchar2(30) not null,   -- object name
  constraint_name      varchar2(30) not null,
  cname                varchar2(30) not null,   -- column name
  cpos                 number,         -- column position
  long_cname           varchar2(4000), -- long column name for adt support
  spare1               number,
  spare2               number,
  spare3               varchar2(30),
  spare4               varchar2(30)
);

create unique index sys.apply$_constraint_columns_uix1 on
  sys.apply$_constraint_columns(owner, name, constraint_name, cname);

-- to facilitate the query: given a constraint name, find out
-- all related objects
create index sys.apply$_constraint_columns_idx1 on
  sys.apply$_constraint_columns(constraint_name);

create table streams$_dest_objs
(
  object_number  number,                          /* destination table obj# */
  property       number,                        /* table property - bit flag */
           /* 0x01 : all columns specified as not to be compared for delete */
           /* 0x02 : all columns specified as not to be compared for update */
  dblink         varchar2(128),        /* database link for HS instantiation */
  spare1         number,
  spare2         number,
  spare3         varchar2(1000),
  spare4         varchar2(1000)
)
/

create unique index streams$_dest_objs_i
  on streams$_dest_objs(object_number, dblink)
/

create table streams$_dest_obj_cols
(
  object_number number,                           /* destination table obj# */
  column_name   varchar2(30),             /* name of the column for which to */
                                              /* turn conflict detection off */
  flag          number,                         /* column property - bit flag */
                                     /* 0x01 -> do not compare for deletes */
                                     /* 0x02 -> do not compare for updates */
  dblink        varchar2(128),         /* database link for HS instantiation */
  spare1        number,
  spare2        varchar2(1000)
)
/

create unique index streams$_dest_obj_cols_i
```

```
  on streams$_dest_obj_cols(object_number, column_name, dblink)
/

Rem end Streams Changes

Rem add query block identifiers to summary metadata tables
Rem Also add (rw_name, dest_stmt, rw_mode) for rewrite
Rem equivalence

alter table sum$ add
(
  numqbnodes integer,                  /* number of query block nodes */
  qbcmarker  integer,                  /* selpos of query block marker */
  markerdty  integer,                  /* query block marker data type */
  rw_name    varchar2(30),             /* name of the rewrite equivalence */
  src_stmt   clob,                     /* source stmt of rw equivalence */
  dest_stmt  clob,                     /* destination stmt of rw equivalence */
  rw_mode    integer                   /* rewrite mode of rw equivalence */
);

alter table sum$ modify (sumtext null, sumtextlen null);

alter table sumdetail$ add (qbcid number default 0 not null);
alter table suminline$ add (qbcid number default 0 not null);
alter table sumkey$    add (qbcid number default 0 not null);
alter table sumagg$    add (qbcid number default 0 not null);
alter table sumjoin$   add (qbcid number default 0 not null);
alter table sumpred$   add (qbcid number default 0 not null);
alter table sumdep$    add (qbcid number default 0 not null);

Rem recreate the indice to include query block id

begin
  execute immediate 'drop index i_sumkey$_1';
exception
   when others then
      if sqlcode = -1418 then null;
      else raise;
      end if;
end;
/

create unique index i_sumkey$_1 on sumkey$
  (sumobj#,sumcolpos#,groupingpos#,ordinalpos,qbcid);

begin
  execute immediate 'drop index i_sumagg$_1';
exception
   when others then
      if sqlcode = -1418 then null;
      else raise;
      end if;
end;
/

create unique index i_sumagg$_1 on sumagg$ (sumobj#,sumcolpos#,qbcid);

Rem create summary query node tree table and indice
```

```
create table sumqb$
(sumobj#           number not null,
 nodeid            number not null,
 pflags            number,
 xpflags           number,
 sflags            number,
 state             number,
 text              long,
 textlen           number,
 marker            varchar2(4000),
 markerlen         number,
 hashval           number,
 hashval2          number,
 rorder            number,
 sorder            number,
 leafcnt           number,
 orignode          number,
 parent            number,
 opttyp            number,
 selcnt            number,
 frompo            number,
 flags             number,
 numdetailtab      integer,
 numaggregates     integer,
 numkeycolumns     integer,
 numjoins          integer,
 numinlines        integer,
 numwhrnodes       integer,
 numhavnodes       integer);

create index i_sumqb$_1 on sumqb$(nodeid);
create index i_sumqb$_2 on sumqb$(hashval);
create index i_sumqb$_3 on sumqb$(hashval2);

Rem
Rem Begin CDC changes here
Rem

Rem Process CDC change sources

Rem Delete the predefined change source for 9i. Users had no way of creating
Rem change sources in 9i, so this is the only possible change source.
delete from cdc_change_sources$
where source_name = 'SYNC_SOURCE';

alter table cdc_change_sources$
add (
  source_type       number not null,                  /* change source type */
  source_database   varchar2(128),            /* source database global name */
  source_dbid       varchar2(16),                     /* source database ID */
  first_scn         number,               /* SCN before LogMiner dict. dump */
  first_logfile     varchar2(2000),     /* first redo log file for ManualLog */
  logfile_format    varchar2(2000),        /* later log format for ManualLog */
  publisher         varchar2(30)              /* publisher of change source */
)
modify (
  logfile_location  varchar2(2000) null         /* shorten and make nullable */
);
```

```
Rem Insert the two predefined change sources for 10.1
insert into cdc_change_sources$
  (source_name,dbid,logfile_location,logfile_suffix,source_description,created,
   source_type, source_database, source_dbid, first_scn, first_logfile,
   logfile_format, publisher)
  values('HOTLOG_SOURCE',NULL,NULL,NULL,'HOTLOG CHANGE SOURCE',SYSDATE,
         3, NULL, NULL, NULL, NULL, NULL, NULL);
insert into cdc_change_sources$
  (source_name,dbid,logfile_location,logfile_suffix,source_description,created,
   source_type, source_database, source_dbid, first_scn, first_logfile,
   logfile_format, publisher)
  values('SYNC_SOURCE',NULL,NULL,NULL,'SYNCHRONOUS CHANGE SOURCE',SYSDATE,
         4, NULL, NULL, NULL, NULL, NULL, NULL);

Rem Process CDC change sets

Rem Delete the predefined change set for 9i. Users had no way of creating
Rem change sets in 9i, so this is the only possible change set.
delete from cdc_change_sets$
where set_name = 'SYNC_SET';

alter table cdc_change_sets$
add (
  stop_on_ddl        char(1) not null,      /* Y or N - stop if DDL detected */
  capture_enabled    char(1) not null,       /* Y or N - can perform capture */
  capture_error      char(1) not null,    /* Y or N - internal capture error */
  capture_name       varchar2(30),            /* Streams capture engine name */
  queue_name         varchar2(30),                       /* AQ/Streams queue */
  queue_table_name   varchar2(30),       /* AQ/Streams spillover queue table */
  apply_name         varchar2(30),             /* Streams apply engine name */
  supplemental_procs number,        /* number of supp. processes CDC can use */
  set_description    varchar2(255),            /* description of change set */
  publisher          varchar2(30)             /* publisher of change source */
)
modify (
  advance_enabled    char(1) null,                          /* make nullable */
  ignore_ddl         char(1) null,                          /* make nullable */
  lowest_scn         number null,                           /* make nullable */
  tablespace         varchar2(30) null                      /* make nullable */
);

Rem Insert the single predefined change set for 10.1
insert into cdc_change_sets$
  (set_name, change_source_name, created, advancing, purging, stop_on_ddl,
   capture_enabled, capture_error, set_description, lowest_scn, publisher)
  values('SYNC_SET', 'SYNC_SOURCE', SYSDATE, 'N', 'N', 'N', 'Y', 'N',
         'SYNCHRONOUS CHANGE SET', 0, NULL);

Rem Process CDC subscriptions

alter table cdc_subscribers$
add (
  subscription_name varchar2(30) default 'NONE' not null,
  reserved1         number
)
modify (
  description    varchar2(255)                          /* increase length */
```

```
);

Rem drop old handle-based unique index
begin
  execute immediate 'DROP INDEX i_cdc_subscribers$';
exception
   when others then
      if sqlcode = -1418 then null;
      else raise;
      end if;
end;
/

Rem generate subscription names for old subscriptions
update cdc_subscribers$
  set subscription_name = 'CDC$SN#' || to_char(handle)
  where subscription_name = 'NONE';

Rem create new subscription_name-based unique index
begin
  execute immediate 'CREATE UNIQUE INDEX i_cdc_subscribers$ on
                     sys.cdc_subscribers$(subscription_name)';
exception
   when others then
      if sqlcode = -942 then null;
      else raise;
      end if;
end;
/

Rem add the new columns into cdc_change_tables$
ALTER TABLE cdc_change_tables$ ADD (source_table_obj# NUMBER);
ALTER TABLE cdc_change_tables$ ADD (source_table_ver NUMBER);

Rem
Rem End CDC changes here
Rem

Rem
Rem Begin online redefinition changes
Rem

rem sequence used to generate ids for online redefinitions and its steps
create sequence redef_seq$ increment by 1 start with 1 nocycle
/

rem table to store the redefinition metadata
create table redef$(
  id          integer      not null,                      /* redefinition id */
  name        varchar2(30) not null,                  /* transformation name */
  state       integer      not null,    /* current state of the redefinition */
  flag        integer                             /* flag (internal use only) */
)
/
create unique index ui_redef_id$ on redef$(id)
/
create unique index ui_redef_name$ on redef$(name)
/
```

```
rem table to store the information about the objects involved while executing
rem a redefinition
create table redef_object$(
  redef_id         integer       not null,               /* redefinition id */
  obj_type         integer       not null,                   /* object type */
  obj_owner        varchar2(30)  not null,          /* original object owner */
  obj_name         varchar2(30)  not null,           /* original object name */
  int_obj_owner    varchar2(30),              /* interim/cloned object owner */
  int_obj_name     varchar2(30),               /* interim/cloned object name */
  bt_owner         varchar2(30),                         /* base table owner */
  bt_name          varchar2(30),                          /* base table name */
  genflag          integer,                      /* flag (internal use only) */
  typflag          integer     /* obj type specific flag (internal use only) */
)
/
create index i_redef_object$ on
 redef_object$(redef_id, obj_type, obj_owner, obj_name)
/
rem table to store the dependent objects that could not be cloned during the
rem online redefinition
create table redef_dep_error$(
  redef_id       integer            not null,             /* redefinition id */
  obj_type       integer            not null,                 /* object type */
  obj_owner      varchar2(30)       not null,       /* original object owner */
  obj_name       varchar2(30)       not null,        /* original object name */
  bt_owner       varchar2(30),                           /* base table owner */
  bt_name        varchar2(30),                            /* base table name */
  ddl_txt        clob                                         /* ddl string */
)
/
create index i_redef_dep_error$ on
 redef_dep_error$(redef_id, obj_type, obj_owner, obj_name)
/

Rem
Rem End online redefinition changes
Rem

Rem Clear the monitoring bit (obsolete)
Rem #(3272499) Also mark the mon_mods$ entries with potentially dubious
Rem statistics then gather_stats_job will run further checks later.
Rem (exclude: iot overflow, temp, external, iot mapping tables)

alter system flush shared_pool;

declare
  is_reupgrade pls_integer;
begin
  select count(*) into is_reupgrade
  from obj$
  where owner#=0
    and name='TAB_STATS$'
    and type#=2;

  -- don't do this if re-upgrade
  if (is_reupgrade = 0) then
    merge into sys.mon_mods$ m
```

```
      using
     (select /*+ dynamic_sampling(4) dynamic_sampling_est_cdn */
        tab.obj# obj#, 0 inserts, 0 updates, 0 deletes, sysdate timestamp,
        2 flags, 0 drop_segments
      from
        (select obj# from sys.tab$             /* non-partitoined tables */
           where bitand(property,32+512+4194304+8388608+2147483648) = 0
             and bitand(flags,536870912) = 0
             and bitand(flags,2097152) = 0     /* monitoring is off */
             and bitand(flags,16) != 0         /* analyzed */
        union all                              /* table partitions */
        select tp.obj# from sys.tabpart$ tp, sys.tab$ t
           where tp.bo# = t.obj#
             and bitand(t.flags,2097152) = 0   /* monitoring is off */
             and bitand(tp.flags,2) != 0       /* analyzed */
        union all                              /* table subpartitions */
        select tsp.obj# from sys.tabsubpart$ tsp,
                             sys.tabcompart$ tp, sys.tab$ t
           where tsp.pobj# = tp.obj# and tp.bo# = t.obj#
             and bitand(t.flags,2097152) = 0   /* monitoring is off */
             and bitand(tsp.flags,2) != 0      /* analyzed */
        ) tab
      ) v on (m.obj# = v.obj#)
      when matched then
        update set flags = flags - bitand(flags,2) + 2
      when NOT matched then
        insert values
          (v.obj#, v.inserts, v.updates, v.deletes, v.timestamp,
           v.flags, v.drop_segments);
  end if;
end;
/

UPDATE sys.tab$ t SET flags = flags - 2097152
  WHERE bitand(t.flags, 2097152) = 2097152;
commit;

Rem ========================================================================
Rem Update IND$ to reset ts# for indices on temporary tables.
Rem Flags to check are 0x400000 and 0x800000, global and session flags.
Rem ========================================================================

alter system flush shared_pool;
update ind$ set ts# = 0
where  ts# != 0 and
       bo# in (select obj# from tab$
               where  bitand(property, 12582912) != 0);

rem fixed object (X$...) information
create table fixed_obj$
( obj#          number not null,                          /* object number */
  timestamp     date not null,             /* object specification timestamp */
  flags         number,                    /* 0x00000001 = analyzed
                                              0x00000002 = locked            */
  spare1        number,
  spare2        number,
  spare3        number,
  spare4        varchar2(1000),
```

```
  spare5        varchar2(1000),
  spare6        date
)
  storage (maxextents unlimited)
/
create unique index i_fixed_obj$_obj# on fixed_obj$(obj#)
  storage (maxextents unlimited)
/

rem table to store optimizer statistics for table and table partition objects
create table tab_stats$
( obj#          number not null,                          /* object number */
  cachedblk     number,                             /* blocks in buffer cache */
  cachehit      number,                                    /* cache hit ratio */
  logicalread   number,                            /* number of logical reads */
  rowcnt        number,                                     /* number of rows */
  blkcnt        number,                                   /* number of blocks */
  empcnt        number,                             /* number of empty blocks */
  avgspc        number,       /* average available free space/iot ovfl stats */
  chncnt        number,                             /* number of chained rows */
  avgrln        number,                                 /* average row length */
  avgspc_flb    number,       /* avg avail free space of blocks on free list */
  flbcnt        number,                              /* free list block count */
  analyzetime   date,                         /* timestamp when last analyzed */
  samplesize    number,                 /* number of rows sampled by Analyze */
  flags         number,                 /* 0x00000001 = user-specified stats */
  spare1        number,
  spare2        number,
  spare3        number,
  spare4        varchar2(1000),
  spare5        varchar2(1000),
  spare6        date
)
  storage (initial 32k next 100k maxextents unlimited pctincrease 0)
/
create unique index i_tab_stats$_obj# on tab_stats$(obj#)
  storage (maxextents unlimited)
/

rem table to store optimizer statistics for index and index partition objects
create table ind_stats$
( obj#          number not null,                          /* object number */
  cachedblk     number,                             /* blocks in buffer cache */
  cachehit      number,                                    /* cache hit ratio */
  logicalread   number,                            /* number of logical reads */
  rowcnt        number,                        /* number of rows in the index */
  blevel        number,                                     /* btree level */
  leafcnt       number,                                  /* # of leaf blocks */
  distkey       number,                                  /* # distinct keys */
  lblkkey       number,                            /* avg # of leaf blocks/key */
  dblkkey       number,                            /* avg # of data blocks/key */
  clufac        number,                                  /* clustering factor */
  analyzetime   date,                         /* timestamp when last analyzed */
  samplesize    number,                 /* number of rows sampled by Analyze */
  flags         number,
  spare1        number,
  spare2        number,
  spare3        number,
```

```
  spare4        varchar2(1000),
  spare5        varchar2(1000),
  spare6        date
)
  storage (initial 32k next 100k maxextents unlimited pctincrease 0)
/
create unique index i_ind_stats$_obj# on ind_stats$(obj#)
  storage (maxextents unlimited)
/

Rem
Rem Bigfile Tablespace changes
Rem
delete from sys.props$ where name = 'DEFAULT_TBS_TYPE';
insert into sys.props$
  values('DEFAULT_TBS_TYPE', 'SMALLFILE', 'Default tablespace type');
Rem table used to store array type info supported by the indextype
create table indarraytype$
( obj#                  number not null,                    /* indextype obj# */
  type                  number not null,      /* data type of indexed column */
                                             /* for ADT column, type# = DTYADT */
  basetypeobj#          number,         /* object number of user-defined type */
  arraytypeobj#         number not null,      /* object number of array type */
  spare1                number,
  spare2                number
)
/
Rem
Rem Add alias_txt to snap$
Rem
alter table snap$ add (alias_txt clob);

Rem Move proxy metadata to tables outside bootstrap region
create table proxy_info$
( client#             NUMBER NOT NULL,                      /* client user ID */
  proxy#              NUMBER NOT NULL,                       /* proxy user ID */
  credential_type#    NUMBER NOT NULL,  /* Type of credential passed by proxy */
                    /*
                     * Values
                     * 0 = No Authentication
                     * 5 = Authentication
                     */
  flags                 NUMBER NOT NULL /* Mask flags of associated with entry */
             /* Flags values:
              * 1 = proxy can activate all client roles
              * 2 = proxy can activate no client roles
              * 4 = role can be activated by proxy,
              * 8 = role cannot be activated by proxy
              */
)
/
insert into proxy_info$
  select client#,
         proxy#,
         decode(credential_type#, 0, 0,/* No Credential => No Authentication */
                                  1, 0,       /* Certificate => No Credential */
                                  2, 0, /*Distinguished Name => No Credential*/
                                  4, 5),/* Oracle Password => Authentication */
```

```
        flags
  from proxy_data$;

delete from proxy_data$;

create unique index i_proxy_info$ on proxy_info$(client#, proxy#)
/
create table proxy_role_info$
( client#       NUMBER NOT NULL,                          /* client user ID */
  proxy#        NUMBER NOT NULL,                           /* proxy user ID */
  role#         NUMBER NOT NULL                                 /* role ID */
)
/
insert into proxy_role_info$
  select client#, proxy#, role# from proxy_role_data$;

delete from proxy_role_data$;

create index i_proxy_role_info$_1 on
  proxy_role_info$(client#, proxy#)
/
create unique index i_proxy_role_info$_2 on
  proxy_role_info$(client#, proxy#, role#)
/

Rem
Rem Begin operator$ changes
Rem

Rem Add a column to the operator$ table to keep track of the next
Rem available binding number
ALTER TABLE operator$ ADD (nextbindnum number default 0 not null);

Rem Populate the nextbindnum field
UPDATE operator$ SET nextbindnum = numbind + 1;

Rem
Rem End operator$ changes
Rem



Rem
Rem start Resource Manager changes
Rem

alter table resource_plan_directive$ add
(
  max_idle_time             number,                 /* max. idle time in sec */
  max_idle_blocker_time     number,    /* max. idle time blocking other sess */
  switch_back               number        /* switch back at end of top call? */
)
/
update resource_plan_directive$ set
  max_idle_time = 4294967295,
  max_idle_blocker_time = 4294967295,
  switch_back = 0
/
create table resource_group_mapping$
```

```
( attribute           varchar2(30),               /* mapping attribute type */
  value               varchar2(128),             /* attribute value to match */
  consumer_group      varchar2(30),               /* name of consumer group */
  status              varchar2(30)              /* whether active or pending */
)
/
create table resource_mapping_priority$
( attribute           varchar2(30),               /* mapping attribute type */
  priority            number,                  /* priority of mapping (1 - 8) */
  status              varchar2(30)              /* whether active or pending */
)
/
truncate table resource_group_mapping$
/
truncate table resource_mapping_priority$
/
insert into resource_group_mapping$
  (attribute, value, consumer_group, status)
  (select 'ORACLE_USER', name,  defschclass, 'ACTIVE' from user$
   where defschclass is not null and defschclass != 'DEFAULT_CONSUMER_GROUP')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('EXPLICIT', 1, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('ORACLE_USER', 7, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('SERVICE_NAME', 6, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('CLIENT_OS_USER', 9, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('CLIENT_PROGRAM', 8, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('CLIENT_MACHINE', 10, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('MODULE_NAME', 5, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('MODULE_NAME_ACTION', 4, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('SERVICE_MODULE', 3, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('SERVICE_MODULE_ACTION', 2, 'ACTIVE')
/
insert into resource_mapping_priority$ (attribute, priority, status)
  values ('CLIENT_ID', 11, 'ACTIVE')
/
commit
/

Rem
```

```
Rem end Resource Manager changes
Rem

Rem
Rem Add the service$ table
Rem
create table service$
(
  service_id         number,                                  /* unique ID */
  name               varchar2(64),                           /* short name */
  name_hash          number,                          /* service name hash */
  network_name       varchar2(512),           /* SERVICE_NAME as used by net */
  failover_method    varchar2(64),            /* TAF failover characterstics */
  failover_type      varchar2(64),
  failover_retries   number(10),
  failover_delay     number(10),
  creation_date      date,                                  /* date created */
  creation_date_hash number,                           /* creation date hash */
  deletion_date      date                               /* date marked deleted */
)
/
rem Create the internal system service
delete from service$
where name = 'SYS$BACKGROUND' or name = 'SYS$USERS'
/

insert into service$
  (service_id, name, creation_date)
  values (1, 'SYS$BACKGROUND', sysdate)
/

insert into service$
  (service_id, name, creation_date)
  values (2, 'SYS$USERS', sysdate)
/

rem table used by import and export for storing xml format of export
rem metadata when doing transportable tablespaces.
create table expimp_tts_ct$(
  owner         varchar2(30) not null,                       /* table owner */
  tablename     varchar2(30) not null,                        /* table name */
  xmlinfo       clob              not null, /* table's metadata from export */
  when          timestamp         not null                   /* for safety */
)
/

Rem=========================================================================
Rem BEGIN director changes
Rem=========================================================================

Rem table used by director that contains all databases in cluster
create global temporary table cluster_databases(
    database_name varchar2(128),
    sparen1       number,
    sparen2       number,
    sparevc1      varchar2(4000),
    sparevc2      varchar2(4000))
  on commit preserve rows
```

```
/

Rem table used by director that contains all nodes in cluster
create global temporary table cluster_nodes(
    node_name varchar2(4000),
    sparen1   number,
    sparen2   number,
    sparevc1  varchar2(4000),
    sparevc2  varchar2(4000))
  on commit preserve rows
/

Rem table used by director that contains all running instances in cluster
create global temporary table cluster_instances(
    instance_number number,
    database_name   varchar2(128),
    inst_name       varchar2(4000),
    node_name       varchar2(4000),
    sparen1         number,
    sparen2         number,
    sparevc1        varchar2(4000),
    sparevc2        varchar2(4000))
  on commit preserve rows
/

Rem table used by director for migrate operations
create table dir$migrate_operations(
   job_name         varchar2(100),
   alert_seq_id     number,
   incarnation_info varchar2(4000),
   service_name     varchar2(4000),
   source_instance  varchar2(4000),
   dest_instance    varchar2(4000),
   session_count    number,
   director_factor  number,
   submit_time      date,
   status           number,
   start_time       date,
   end_time         date,
   actual_count     number,
   error_message    varchar2(4000),
   sparen1          number,
   sparen2          number,
   sparen3          number,
   sparen4          number,
   sparen5          number,
   sparevc1         varchar2(4000),
   sparevc2         varchar2(4000),
   sparevc3         varchar2(4000),
   sparevc4         varchar2(4000),
   sparevc5         varchar2(4000))
tablespace sysaux
/
create unique index sys.i_dir$migrate_ui
  on sys.dir$migrate_operations(job_name)
  tablespace sysaux
/
create index sys.i_dir$migrate_end_time
```

```
  on sys.dir$migrate_operations(end_time)
  tablespace sysaux
/
create index sys.i_dir$migrate_alert_seq_id
  on sys.dir$migrate_operations(alert_seq_id)
  tablespace sysaux
/

Rem table used by director for service operations
create table dir$service_operations(
   job_name         varchar2(100),
   alert_seq_id     number,
   job_type         number,
   incarnation_info varchar2(4000),
   service_name     varchar2(4000),
   instance_name    varchar2(4000),
   director_factor  number,
   submit_time      date,
   status           number,
   start_time       date,
   end_time         date,
   error_message    varchar2(4000),
   sparen1          number,
   sparen2          number,
   sparen3          number,
   sparen4          number,
   sparen5          number,
   sparevc1         varchar2(4000),
   sparevc2         varchar2(4000),
   sparevc3         varchar2(4000),
   sparevc4         varchar2(4000),
   sparevc5         varchar2(4000))
tablespace sysaux
/
create unique index sys.i_dir$service_ui
  on sys.dir$service_operations(job_name)
  tablespace sysaux
/
create index sys.i_dir$service_end_time
  on sys.dir$service_operations(end_time)
  tablespace sysaux
/
create index sys.i_dir$service_alert_seq_id
  on sys.dir$service_operations(alert_seq_id)
  tablespace sysaux
/

rem table used by director for escalate operations
rem this is used to keep track of escalations from
rem the database director to the cluster director
create table dir$escalate_operations(
   escalation_id    varchar2(200),
   alert_seq_id     number,
   escalation       VARCHAR2(20),
   incarnation_info varchar2(4000),
   instance_name    varchar2(4000),
   submit_time      date,
   status           number,
```

```
   start_time       date,
   end_time         date,
   retry_time       date,
   retry_count      number,
   error_message    varchar2(4000),
   sparen1          number,
   sparen2          number,
   sparen3          number,
   sparen4          number,
   sparen5          number,
   sparevc1         varchar2(4000),
   sparevc2         varchar2(4000),
   sparevc3         varchar2(4000),
   sparevc4         varchar2(4000),
   sparevc5         varchar2(4000))
tablespace sysaux
/
create unique index sys.i_dir$escalate_ui
  on sys.dir$escalate_operations(escalation_id)
  tablespace sysaux
/
create index sys.i_dir$escalate_end_time
  on sys.dir$escalate_operations(end_time)
  tablespace sysaux
/
create index sys.i_dir$escalate_alert_seq_id
  on sys.dir$escalate_operations(alert_seq_id)
  tablespace sysaux
/

rem table used by database director for
rem recording quiesce operations
create table dir$quiesce_operations
(
   job_name         varchar2(100),
   alert_seq_id     number,
   job_type         number,
   incarnation_info varchar2(4000),
   instance_name    varchar2(4000),
   submit_time      date,
   status           number,
   start_time       date,
   end_time         date,
   error_message    varchar2(4000),
   sparen1          number,
   sparen2          number,
   sparen3          number,
   sparen4          number,
   sparen5          number,
   sparevc1         varchar2(4000),
   sparevc2         varchar2(4000),
   sparevc3         varchar2(4000),
   sparevc4         varchar2(4000),
   sparevc5         varchar2(4000))
tablespace sysaux
/

create unique index sys.i_dir$quiesce_ui
```

```
  on sys.dir$quiesce_operations(job_name)
  tablespace sysaux
/
create index sys.i_dir$quiesce_status
  on sys.dir$quiesce_operations(status)
  tablespace sysaux
/
create index sys.i_dir$quiesce_end_time
  on sys.dir$quiesce_operations(end_time)
  tablespace sysaux
/
create index sys.i_dir$quiesce_alert_seq_id
  on sys.dir$quiesce_operations(alert_seq_id)
  tablespace sysaux
/

rem table used by database director for
rem recording specific instance actions
rem done by a job
create table dir$instance_actions
(
   job_name         varchar2(100),
   action_type      number,
   instance_name    varchar2(4000),
   submit_time      date,
   start_time       date,
   end_time         date,
   error_message    varchar2(4000),
   sparen1          number,
   sparen2          number,
   sparen3          number,
   sparen4          number,
   sparen5          number,
   sparevc1         varchar2(4000),
   sparevc2         varchar2(4000),
   sparevc3         varchar2(4000),
   sparevc4         varchar2(4000),
   sparevc5         varchar2(4000))
tablespace sysaux
/


create index sys.i_dir$instance_job_name
  on sys.dir$instance_actions(job_name)
  tablespace sysaux
/

create index sys.i_dir$instance_acttyp
  on sys.dir$instance_actions(action_type)
  tablespace sysaux
/

create index sys.i_dir$instance_end_time
  on sys.dir$instance_actions(end_time)
  tablespace sysaux
/
```

```
Rem table used by director for resonate operations
create table dir$resonate_operations
(
   job_name         varchar2(100),
   alert_name       varchar2(200),
   job_type         number,
   incarnation_info varchar2(4000),
   database_name    varchar2(128),
   instance_name    varchar2(4000),
   node_name        varchar2(4000),
   submit_time      date,
   status           number,
   start_time       date,
   end_time         date,
   error_message    varchar2(4000),
   priority         number,
   sparen1          number,
   sparen2          number,
   sparen3          number,
   sparen4          number,
   sparen5          number,
   sparevc1         varchar2(4000),
   sparevc2         varchar2(4000),
   sparevc3         varchar2(4000),
   sparevc4         varchar2(4000),
   sparevc5         varchar2(4000))
tablespace sysaux
/
create unique index sys.i_dir$resonate_ui
  on sys.dir$resonate_operations(job_name)
  tablespace sysaux
/
create index sys.i_dir$resonate_status
  on sys.dir$resonate_operations(status)
  tablespace sysaux
/
create index sys.i_dir$resonate_end_time
  on sys.dir$resonate_operations(end_time)
  tablespace sysaux
/
create index sys.i_dir$resonate_alert_name
  on sys.dir$resonate_operations(alert_name)
  tablespace sysaux
/

rem table used by director for database priorities
create table dir$database_attributes
(
  database_name        varchar2(128),
  attribute_name       varchar2(30),
  attribute_value      varchar2(4000),
  sparen1              number,
  sparen2              number,
  sparen3              number,
  sparen4              number,
  sparen5              number,
  sparevc1             varchar2(4000),
  sparevc2             varchar2(4000),
```

```
  sparevc3             varchar2(4000),
  sparevc4             varchar2(4000),
  sparevc5             varchar2(4000))
tablespace sysaux
/
create unique index sys.i_dir$db_attributes_ui
  on sys.dir$database_attributes(database_name)
  tablespace sysaux
/

rem table used by director for victim database policy function
create table dir$victim_policy
(
  user_name            varchar2(30),
  policy_function_name varchar2(98),
  version              number,
  sparen1              number,
  sparen2              number,
  sparen3              number,
  sparen4              number,
  sparen5              number,
  sparen6              number,
  sparen7              number,
  sparevc1             varchar2(4000),
  sparevc2             varchar2(4000),
  sparevc3             varchar2(4000),
  sparevc4             varchar2(4000),
  sparevc5             varchar2(4000))
tablespace sysaux
/

Rem Table for keeping node attributes
create table dir$node_attributes
( node_name            varchar2(4000),
  attribute_name       varchar2(30),
  attribute_value      varchar2(4000),
  sparen1              number,
  sparen2              number,
  sparen3              number,
  sparen4              number,
  sparen5              number,
  sparevc1             varchar2(4000),
  sparevc2             varchar2(4000),
  sparevc3             varchar2(4000),
  sparevc4             varchar2(4000),
  sparevc5             varchar2(4000))
tablespace sysaux
/

create index sys.i_dir$node_attributes_attr
  on sys.dir$node_attributes(attribute_name)
  tablespace sysaux
/

Rem Table for keeping service attributes
create table dir$service_attributes
( service_id           number,
  attribute_name       varchar2(30),
```

```
  attribute_value      varchar2(4000),
  sparen1              number,
  sparen2              number,
  sparen3              number,
  sparen4              number,
  sparen5              number,
  sparevc1             varchar2(4000),
  sparevc2             varchar2(4000),
  sparevc3             varchar2(4000),
  sparevc4             varchar2(4000),
  sparevc5             varchar2(4000))
tablespace sysaux
/
create index sys.i_dir$service_attributes_serv
  on sys.dir$service_attributes(service_id)
  tablespace sysaux
/

create index sys.i_dir$service_attributes_attr
  on sys.dir$service_attributes(attribute_name)
  tablespace sysaux
/

Rem=========================================================================
Rem END director changes
Rem=========================================================================

Rem=========================================================================
Rem Add changes to other SYS dictionary objects here
Rem=========================================================================

Rem Add namespace to registry$

ALTER TABLE registry$ ADD
(
    namespace   VARCHAR2(30),
    org_version VARCHAR2(30),
    prv_version VARCHAR2(30)
);

BEGIN
  EXECUTE IMMEDIATE
    'UPDATE registry$ set namespace = ''SERVER'' where namespace IS NULL';
  COMMIT;
EXCEPTION
  WHEN OTHERS THEN
    IF sqlcode = -942 THEN NULL; -- registry$ does not exist, pre-92 db
    ELSE RAISE;
    END IF;
END;
/

ALTER TABLE registry$ DROP CONSTRAINT registry_parent_fk;
ALTER TABLE registry$ DROP CONSTRAINT registry_pk;

ALTER TABLE registry$ ADD CONSTRAINT registry_pk
      PRIMARY KEY (namespace, cid);
```

```
ALTER TABLE registry$ ADD CONSTRAINT registry_parent_fk
      FOREIGN KEY (namespace, pid)
      REFERENCES registry$ (namespace, cid)
      ON DELETE CASCADE;

drop public synonym XMLConcat;
drop function xmlconcat;

Rem ========================================
Rem Remove public grant on DBA_PROCEDURES
Rem ========================================

BEGIN
  EXECUTE IMMEDIATE
   'REVOKE SELECT on DBA_PROCEDURES from PUBLIC';
EXCEPTION
   WHEN OTHERS THEN
      IF SQLCODE IN (-942, -1917, -1918, -1919, -1951, -1952) THEN NULL;
      ELSE RAISE;
      END IF;
END;
/

Rem ========================================
Rem smon_scn_time table
Rem See comments in sql.bsq
Rem ========================================

Rem add columns to 9.0.2 table
alter table sys.smon_scn_time
ADD
(
   num_mappings number,
   tim_scn_map  raw(1200) default null
)
/

Rem add columns to 10.1 beta1 table
alter table sys.smon_scn_time
ADD
(
   scn number default 0,                  /* scn */
   orig_thread number default 0           /* for downgrade */
)
/

update smon_scn_time set scn = scn_wrp * 4294967295 + scn_bas where scn=0;
update smon_scn_time set orig_thread=thread, thread=0
       where orig_thread=0 and thread<>0;



rem
rem transparent session migration
rem
create table tsm_hist$
(
  source_sid           number,              /* session id on source instance */
  source_serial#       number,                 /* serial# on source instance */
```

```
  state                number,                          /* migration state */
  cost                 number,                  /* estimated migration cost */
  source               varchar2(4000),                  /* source instance */
  destination          varchar2(4000),              /* destination instance */
  connect_string       varchar2(4000),         /* destination connect string */
  failure_reason       number,            /* reason for failure of migration */
  destination_sid      number,          /* session id on destination instance */
  destination_serial#  number,            /* serial# on destination instance */
  start_time           date,                          /* migration start time */
  end_time             date                             /* migration end time */
)
tablespace SYSAUX
/
create index i_tsm_hist1 on tsm_hist$(source_sid, source_serial#)
tablespace SYSAUX
/


Rem Set Logical Standby bit in tab$ & seq$ to ensure tables are always guarded.
alter system flush shared_pool;
UPDATE SYS.TAB$ SET FLAGS = FLAGS + 1073741824
WHERE BITAND(FLAGS, 1073741824) != 1073741824    /* lsby bit not already set */
  AND BITAND(PROPERTY, 4194304) != 4194304                /* not temp table */
  AND OBJ# IN
   (SELECT O.OBJ# FROM SYS.OBJ$ O, SYS.USER$ U
    WHERE U.USER# = O.OWNER# AND O.TYPE# = 2
      AND U.NAME != 'SYS' AND U.NAME != 'SYSTEM'
      AND U.NAME != 'OUTLN' AND U.NAME != 'DBSNMP');
COMMIT;
alter system flush shared_pool;
UPDATE SYS.SEQ$ SET FLAGS = FLAGS + 8
WHERE BITAND(FLAGS, 8) != 8                      /* lsby bit not already set */
  AND OBJ# IN
   (SELECT O.OBJ# FROM SYS.OBJ$ O, SYS.USER$ U
    WHERE U.USER# = O.OWNER# AND O.TYPE# = 6
      AND U.NAME != 'SYS' AND U.NAME != 'SYSTEM'
      AND U.NAME != 'OUTLN' AND U.NAME != 'DBSNMP');
COMMIT;
alter system flush shared_pool;

Rem ========================================================================
Rem upgrade rules engine objects
Rem ========================================================================

ALTER TABLE sys.rec_tab$
ADD
(
      tab_id            number,                      /* index of table alias */
      tab_obj#          number                        /* table object number */
)
/

ALTER TABLE sys.rec_var$
ADD
(
      var_id            number,                        /* index of variable */
      var_dty           number,                                 /* oacdty */
      precision#        number,                               /* precision */
```

```
      scale             number,                                  /* scale */
      maxlen            number,                            /* maximum length */
      charsetid         number,                       /* NLS character set id */
      charsetform       number,                         /* character set form */
      toid              raw(16),                              /* OID for ADTs */
      version           number,                      /* TOID version for ADTs */
      num_attrs         number      /* number of flattened attributes in var */
)
/

DECLARE
  INDEX_NOT_EXIST exception;
  pragma          EXCEPTION_INIT(INDEX_NOT_EXIST, -1418);
BEGIN
  EXECUTE IMMEDIATE 'DROP INDEX sys.i_rec_tab';
EXCEPTION
  WHEN INDEX_NOT_EXIST THEN
    NULL;
END;
/

DECLARE
  INDEX_NOT_EXIST exception;
  pragma          EXCEPTION_INIT(INDEX_NOT_EXIST, -1418);
BEGIN
  EXECUTE IMMEDIATE 'DROP INDEX sys.i_rec_var';
EXCEPTION
  WHEN INDEX_NOT_EXIST THEN
    NULL;
END;
/

UPDATE sys.obj$ SET status = 5
where obj# in
  ((select obj# from obj$ where type# = 62 or type# = 46)
   union all
   (select /*+ index (dependency$ i_dependency2) */
      d_obj# from dependency$
      connect by prior d_obj# = p_obj#
      start with p_obj# in
        (select obj# from obj$ where type# = 62 or type# = 46)))
/
commit
/

-- Remove lcr$_row_record methods whose signatures have changed
-- between 9.2.0.1 and higher releases
DECLARE
   version       varchar2(30);
   alt_typ_stmt  varchar2(500);
BEGIN
  EXECUTE IMMEDIATE
    'SELECT substr(dbms_registry.version(''CATPROC''),1,7) FROM DUAL'
      INTO version;

  IF version = '9.2.0.1' THEN
    alt_typ_stmt :=
      'ALTER TYPE sys.lcr$_row_record DROP MEMBER FUNCTION ' ||
```

```
      '  get_value (value_type IN VARCHAR2, column_name IN VARCHAR2) ' ||
      '  RETURN sys.AnyData CASCADE';
    EXECUTE IMMEDIATE alt_typ_stmt;

    alt_typ_stmt :=
      'ALTER TYPE sys.lcr$_row_record DROP MEMBER FUNCTION ' ||
      '  get_values (value_type IN VARCHAR2) ' ||
      '  RETURN sys.lcr$_row_list CASCADE';
    EXECUTE IMMEDIATE alt_typ_stmt;

    alt_typ_stmt :=
      'ALTER TYPE sys.lcr$_row_record DROP MEMBER FUNCTION ' ||
      '  get_lob_information (value_type IN VARCHAR2,' ||
      '  column_name IN VARCHAR2) RETURN NUMBER CASCADE';
    EXECUTE IMMEDIATE alt_typ_stmt;
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    IF sqlcode = -904 THEN NULL; /* dbms_registry package does not
                                  * exist, pre-92 db
                                  */
    ELSE RAISE;
    END IF;
END;
/

Rem=========================================================================
Rem  Add changes to SYSTEM objects here
Rem=========================================================================
Rem Begin Logical Standby changes.

Rem
Rem Logical Standby SCN table
Rem
ALTER TABLE system.logstdby$scn add
(
  objname   varchar2(4000),                                 /* Object name */
  schema    varchar2(30),                                   /* Schema name */
  type      varchar2(20)
);

Rem
Rem Logical Standby Skip table
Rem
ALTER TABLE system.logstdby$skip add
(
  use_like  number,       /* 0 = exact match, 1 = like, 2 = like with escape */
  esc       varchar2(1)                    /* Escape character if using like */
);

Rem
Rem Logical Standby apply_milestone table
Rem
ALTER TABLE system.logstdby$apply_milestone add
(
  fetchlwm_scn    number default(0) not null     /* maximum SCN ever fetched */
);
Rem End Logical Standby changes.
```

```
ALTER TABLE system.aq$_queues ADD (service_name VARCHAR2(64));
ALTER TABLE system.aq$_queues ADD (network_name VARCHAR2(256));

ALTER TABLE sys.aq$_message_types ADD (network_name VARCHAR2(256));


ALTER TABLE sys.aq$_replay_info ADD (ack NUMBER);

Rem Evolve Type sys.aq$_reg_info

ALTER TYPE sys.aq$_reg_info
ADD ATTRIBUTE(anyctx SYS.ANYDATA, ctxtype NUMBER) CASCADE;

ALTER TYPE sys.aq$_reg_info ADD CONSTRUCTOR FUNCTION aq$_reg_info(
  name             VARCHAR2,
  namespace        NUMBER,
  callback         VARCHAR2,
  context          RAW)
RETURN SELF AS RESULT CASCADE;

ALTER TYPE sys.aq$_srvntfn_message
ADD ATTRIBUTE(anysub_context SYS.ANYDATA, context_type NUMBER) CASCADE;

Rem TODO: Fix later
Rem UPDATE SYS.AQ_SRVNTFN_TABLE tab
Rem SET tab.user_data.context_type = 0;

Rem Begin Summary Advisor changes.

alter table system.mview$_adv_workload
  modify (application varchar2(64));

Rem End Summary Advisor changes.

Rem ========================================================================
Rem Upgrade system types to 10.1
Rem ========================================================================

Rem Upgrading the type manager to refresh to the latest
Rem

Rem c1001000.sql contributions START here. tbgraves

Rem=========================================================================
Rem Add new system privileges here
Rem=========================================================================

Rem Add SQL Tuning Base privileges
insert into SYSTEM_PRIVILEGE_MAP values (-274, 'CREATE ANY SQL PROFILE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-270, 'DROP ANY SQL PROFILE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-271, 'ALTER ANY SQL PROFILE', 0);

Rem add sql tuning Set privileges
insert into SYSTEM_PRIVILEGE_MAP
       values (-272, 'ADMINISTER SQL TUNING SET', 0);
insert into SYSTEM_PRIVILEGE_MAP
```

```
       values (-273, 'ADMINISTER ANY SQL TUNING SET', 0);

Rem add secure client_id privilege
insert into SYSTEM_PRIVILEGE_MAP values (-275, 'EXEMPT IDENTITY POLICY', 0);

grant all privileges, analyze any dictionary to dba with admin option;

Rem=========================================================================
Rem Add new object privileges here
Rem=========================================================================

Rem A grant/revoke on a nested table column is supposed
Rem to propagate the privileges to all the underlying nested tables.
Rem Prior to 10.1, these privileges were being propagated only to the
Rem first level nested tables. The following piece of code fixes the
Rem nested table while upgrading to 10.1 in case privileges had been
Rem granted on the first level tables.

create or replace procedure u$grant$nested$priv
          (pobj in number) is
  ntabobjn  number;
  cnt       integer;
  cursor c2(pobj number) is select ntab# from ntab$ where obj#=pobj;
begin
  open c2(pobj);
  loop
    fetch c2 into ntabobjn;
    exit when c2%NOTFOUND;

    -- If there is already a row present, ignore it
    select count(*) into cnt
    from objauth$ where obj# = ntabobjn;

    if (cnt = 0) then
       -- Add rows for this table.
       insert into objauth$ (obj#, grantor#, grantee#, privilege#,
                             sequence#, parent, option$, col#)
       select ntabobjn, grantor#, grantee#, privilege#, object_grant.nextval,
              parent, option$, col#
       from objauth$ where obj#=pobj;

       -- Recurse to find nested tables
       u$grant$nested$priv(ntabobjn);
    end if;
  end loop;
end;
/

declare

  n        integer;
  pobj     number;
  grantor  number;
  grantee  number;
  priv     number;
  options  number;
  cursor c1 is select a.obj#
               from tab$ t, objauth$ a
```

```
              where t.obj#=a.obj# and bitand(t.property, 8192) = 8192 and
              bitand(t.property, 4) = 4;

begin

  -- Get number of first level nested tables which were granted privileges
  -- in prior releases and which have further nested tables under them.
  -- If this count is zero, we have nothing to do.

  select count(*) into n
  from tab$ t, objauth$ a
  where t.obj#=a.obj# and
        bitand(t.property, 8192) = 8192 and
        bitand(t.property, 4) = 4;

  if (n > 0) then

    -- Open a cursor to fetch each such first level nested table
    -- For each parent table, call a recursive procedure to grant
    -- privileges to the nested tables.

    open c1;
    loop
      fetch c1 into pobj;
      exit when c1%NOTFOUND;
      u$grant$nested$priv(pobj);
    end loop;

  end if;
end;
/

drop procedure u$grant$nested$priv;

Rem================i============================================================
Rem Add new audit options here
Rem=====================================================================

Rem Add SQL Tuning Base options
insert into STMT_AUDIT_OPTION_MAP values (274, 'CREATE ANY SQL PROFILE',0);
insert into STMT_AUDIT_OPTION_MAP values (270, 'DROP ANY SQL PROFILE',0);
insert into STMT_AUDIT_OPTION_MAP values (271, 'ALTER ANY SQL PROFILE',0);

Rem add sql tuning Set options
insert into STMT_AUDIT_OPTION_MAP
       values (272, 'ADMINISTER SQL TUNING SET', 0);
insert into STMT_AUDIT_OPTION_MAP
       values (273, 'ADMINISTER ANY SQL TUNING SET', 0);

Rem add secure client_id privilege
insert into STMT_AUDIT_OPTION_MAP values (275, 'EXEMPT IDENTITY POLICY', 0);

Rem=====================================================================
Rem Drop views removed from last release here
Rem remove obsolete dependencies for any fixed views in i1001000.sql
Rem=====================================================================
```

```
Rem============================================================================
Rem Drop packages removed from last release here
Rem============================================================================


Rem============================================================================
Rem Add changes to sql.bsq dictionary tables here
Rem============================================================================

Rem Add SQL Tuning Base
create table sql$                         /* base table for SQL Tuning Base */
(
  signature    number  not null,         /* signature of normalized SQL text */
  nhash number         not null,           /* hash value for normalized text */
  sqlarea_hash number  not null,                      /* sql cache hash value */
  last_used    date    not null,                          /* week of last use */
  inuse_features number not null, /* bit map of features used by this object */
                             /* 0x01 - SQLProfiles, 0x02 - stored outlines */
  flags        number  not null,                       /* not used currently */
  modified     date    not null,              /* last modification timestamp */
  incarnation  number  not null,          /* modification incarnation number */
  spare1       number,                                      /* spare column */
  spare2       varchar2(1000)                               /* spare column */
)
/
create unique index i_sql$signature on sql$(signature)
/
create table sql$text  /* holds SQL text for sql$ entries */
(
  signature    number  not null,         /* signature of normalized SQL text */
  sql_text     CLOB    not null,                   /* un-normalized SQL text */
  sql_len      number  not null                        /* length of SQL text */
)
/
create index i_sql$text on sql$text(signature)
/
create table sqlprof$          /* base table for storing SQL profile objects */
(
  sp_name     varchar2(30)       not null,   /* name (potentially generated) */
  signature   number             not null,/* signature of normalized SQL txt */
  category    varchar2(30)       not null,                  /* category name */
  nhash       number             not null, /* hash value for normalized text */
  created     date               not null,                  /* creation date */
  last_modified date             not null,             /* last modified date */
  type        number             not null,                /* '1' for manual, */
                                                       /* '2' for auto-tune */
  status      number             not null,              /* '1' for enabled, */
                                          /* '2' for disabled, '3' for void */
  flags       number             not null,                       /* not used */
  spare1      number,                                      /* spare column */
  spare2      varchar2(1000)                               /* spare column */
)
/
create unique index i_sqlprof$ on sqlprof$(signature, category)
/
create unique index i_sqlprof$name on sqlprof$(sp_name)
/
create table sqlprof$desc                  /* descriptions for SQL profiles */
```

```
(
  signature   number             not null,/* signature of normalized SQL txt */
  category    varchar2(30)       not null,        /* join key: category name */
  description varchar2(500)   /* profile description (potentially generated) */
)
/
create unique index i_sqlprof$desc on sqlprof$desc(signature, category)
/
create table sqlprof$attr    /* table containing attributes for SQL profiles */
(
  signature   number             not null,/* signature of normalized SQL txt */
  category    varchar2(30)       not null,        /* join key: category name */
  attr#       number             not null,     /* attr number within profile */
  attr_val    varchar2(500)      not null                 /* attribute value */
)
/
create unique index i_sqlprof$attr on sqlprof$attr
 (signature, category, attr#)
/

rem table to monitor lifetime caching statistics

create table cache_stats_1$ (
dataobj# number not null,
inst_id number not null,
cached_avg number,
cached_sqr_avg number,
cached_no integer,
cached_seq_no integer,
chr_avg number,
chr_sqr_avg number,
chr_no integer,
chr_seq_no integer,
lgr_sum number,
lgr_last number,
phr_last number,
spare1 number,
spare2 number,
spare3 number,
spare4 number,
spare5 number
)
  storage (maxextents unlimited)
/

create index i_cache_stats_1 on cache_stats_1$(dataobj#, inst_id)
  storage (maxextents unlimited)
/

create sequence cache_stats_seq_1 start with 1 increment by 1
/

rem table to monitor workload caching statistics

create table cache_stats_0$ (
dataobj# number not null,
inst_id number not null,
cached_avg number,
```

```
cached_sqr_avg number,
cached_no integer,
cached_seq_no integer,
chr_avg number,
chr_sqr_avg number,
chr_no integer,
chr_seq_no integer,
lgr_sum number,
lgr_last number,
phr_last number,
spare1 number,
spare2 number,
spare3 number,
spare4 number,
spare5 number
)
  storage (maxextents unlimited)
/

create index i_cache_stats_0 on cache_stats_0$(dataobj#, inst_id)
  storage (maxextents unlimited)
/

create sequence cache_stats_seq_0 start with 1 increment by 1
/

/* target list for automated stats collection */
create table stats_target$ (
  staleness number not null,
         /* -100 = no stats, -1.0 ... +1.0 = staleness factor on a log scale */
  osize number not null,                    /* roughly calculated object size */
  obj#  number not null,                                /* target object obj# */
  type# number not null,                    /* target object type# as in obj$ */
  flags number not null, /* 0x0001 = failed with timeout last time           */
                         /* 0x0002 = non-segment level of partitioned object */
  status number not null,
        /* 0 = pending, 1 = gathering in progress, 2 = completed, 3 = failed */
  sid     number, /* session id of the session working/worked on this object */
  serial# number,    /* serial# of the session working/worked on this object */
  part#  number,                   /* [sub]partition# if applicable else null */
  bo# number                                           /* base or parent obj# */
   /* table partition: obj# of the parent table                               */
   /* table subpartition: obj# of the parent table partition                  */
   /* non-partitioned or global index: obj# of the base table                 */
   /* local index partition: obj# of the corresponding table partition        */
   /* local index subpartition: obj# of the corresponding table subpartition */
   /* else: null                                                              */
)
  storage (maxextents unlimited)
  tablespace sysaux;
create index i_stats_target1 on stats_target$ (staleness, osize, obj#, status)
  storage (maxextents unlimited)
  tablespace sysaux;
create unique index i_stats_target2 on stats_target$ (obj#)
  storage (maxextents unlimited)
  tablespace sysaux;

/* alter storage parameters for some existing objects */
```

```
begin
  execute immediate
     'alter cluster c_obj#_intcol#
        storage (next 200k maxextents unlimited pctincrease 0)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

begin
  execute immediate
     'alter index i_obj#_intcol# storage (maxextents unlimited)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

begin
  execute immediate
     'alter table hist_head$
        storage (next 100k maxextents unlimited pctincrease 0)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

begin
  execute immediate
    'alter index i_hh_obj#_col# storage (maxextents unlimited)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

begin
  execute immediate
    'alter index i_hh_obj#_intcol# storage (maxextents unlimited)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

begin
  execute immediate
    'alter table mon_mods$
       storage (next 100k maxextents unlimited pctincrease 0)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/
```

```
begin
  execute immediate
    'alter index i_mon_mods$_obj storage (maxextents unlimited)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

begin
  execute immediate
    'alter table col_usage$
      storage (next 100k maxextents unlimited pctincrease 0)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

begin
  execute immediate
    'alter index i_col_usage$ storage (maxextents unlimited)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

begin
  execute immediate
    'alter table object_usage storage (maxextents unlimited)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

begin
  execute immediate
    'alter index i_stats_obj# storage (maxextents unlimited)';
exception
  when others then
    if (sqlcode = -25150) then null; else raise; end if;
end;
/

Rem=========================================================================
Rem Plan Stability changes
Rem=========================================================================

alter table outln.ol$nodes add node_name varchar2(64);

Rem c1001000.sql contributions END here. tbgraves

Rem=========================================================================
Rem  Add changes to SYSTEM objects here
Rem=========================================================================
```

```
Rem ========================================================================
Rem Upgrade system types to 10.2.0
Rem ========================================================================

CREATE OR REPLACE LIBRARY UPGRADE_LIB TRUSTED AS STATIC
/

CREATE OR REPLACE PROCEDURE upgrade_system_types_from_920 IS
LANGUAGE C
NAME "UPG_FROM_920"
LIBRARY UPGRADE_LIB;
/

DECLARE
x_null CHAR(1);
BEGIN
   SELECT NULL INTO x_null
   from obj$ o, user$ u
   where o.name in  ('BINARY_DOUBLE', 'BINARY_FLOAT') and
     o.owner#=u.user# and u.name='SYS' and o.type#=13
     and rownum<=1;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
   upgrade_system_types_from_920();
END;
/

drop procedure upgrade_system_types_from_920;

Rem Drop these types so that they will be recreated (no longer evolved)
DROP TYPE ODCIColInfo FORCE;
DROP TYPE ODCIIndexInfo FORCE;
DROP TYPE ODCICost FORCE;
DROP TYPE ODCIArgDesc FORCE;
DROP TYPE ODCIEnv FORCE;

-- Upgrade xmlgenformattype

-- drop xmlgenformattype.createformat() static function
begin
  execute immediate 'alter type sys.xmlgenformattype drop static function createFormat(enclTag IN varchar2, schemaType IN
varchar2, schemaName IN varchar2, targetNameSpace IN varchar2, dburlPrefix IN varchar2, processingIns IN varchar2) RETURN
XMLGenFormatType cascade';
exception
  when others then
     if sqlcode = -22324 then null;
     else raise;
     end if;
end;
/


alter type sys.xmlgenformattype add static function createFormat(
      enclTag IN varchar2 := 'ROWSET',
      schemaType IN varchar2 := 'NO_SCHEMA',
      schemaName IN varchar2 := null,
      targetNameSpace IN varchar2 := null,
```

```
      dburlPrefix IN varchar2 := null,
      processingIns IN varchar2 := null) RETURN XMLGenFormatType
        deterministic parallel_enable cascade;

begin
  execute immediate 'alter type sys.xmlgenformattype drop  CONSTRUCTOR FUNCTION XMLGenFormatType (enclTag IN varchar2 :=
''ROWSET'', schemaType IN varchar2 := ''NO_SCHEMA'', schemaName IN varchar2 := null, targetNameSpace IN varchar2 := null,
dbUrlPrefix IN varchar2 := null, processingIns IN varchar2 := null) RETURN SELF AS RESULT cascade';
exception
  when others then
     if sqlcode = -22324 then null;
     else raise;
     end if;
end;
/

alter type sys.xmlgenformattype add CONSTRUCTOR FUNCTION XMLGenFormatType (
      enclTag IN varchar2 := 'ROWSET',
      schemaType IN varchar2 := 'NO_SCHEMA',
      schemaName IN varchar2 := null,
      targetNameSpace IN varchar2 := null,
      dbUrlPrefix IN varchar2 := null,
      processingIns IN varchar2 := null) RETURN SELF AS RESULT
       deterministic parallel_enable cascade;

alter type sys.xmlgenformattype add STATIC function createFormat2(
       enclTag in varchar2 := 'ROWSET',
       flags in raw) return sys.xmlgenformattype
       deterministic parallel_enable cascade;

alter type sys.xmlgenformattype add attribute controlflag raw(4) cascade;

Rem ========================================================================
Rem  All additions/modifications to lcr$_row_XXX must go here.
Rem ========================================================================

Rem Workaround for bug 2897618
Rem Drop methods from lcr$_row_record before lcr$_row_unit type evolution and
Rem add them back after the type has evolved
Rem These methods are:
Rem 1. lcr$_row_record.construct  : added in 9201
Rem 2. lcr$_row_record.set_values : added in 9201
Rem 3. lcr$_row_record.get_values : added in 9202 : w/ row_list in signature

ALTER TYPE lcr$_row_record DROP STATIC FUNCTION construct(
     source_database_name       in varchar2,
     command_type               in varchar2,
     object_owner               in varchar2,
     object_name                in varchar2,
     tag                        in raw               DEFAULT NULL,
     transaction_id             in varchar2          DEFAULT NULL,
     scn                        in number            DEFAULT NULL,
     old_values                 in sys.lcr$_row_list DEFAULT NULL,
     new_values                 in sys.lcr$_row_list DEFAULT NULL
   )  RETURN lcr$_row_record CASCADE;

ALTER TYPE lcr$_row_record DROP MEMBER procedure set_values(
  self in out nocopy lcr$_row_record,
```

```
  value_type          IN VARCHAR2,
  value_list          IN sys.lcr$_row_list) CASCADE;

-- Remove lcr$_row_record methods that were introduced between 9.2.0.2 and
-- higher releases and refer to lcr$_row_list
DECLARE
   version       varchar2(30);
   alt_typ_stmt  varchar2(500);
BEGIN
  EXECUTE IMMEDIATE
    'SELECT substr(dbms_registry.version(''CATPROC''),1,7) FROM DUAL'
      INTO version;

  -- drop these methods only for 9.2.0.2 and higher releases
  IF substr(version, 1, 3) = '9.2' AND
     version != '9.2.0.1' THEN
    alt_typ_stmt :=
      'ALTER TYPE lcr$_row_record DROP MEMBER FUNCTION get_values( ' ||
      '  value_type          IN VARCHAR2, ' ||
      '  use_old             IN VARCHAR2  DEFAULT ''Y'') ' ||
      '  return sys.lcr$_row_list CASCADE';
    EXECUTE IMMEDIATE alt_typ_stmt;
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    IF sqlcode = -904 THEN NULL; /* dbms_registry package does not
                                  * exist, pre-92 db
                                  */
    ELSE RAISE;
    END IF;
END;
/

Rem Evolve type lcr$_row_unit

ALTER TYPE lcr$_row_unit ADD ATTRIBUTE long_information NUMBER CASCADE;

ALTER TYPE lcr$_row_unit ADD CONSTRUCTOR FUNCTION lcr$_row_unit(
    column_name        VARCHAR2,
    data               SYS.ANYDATA,
    lob_information    NUMBER,
    lob_offset         NUMBER,
    lob_operation_size NUMBER)
    RETURN SELF AS RESULT CASCADE;

Rem Now add those methods back to lcr$_row_record

ALTER TYPE lcr$_row_record ADD STATIC FUNCTION construct(
     source_database_name       in varchar2,
     command_type               in varchar2,
     object_owner               in varchar2,
     object_name                in varchar2,
     tag                        in raw               DEFAULT NULL,
     transaction_id             in varchar2          DEFAULT NULL,
     scn                        in number            DEFAULT NULL,
     old_values                 in sys.lcr$_row_list DEFAULT NULL,
     new_values                 in sys.lcr$_row_list DEFAULT NULL
   )  RETURN lcr$_row_record CASCADE;
```

```
ALTER TYPE lcr$_row_record ADD MEMBER FUNCTION get_values(
        value_type          IN VARCHAR2,
        use_old             IN VARCHAR2  DEFAULT 'Y')
        return sys.lcr$_row_list CASCADE;

ALTER TYPE lcr$_row_record ADD MEMBER procedure set_values(
        self in out nocopy lcr$_row_record,
        value_type          IN VARCHAR2,
        value_list          IN sys.lcr$_row_list) CASCADE;

Rem Now add the new methods added in 10.1

ALTER TYPE lcr$_row_record ADD MEMBER FUNCTION
   get_extra_attribute(
        attribute_name         IN VARCHAR2) RETURN Sys.AnyData CASCADE;

ALTER TYPE lcr$_row_record ADD MEMBER PROCEDURE
    set_extra_attribute(self in out nocopy lcr$_row_record,
        attribute_name         IN VARCHAR2,
        attribute_value        IN Sys.AnyData) CASCADE;

ALTER TYPE lcr$_row_record ADD MEMBER FUNCTION
   get_compatible RETURN NUMBER CASCADE;

ALTER TYPE lcr$_row_record ADD MEMBER FUNCTION
   get_long_information(
        value_type             IN VARCHAR2,
        column_name            IN VARCHAR2,
        use_old                IN VARCHAR2  DEFAULT 'Y') RETURN NUMBER CASCADE;

ALTER TYPE lcr$_row_record ADD MEMBER PROCEDURE
   convert_long_to_lob_chunk(
        self in out nocopy lcr$_row_record) CASCADE;

ALTER TYPE lcr$_ddl_record ADD MEMBER FUNCTION
   get_extra_attribute(
        attribute_name         IN VARCHAR2) RETURN Sys.AnyData CASCADE;

ALTER TYPE lcr$_ddl_record ADD MEMBER PROCEDURE
    set_extra_attribute(self in out nocopy lcr$_ddl_record,
        attribute_name         IN VARCHAR2,
        attribute_value        IN Sys.AnyData) CASCADE;

ALTER TYPE lcr$_ddl_record ADD MEMBER FUNCTION
   get_compatible RETURN NUMBER CASCADE;


Rem=========================================================================
Rem Supplemental log related metadata fixups go here
Rem=========================================================================

-- for large databases, limit the number of rows changed before commit to
-- avoid rollback problems during upgrade
begin
  loop
      execute immediate
        'update sys.ccol$ set spare1 = 0
```

```
               where spare1 IS NULL and
                     rownum <10000';
      exit when sql%rowcount = 0;
      commit;
   end loop;
end;
/
commit;

REM ========================================================================
REM BEGIN Drop Time Series
REM ========================================================================

delete from sys.exppkgact$
  where package = 'ORDTEXP' and
        schema  = 'ORDSYS';
commit;

drop public synonym DBA_TIMESERIES_COLS;
drop public synonym DBA_TIMESERIES_OBJS;
drop public synonym DBA_TIMESERIES_GROUPS;
drop public synonym ALL_TIMESERIES_GROUPS;
drop public synonym ALL_TIMESERIES_COLS;
drop public synonym ALL_TIMESERIES_OBJS;
drop public synonym USER_TIMESERIES_OBJS;
drop public synonym USER_TIMESERIES_GROUPS;
drop public synonym USER_TIMESERIES_COLS;

declare
 cnt number := 0;
begin
 select count(1) into cnt
   from user$ where name = 'ORDSYS';

 if (cnt = 0) then
   return;
 else

   select count(1) into cnt
     from obj$
    where name = 'ORD_INSTALLATIONS'
      and type# = 2
      and owner# = (select user# from user$ where name = 'ORDSYS');

   if (cnt != 0) then
     execute immediate 'delete from ORDSYS.ORD_INSTALLATIONS ' ||
                       ' where short_name=''ORDTS'' ';
     commit;
   end if;

   execute immediate 'drop view ORDSYS.DBA_TIMESERIES_GROUPS';
   execute immediate 'drop view ORDSYS.ALL_TIMESERIES_GROUPS';
   execute immediate 'drop view ORDSYS.USER_TIMESERIES_GROUPS';
   execute immediate 'drop view ORDSYS.DBA_TIMESERIES_OBJS';
   execute immediate 'drop view ORDSYS.ALL_TIMESERIES_OBJS';
   execute immediate 'drop view ORDSYS.USER_TIMESERIES_OBJS';
   execute immediate 'drop view ORDSYS.DBA_TIMESERIES_COLS';
   execute immediate 'drop view ORDSYS.ALL_TIMESERIES_COLS';
```

```
execute immediate 'drop view ORDSYS.USER_TIMESERIES_COLS';

execute immediate 'drop package  ORDSYS.TIMESERIES';
execute immediate 'drop package  ORDSYS.TIMESCALE';
execute immediate 'drop package  ORDSYS.TSTOOLS';
execute immediate 'drop package  ORDSYS.CALENDAR';
execute immediate 'drop package  ORDSYS.ORDTMATH';
execute immediate 'drop package  ORDSYS.ORDTMOVE';
execute immediate 'drop package  ORDSYS.ORDTCUME';
execute immediate 'drop package  ORDSYS.ORDTTRANS';
execute immediate 'drop package  ORDSYS.ORDTSCALE';
execute immediate 'drop package  ORDSYS.ORDTGET';
execute immediate 'drop package  ORDSYS.ORDTAGG';
execute immediate 'drop package  ORDSYS.ORDTDDL';
execute immediate 'drop package  ORDSYS.ORDTEXP';
execute immediate 'drop package  ORDSYS.ORDTCUTL';
execute immediate 'drop package  ORDSYS.ORDTTUTL';
execute immediate 'drop package  ORDSYS.ORDTSYS';
execute immediate 'drop package  ORDSYS.ORDTUTL';
execute immediate 'drop package  ORDSYS.ORDTTUTL2';

execute immediate 'drop library ORDSYS.ORDTSLIBT';

declare
  type obj_cur_typ is ref cursor;
  obj_cur obj_cur_typ;
  obj_owner varchar2(50);
  obj_name varchar2(50);
  obj_type varchar2(50);
  cnt number;
  no_such_table exception;
  pragma exception_init(no_such_table, -942);
begin
  open obj_cur for
    'select tso.owner,  tso.obj_name, tso.obj_type ' ||
    '  from ORDSYS.ORDT_TIMESERIES_OBJS tso ' ||
    '  where tso.obj_type in ( ''VIEW'', ''TRIGGER'') ';
  loop
    fetch obj_cur into obj_owner, obj_name, obj_type;
    exit when obj_cur%NOTFOUND;
    select count(1) into cnt from obj$ o, user$ u
     where u.user# = o.owner# and o.type# in (4,12)
       and o.name = obj_name and u.name = obj_owner;
    if (cnt = 1) then
      execute immediate 'drop ' || obj_type ||' '
                        || obj_owner || '.' || obj_name;
    end if;
  end loop;
exception when no_such_table then
  null;
end;

execute immediate 'drop TABLE ORDSYS.ORDT_TIMESERIES PURGE';
execute immediate 'drop TABLE ORDSYS.ORDT_TIMESERIES_OBJS PURGE';
execute immediate 'drop TABLE ORDSYS.ORDT_FLAT_ATTRIBUTES PURGE';
execute immediate 'drop TABLE ORDSYS.ORDT_OBJECT_ATTRIBUTES PURGE';
execute immediate 'drop TABLE ORDSYS.ORDT_TIMESERIES_COLS PURGE';
```

```
   execute immediate 'drop type ORDSYS.ORDTDateTab';
   execute immediate 'drop type ORDSYS.ORDTDateRangeTab';
   execute immediate 'drop type ORDSYS.ORDTDateRange';
   execute immediate 'drop type ORDSYS.ORDTNumSeriesIOTRef';
   execute immediate 'drop type ORDSYS.ORDTVarchar2SeriesIOTRef';
   execute immediate 'drop type ORDSYS.ORDTNumSeries';
   execute immediate 'drop type ORDSYS.ORDTVarchar2Series';
   execute immediate 'drop type ORDSYS.ORDTNumTab';
   execute immediate 'drop type ORDSYS.ORDTVarchar2Tab';
   execute immediate 'drop type ORDSYS.ORDTNumCell';
   execute immediate 'drop type ORDSYS.ORDTVarchar2Cell';
   execute immediate 'drop type ORDSYS.ORDTCalendar';
   execute immediate 'drop type ORDSYS.ORDTExceptions';
   execute immediate 'drop type ORDSYS.ORDTPattern';
   execute immediate 'drop type ORDSYS.ORDTPatternBits';
 end if;
end;
/

REM ========================================================================
REM END Drop Time Series
REM ========================================================================

Rem ============================== Start of MV upgrade ==================

Rem ========================================================================
Rem  Fix bug #3320404: Delete the fast refresh operations for LOB MVs
Rem  as the refresh operations are different in 10g.
Rem ========================================================================

Rem  Set status of LOB MVs to regenerate refresh operations
UPDATE sys.snap$ s SET s.status = 0
 WHERE bitand(s.flag, 512) = 512 AND s.instsite = 0 ;

Rem  Delete old fast refresh operations for LOB MVs
DELETE FROM sys.snap_refop$ sr
 WHERE EXISTS
  ( SELECT 1 from sys.snap$ s
     WHERE bitand(s.flag, 512) = 512 AND s.instsite = 0
            AND sr.sowner = s.sowner
            AND sr.vname = s.vname ) ;
COMMIT;

Rem ============================== End of MV upgrade ==================

Rem ------------------------------------------------------------------------
Rem If this is a little endian machine with varying width LOB,
Rem then set a flag in LOB$ saying that this LOB columns stores data
Rem in AL16UTF16LE. This needs to be done before any
Rem inserts or selects are done in any LOB column during upgrade
Rem (e.g., AQ rules)
Rem Create function platform_little_endian here as a trusted callout
Rem so the lob$ block will not depend on any dbms* package being loaded.
Rem ------------------------------------------------------------------------

CREATE OR REPLACE LIBRARY UPGRADE_LIB TRUSTED AS STATIC
/
```

```
CREATE OR REPLACE FUNCTION platform_little_endian return boolean IS
LANGUAGE C
NAME "IS_LITTLE_ENDIAN"
LIBRARY UPGRADE_LIB;
/

declare
begin
  if (platform_little_endian = TRUE) then
    update lob$ l set property=property+512 where bitand(property, 512)=0
      and exists
      (select c.obj# from col$ c where l.obj# = c.obj# and l.intcol#=c.intcol#
       and c.type# = 112 and ((c.charsetid > 800 AND c.charsetid < 1000) OR
                              c.charsetid > 2000));
  end if;
end;
/
drop function platform_little_endian;

Rem=========================================================================
Rem END STAGE 1: upgrade from 9.2.0 to 10.1
Rem=========================================================================

Rem=========================================================================
Rem BEGIN STAGE 2: invoke script for subsequent release
Rem=========================================================================

@@c1001000

Rem=========================================================================
Rem END STAGE 2: invoke script for subsequent release
Rem=========================================================================

Rem*************************************************************************
Rem END c0902000.sql
Rem*************************************************************************
```