oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（27）c0900010.sql

oracle 10g 研究ORACLE_HOME rdbms admin 下的脚本的功能（27）c0900010.sql

```
#upgrade Oracle RDBMS from 9.0.1 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 a0900010.sql
Rem      as catalog.sql and catproc.sql will be run before a0900010.sql
Rem      is invoked.
Rem
Rem      This script is called from u0900010.sql and c0801070.sql
Rem
Rem      This script performs the upgrade in the following stages:
Rem        STAGE 1: upgrade from 9.0.1 to new release
Rem        STAGE 2: call catalog.sql and catproc.sql
#

Rem
Rem $Header: c0900010.sql 02-sep-2004.08:17:07 rburns Exp $
Rem
Rem c0900010.sql
Rem
Rem Copyright (c) 1999, 2004, Oracle. All rights reserved.
Rem
Rem    NAME
Rem      c0900010.sql - upgrade Oracle RDBMS from 9.0.1 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 a0900010.sql
Rem      as catalog.sql and catproc.sql will be run before a0900010.sql
Rem      is invoked.
Rem
Rem      This script is called from u0900010.sql and c0801070.sql
Rem
Rem      This script performs the upgrade in the following stages:
Rem        STAGE 1: upgrade from 9.0.1 to new release
Rem        STAGE 2: call catalog.sql and catproc.sql
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    rburns      09/02/04 - remove serveroutput
Rem    rburns      07/15/04 - remove dbms_output compiles
Rem    arithikr    03/25/04 - 3473968 - correct mispell privilege
Rem    nireland    01/05/04 - Fix ts# for indices on temp tables. #3238525
Rem    ssubrama    06/10/02 - bug 2385207 move delete from dependency$
Rem    ssubrama    06/02/02 - bug 2385207 delete dependency for no-existant
Rem                           x$ tables from dependency$
Rem    rburns      02/13/02 - call 9.2.0 script
Rem    nbhatt      02/08/02 - transformation upgrade changes
Rem    araghava    01/23/02 - remove unnecessary indexes on partitioning
Rem                           metadata.
```

```
Rem    cmlim       01/15/02 - bug 2093119 (remove x$kllcnt & x$klltab)
Rem    nbhatt      01/23/02 - add columns to the transformation$ table
Rem    rburns      01/09/02 - reset serveroutput
Rem    rburns      12/18/01 - add drops of HS and logstdby objects
Rem    wojeil      12/05/01 - upgrade script for map_object.
Rem    wesmith     11/19/01 - add additional columns to Streams tables
Rem    ayoaz       11/15/01 - change upg func names for 9.2.0
Rem    rburns      11/16/01 - fix map_extelement table
Rem    avaliani    11/16/01 - rm change NULL to UB4MAXVAL
Rem    weiwang     11/13/01 - change index i_objtype to unique index
Rem    rherwadk    11/09/01 - #1817695: unlimit default resmgr parameter values
Rem    rburns      11/05/01 - cleanup
Rem    weiwang     11/05/01 - rules engine upgrade
Rem    wesmith     11/02/01 - Streams upgrade
Rem    kmeiyyap    11/02/01 - add streams$_propagation_process
Rem    najain      11/01/01 - add aq$_replay_info
Rem    kmeiyyap    11/02/01 - add streams$_propagation_process
Rem    najain      11/01/01 - add aq$_replay_info
Rem    lvbcheng    11/05/01 - action line no offset
Rem    celsbern    10/25/01 - merging LOG into MAIN
Rem    rburns      10/26/01 - wrap drop index
Rem    vshukla     11/01/01 - set KQLDTVNTF_HAS_MPR bit in tab$ if
Rem                           KQLDTVCM_MPR bit is set.
Rem    rburns      10/24/01 - fix i_rls statement
Rem    clei        10/02/01 - add synonym id to rls_grp$ and rls_ctx$
Rem    dmwong      10/13/01 - clob lsqltext for fga_log$
Rem    jcarey      09/24/01 - more aw$ and ps$
Rem    wojeil      10/30/01 - modified map_extelement table.
Rem    clei        09/15/01 - re-create i_rls as non unique index
Rem    esoyleme    09/10/01 - AW$ and PS$.
Rem    akalra      08/31/01 - add flashback privilege.
Rem    sbasu       08/24/01 - Add hiboundlen, hiboundval, bhiboundval to
Rem                           [tab|ind]subpart$
Rem    rburns      08/23/01 - move attribute synonym columns to i script
Rem    ayoaz       08/22/01 - upgrade type system to 9.2.0
Rem    ayoaz       08/17/01 - Add synobj# column to coltype$.
Rem    ayoaz       08/16/01 - Add synobj# column to attr$,coll$,res$,param$
Rem    eyho        07/02/01 - drop old ext_to_obj view and table
Rem    dcwang      07/16/01 - add new privilege: grant any object privilege.
Rem    mxiao       07/02/01 - update audit options
Rem    dmwong      06/19/01 - add the missing delete entry on fga_log$.
Rem    lbarton     06/11/01 - add indexes on lob$(lob#), lobcomppart$(partobj#)
Rem    rburns      06/07/01 - Merged rburns_setup_901_upgrade
Rem    rburns      06/04/01 - Created

Rem=========================================================================
Rem BEGIN STAGE 1: upgrade from 9.0.1 to 9.2.0
Rem=========================================================================

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

insert into SYSTEM_PRIVILEGE_MAP values (-243, 'FLASHBACK ANY TABLE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-244, 'GRANT ANY OBJECT PRIVILEGE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-64, 'CREATE RULE', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-65, 'CREATE ANY RULE', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-66, 'ALTER ANY RULE', 1);
```

```
insert into SYSTEM_PRIVILEGE_MAP values (-67, 'DROP ANY RULE', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-68, 'EXECUTE ANY RULE', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-245, 'CREATE EVALUATION CONTEXT', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-246, 'CREATE ANY EVALUATION CONTEXT',
                                         1);
insert into SYSTEM_PRIVILEGE_MAP values (-247, 'ALTER ANY EVALUATION CONTEXT',
                                         1);
insert into SYSTEM_PRIVILEGE_MAP values (-248, 'DROP ANY EVALUATION CONTEXT',
                                         1);
insert into SYSTEM_PRIVILEGE_MAP values (-249,
                                         'EXECUTE ANY EVALUATION CONTEXT', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-250, 'CREATE RULE SET', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-251, 'CREATE ANY RULE SET', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-252, 'ALTER ANY RULE SET', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-253, 'DROP ANY RULE SET', 1);
insert into SYSTEM_PRIVILEGE_MAP values (-254, 'EXECUTE ANY RULE SET', 1);
grant all privileges to dba with admin option;

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

grant delete on fga_log$ to delete_catalog_role;

insert into TABLE_PRIVILEGE_MAP values (27, 'FLASHBACK');

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

Rem update STMT_AUDIT_OPTION_MAP created in sql.bsq
update STMT_AUDIT_OPTION_MAP set NAME = 'MATERIALIZED VIEW'
  where OPTION# = 39;

insert into STMT_AUDIT_OPTION_MAP values (243, 'FLASHBACK ANY TABLE', 0);
insert into STMT_AUDIT_OPTION_MAP values (244, 'GRANT ANY OBJECT PRIVILEGE', 0);

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

Rem These views were missed if the database had been upgraded to 9.0.1.0.0 or
Rem 9.0.1.1.0
drop view HS_EXTERNAL_OBJECTS;
drop public synonym HS_EXTERNAL_OBJECTS;
drop view HS_EXTERNAL_OBJECT_PRIVILEGES;
drop public synonym HS_EXTERNAL_OBJECT_PRIVILEGES;
drop view HS_EXTERNAL_USER_PRIVILEGES;
drop public synonym HS_EXTERNAL_USER_PRIVILEGES;

drop view V_$LOGSTDBY_APPLY;
drop public synonym V$LOGSTDBY_APPLY;
drop view GV_$LOGSTDBY_APPLY;
drop public synonym GV$LOGSTDBY_APPLY;
drop view V_$LOGSTDBY_COORDINATOR;
drop public synonym V$LOGSTDBY_COORDINATOR;
drop view GV_$LOGSTDBY_COORDINATOR;
drop public synonym GV$LOGSTDBY_COORDINATOR;
```

```
Rem remove obsolete fixed view information
delete from dependency$ where d_obj# in (select obj# from obj$ where name in
  ('V_$LOADCSTAT', 'GV_$LOADCSTAT', 'V_$LOADTSTAT', 'GV_$LOADTSTAT'));
commit;
alter system flush shared_pool;

drop view V_$LOADCSTAT;
drop public synonym V$LOADCSTAT;

drop view GV_$LOADCSTAT;
drop public synonym GV$LOADCSTAT;

drop view V_$LOADTSTAT;
drop public synonym V$LOADTSTAT;

drop view GV_$LOADTSTAT;
drop public synonym GV$LOADTSTAT;
Rem

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

Rem This package was missed if the database had been upgraded to 9.0.1.0.0 or
Rem 9.0.1.1.0
drop package DBMS_HS_EXTPROC;
drop package UTL_LOGSTDBY;

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

REM add instance# to sumdep$
ALTER TABLE sys.sumdep$
ADD
(
    instance#    number                          /* inline view instance # */
)
/

CREATE TABLE sys.aq$_replay_info(
 eventid   NUMBER NOT NULL, -- queue id used as event id
        agent           sys.aq$_agent NOT NULL, -- sender agent
        correlationid   varchar2(128)           -- correlation id.
)
/

rem
rem Streams tables
rem
rem NOTE: the following tables are created in i0900010.sql:
rem   - streams$_prepare_object
rem   - streams$_prepare_ddl
rem   - apply$_source_obj
rem   - apply$_source_schema
rem
```

```
create table streams$_capture_process
(
  queue_oid       raw(16)             not null,       /* AQ queue identifier */
  queue_owner     varchar2(30      )  not null,            /* AQ queue owner */
  queue_name      varchar2(30      )  not null,             /* AQ queue name */
  capture#        number              not null,                   /* 1 to 99 */
  capture_name    varchar2(30      )  not null,
  status          number,      /* capture process status: START, STOP, ABORT */
  ruleset_owner   varchar2(30      ),                      /* rule set owner */
  ruleset_name    varchar2(30      ),                       /* rule set name */
  logmnr_sid      number,           /* id of the persistent logminer session */
                                 /* needed for creating a persistent session */
  predumpscn      number,            /* scn before dictionary dump was taken */
  dumpseqbeg      number,            /* first log containing dictionary dump */
  dumpseqend      number,            /* last  log containing dictionary dump */
  postdumpscn     number,         /* scn after dictionary dump was processed */
  flags           number,
  start_scn       number,
  capture_userid  number,                         /* capture security context */
  spare1          number,           /* used for last_enqueued_message_number */
  spare2          number,
  spare3          number
)
/
create unique index i_streams_capture_process1 on streams$_capture_process
 (capture#)
/
create unique index i_streams_capture_process2 on streams$_capture_process
 (capture_name)
/

create sequence streams$_capture_inst        /* capture instantiation number */
  increment by 1
  start with 1
  minvalue 1
  maxvalue 4294967295                           /* max portable value of UB4 */
  cycle
  nocache
/

create table streams$_apply_process
(
  apply#          number           not null, /* apply#0 is reserved for HaDB */
  apply_name      varchar2(30      ) not null,          /* apply process name */
  queue_oid       raw(16)            not null,        /* AQ queue identifier */
  queue_owner     varchar2(30      ) not null,             /* AQ queue owner */
  queue_name      varchar2(30      ) not null,              /* AQ queue name */
  status          number,        /* apply process status: START, STOP, ABORT */
  flags           number,                                /* apply process flags */
  ruleset_owner   varchar2(30      ),                      /* rule set owner */
  ruleset_name    varchar2(30      ),                       /* rule set name */
  message_handler varchar2(92),                           /* message handler */
  ddl_handler     varchar2(92),                              /* DDL handler */
  apply_userid    number,                          /* apply security context */
  apply_dblink    varchar2(128     ),                /* apply database link */
  apply_tag       raw(2000    ),                                 /* apply tag */
  spare1          number,
  spare2          number,
```

```
  spare3          number
)
/
create unique index i_streams_apply_process1 on
  streams$_apply_process (apply#)
/
create unique index i_streams_apply_process2 on
  streams$_apply_process (apply_name)
/
create index i_streams_apply_process3 on
  streams$_apply_process (queue_oid)
/

create table streams$_propagation_process
(
  propagation_name           varchar2(30      ) not null,
  source_queue_schema        varchar2(30      ),
  source_queue               varchar2(30      ),
  destination_queue_schema   varchar2(30      ),
  destination_queue          varchar2(30      ),
  destination_dblink         varchar2(128     ),
  ruleset_schema             varchar2(30      ),
  ruleset                    varchar2(30      ),
  spare1                     number,
  spare2                     varchar2(128     )
)
/
create unique index streams$_prop_p_i1 on streams$_propagation_process
(propagation_name)
/
create unique index streams$_prop_p_i2 on streams$_propagation_process
(source_queue_schema,source_queue, destination_queue_schema,
 destination_queue, destination_dblink)
/

rem Table to store parameters for capture and apply processes.
create table streams$_process_params
(
  process_type       number not null,               /* 1 -> apply   process */
                                                    /* 2 -> capture process */
  process#           number not null,                        /* X_process # */
  name               varchar2(128     ) not null,          /* parameter name */
  value              varchar2(4000    ),                  /* parameter value */
  user_changed_flag  number,            /* 1 if changed by user, 0 otherwise */
  internal_flag      number,    /* 1 if internal param, 0 if exposed to user */
  spare1             number
)
/
create unique index i_streams_process_params1 on
  streams$_process_params (process_type, process#, name)
/

create table streams$_apply_milestone
(
  apply#          number             not null,
  source_db_name  varchar2(128     ) not null,
  oldest_scn      number             not null,
  commit_scn      number             not null,
```

```
  synch_scn       number             not null,           /* Synch-point SCN. */
  epoch           number             not null,         /* Incarnation number */
  processed_scn   number             not null,
                            /* all complete txns < processed_scn are applied */
  apply_time      date,
  applied_message_create_time date,
  spare1          number
)
/
create unique index i_streams_apply_milestone1 on streams$_apply_milestone
(apply#, source_db_name)
/

rem No constraints on this table it has to be really high performance
rem since it is inserted on every txn
create table streams$_apply_progress
(
  apply#          number,
  source_db_name  varchar2(128     ),
  xidusn          number,
  xidslt          number,
  xidsqn          number,
  commit_scn      number,
  spare1          number
)
/

create table streams$_key_columns
(
  sname       varchar2(30      ) not null,
  oname       varchar2(30      ) not null,
  type        number             not null,
  cname       varchar2(30      ) not null,
  dblink      varchar2(128     ),
  long_cname  varchar2(4000    ),
  spare1      number
)
/
create unique index i_streams_key_columns on
  streams$_key_columns(sname, oname, type, cname, dblink)
/

rem table used for deferred prcedure calls
create table streams$_def_proc
(
  base_obj_num     number,
  flags            number,
  owner            varchar2(30      ),
  package_name     varchar2(30      ),
  procedure_name   varchar2(30      ),
  param_name       varchar2(30      ),
  param_type       number,
  raw_value        raw(2000    ),
  number_value     number,
  date_value       date,
  varchar2_value   varchar2(4000    ),
  nvarchar2_value  nvarchar2(1000),
  clob_value       clob,
```

```
  blob_value       blob,
  nclob_value      nclob
)
/

rem streams$_rules is populated by APIs in dbms_streams_adm
create table streams$_rules
(
  streams_name         varchar2(30      ),     /* capture/apply/prop process */
  streams_type         number,      /* capture (1), propagation(2), apply (3)*/
  rule_type            number,                          /* dml (1), ddl (2) */
  include_tagged_lcr   number,                                  /* 0 or 1 */
  source_database      varchar2(128     ),           /* source database name */
  rule_owner           varchar2(30      ),                    /* rule owner */
  rule_name            varchar2(30      ),     /* system generated rule name */
  rule_condition       varchar2(4000    ),  /* system generated rule context */
  dml_condition        varchar2(4000    ), /* NULL except for row subsetting */
  subsetting_operation number,    /* null, insert (1), update(2), delete (3) */
  schema_name          varchar2(30      ),  /* schema name, null for db type */
  object_name          varchar2(30      ),
                                  /* table name, null for schema/db type */
  object_type          number,       /*  table(1), schema(2),  database (3)  */
  spare1               number,
  spare2               number,
  spare3               number
)
/
create unique index i_streams_rules1 on
  streams$_rules(rule_owner, rule_name)
/
create index  i_streams_rules2 on
  streams$_rules(schema_name, object_name)
/

rem This table allows multiple objects in the destination subscribed to the
rem same source object.
create table apply$_dest_obj
(
  id              number             not null,                      /* seq # */
  source_owner    varchar2(30      ) not null,           /* source obj owner */
  source_name     varchar2(30      ) not null,            /* source obj name */
  type            number       not null,  /* type of source obj and dest obj */
  owner           varchar2(30      ) not null,             /* dest obj owner */
  name            varchar2(30      ) not null,              /* dest obj name */
  apply#          number,             /* apply process assigned to this dest */
  status          number,                  /* such as pending, ready, error */
  error_notifier  varchar2(92),            /* function to invoke for errors */
  spare1          number
)
/
create unique index i_apply_dest_obj1 on
  apply$_dest_obj (id)
/
rem source and apply# uniquely identify a destination
create unique index i_apply_dest_obj2 on
  apply$_dest_obj (source_owner, source_name, type, apply#)
/
rem destination and apply# uniquely identify a source
```

```
create unique index i_apply_dest_obj3 on
  apply$_dest_obj (owner, name, type, apply#)
/
rem sequence for apply$_dest_obj.id
create sequence apply$_dest_obj_id nocache
/

rem column mapping between source and destination tables
create table apply$_dest_obj_cmap
(
  dest_id          number             not null,       /* id of parent row in */
                                                          /* apply$_dest_obj */
  src_long_cname   varchar2(4000    ) not null,        /* source column name */
  dest_long_cname  varchar2(4000    ),            /* destination column name */
                                        /* if null, same as src_long_cname */
  spare1           number
)
/
rem we need to add src_long_cname to this index but it is > max key len
create index i_apply_dest_obj_cmap1 on
  apply$_dest_obj_cmap (dest_id)
/

rem apply operations associated with destination object
create table apply$_dest_obj_ops
(
  object_number        number not null,               /* id of parent row in */
                                                      /* obj$                */
  apply_operation      number not null,              /* apply operation type */
                                                     /* 1 -> INSERT          */
                                                     /* 2 -> UPDATE          */
                                                     /* 3 -> DELETE          */
                                                     /* 4 -> BLOB_UPDATE     */
                                                     /* 5 -> CLOB_UPDATE     */
                                                     /* 6 -> NCLOB_UPDATE    */
  error_handler        char(1),                      /* 'Y' if error handler */
                                                     /* 'N' if not           */
  user_apply_procedure varchar2(92),  /* if user_apply_procedure is null,    */
                                      /* default apply rules will be used    */
                                      /* or if there is no child row in      */
                                      /* apply$_dest_obj_ops                 */
                                      /* for apply$_dest_obj.id              */
  spare1               number,
  spare2               number,
  spare3               number
)
/
create unique index i_apply_dest_obj_ops1 on
  apply$_dest_obj_ops (object_number, apply_operation)
/

rem table used to store error transaction information
create table apply$_error
(
  local_transaction_id  varchar2(22      ),     /* Tid of error creation txn */
  source_transaction_id varchar2(22      ),  /* transaction id at the source */
  source_database       varchar2(128     ),/* node which originated this txn */
  queue_owner           varchar2(30      ) not null,    /* local queue owner */
```

```
  queue_name            varchar2(30      ) not null,     /* local queue name */
  apply#                number  not null, /* apply engine processing the txn */
  message_number        number,            /* message which caused the error */
  message_count         number,             /* Number of messages in the txn */
  min_step_no           number,            /* min step no in exception queue */
  recipient_id          number,          /* User ID of the original receiver */
  recipient_name        varchar2(30      ),
                                        /* User name of the original receiver */
  source_commit_scn     number,           /* original commit SCN for the txn */
  error_number          number,                     /* error number reported */
  error_message         varchar2(4000    ),          /* explanation of error */
  aq_transaction_id     varchar2(30),                   /* AQ transaction id */
  spare1                number,
  spare2                number,
  spare3                number
)
/
create unique index streams$_apply_error_unq
 on apply$_error(local_transaction_id)
/

rem tables required for conflict resolution
rem apply$_error_handler_sequence is used to generate a value
rem for log_group_id in apply$_error_handler.

create sequence apply$_error_handler_sequence start with 1
/

rem stores all conflict resolution methods
create table apply$_error_handler
(
  object_number          number,  /* table obj# error handler is defined for */
  method_name            varchar2(92),                     /* name of method */
  resolution_column      varchar2(4000    ), /* column used to resolve error */
  resolution_id          number,          /* id number for the error handler */
  spare1                 number
)
/
create unique index apply$_error_handler_unq
 on apply$_error_handler(resolution_id)
/

rem stores the column list for update column resolution
create table apply$_conf_hdlr_columns
(
  object_number number,           /* table obj# error handler is defined for */
  resolution_id number,                   /* id number for the error handler */
  column_name   varchar2(30      ),   /* name of a column in the column list */
                                             /* for a update conf handler */
  spare1        number
)
/
create unique index apply$_conf_hdlr_columns_unq1
 on apply$_conf_hdlr_columns(object_number, column_name)
/
create unique index apply$_conf_hdlr_columns_unq2
 on apply$_conf_hdlr_columns(resolution_id, column_name)
/
```

```
Rem
Rem End: Streams tables
Rem

REM support for aw$ and ps$
create table aw$
(awname varchar2(30),                    /* name of AW */
 owner#  number not null,                /* owner of AW */
 awseq#  number not null)                /* aw sequence number */
/

create unique index aw_ind$ on aw$(awname, owner#)
/
create table ps$
(awseq# number not null,                 /* aw sequence number */
 psnumber number(10),                    /* pagespace number */
 psgen number(4),                        /* pagespace generation */
 mapoffset number,                       /* offset of map */
 maxpages number,                        /* max pages in ps */
 almap raw(8),                           /* location of map in lob */
 header raw(200),                        /* our header */
 gelob  blob)                            /* gel storage */
 lob (gelob) store as (disable storage in row)
/
create unique index i_ps$ on ps$ (awseq#, psnumber, psgen)
/
create sequence psindex_seq$ /* sequence for pagespace index */
 start with 100
 increment by 1
 nocache
 nocycle
 maxvalue 18446744073709551615
/

create sequence awseq$ /* sequence for aw index */
  start with 1000
  increment by 1
  nocache
  nocycle
  maxvalue 4294967295
/

REM add action line offset to trigger$

ALTER TABLE sys.trigger$
ADD
(
    actionlineno number
)
/

Rem ========================================================================
Rem Update TAB$ to indicate that ROW MOVEMENT was enabled sometime in the
Rem past if ROW MOVEMENT is currently set.
Rem ========================================================================

update tab$ set trigflag = trigflag + 2097152
```

```
where  bitand(flags, 131072) = 131072 and
       bitand(trigflag, 2097152) = 0;

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 ========================================================================
Rem The following SQL stmts. add support for Range List partitioned objects
Rem add columns hiboundlen, hiboundval and bhiboundval to {tab|ind}subpart$
Rem ============ begin of Range List partitioned objects upgrade ===========
alter table tabsubpart$ add (
        hiboundlen  number,   /* length, high bound value */
        hiboundval  long,     /* text, high-bound value */
        bhiboundval blob )    /* binary linear key, high bound */
/
update tabsubpart$ set hiboundlen = 0
/
alter table tabsubpart$ modify (hiboundlen not null)
/

alter table indsubpart$ add (
        hiboundlen    number,   /* length, high bound value */
        hiboundval    long,     /* text, high-bound value */
        bhiboundval   blob )    /* binary linear key, high bound */
/
update indsubpart$ set hiboundlen = 0
/
alter table indsubpart$ modify (hiboundlen not null)
/

Rem ============================================================
Rem remove unnecessary indexes. they don't improve select
Rem performance and actually degrade partition DDL performance.
Rem create indexes on lobfrag$ and lobcomppart$.
Rem

drop index i_tabpart$_bopart$
/
drop index i_indpart$_bopart$
/
drop index i_tabsubpart$_pobjsubpart$
/
drop index i_indsubpart$_pobjsubpart$
/
drop index i_tabcompart$_bopart$
/
drop index i_indcompart$_bopart$
/
drop index i_lobfrag$_parentobjfrag$
/
```

```
drop index i_lobcomppart$_partlobj$
/
create index i_lobfrag$_parentobj$ on lobfrag$(parentobj#)
/
create index i_lobcomppart$_partlobj$ on lobcomppart$(lobj#)
/
create index i_lobfrag$_fragobj$ on lobfrag$(fragobj#)
/

Rem ============================================================
Rem resource_plan_directive$ column default value changes
Rem ============================================================
update resource_plan_directive$
set    parallel_degree_limit_p1=4294967295
where  parallel_degree_limit_p1=1000000
/
update resource_plan_directive$
set    active_sess_pool_p1=4294967295
where  active_sess_pool_p1=1000000
/
update resource_plan_directive$
set    queueing_p1=4294967295
where  queueing_p1=1000000
/
update resource_plan_directive$
set    switch_time=4294967295
where  switch_time=1000000
/
update resource_plan_directive$
set    max_est_exec_time=4294967295
where  max_est_exec_time=1000000
/
update resource_plan_directive$
set    undo_pool=4294967295
where  undo_pool=1000000
/
commit
/


Rem=========================================================================
REM Add changes to security dictionary objects here
Rem=========================================================================

REM re-create i_rls - synonym and it's base object may have the same policy name
BEGIN
  EXECUTE IMMEDIATE 'drop index i_rls';
EXCEPTION
   WHEN OTHERS THEN
      IF SQLCODE = -1418 THEN NULL;
      ELSE RAISE;
      END IF;
END;
/

create index i_rls on rls$(obj#, gname, pname);

REM add synonym id for policy group
```

```
alter table rls_grp$ add (synid number default null);

REM add synonym id for driving context
alter table rls_ctx$ add (synid number default null);

alter table fga_log$ add (
         lsqltext clob,
         plhol    long );

update fga_log$ set lsqltext = to_clob(sqltext);

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

Rem Add indexes on lob$ and lobcomppart$ to improve Metadata API performance.
create unique index i_lob2 on lob$(lobj#)
/
create index i_lobcomppart$_partobj$ on lobcomppart$(partobj#)
/

Rem remove obsoleted table and view for ext_to_obj
drop table EXT_TO_OBJ;
drop view EXT_TO_OBJ_VIEW;

Rem add data dictionary objects for file mapping

create table map_file$ (
  file_idx      number,                       /* file index */
  file_cfgid    varchar2(2000),               /* file configuration id */
  file_status   number,                       /* file status */
  file_name     varchar2(2000),               /* file name */
  file_struct   number,                       /* file structure */
  file_type     number,                       /* file type */
  file_size     number,                       /* file size */
  file_nexts    number                        /* file number of extents */
)
/
create table map_file_extent$(
  file_idx     number,                     /* file index */
  ext_num      number,                     /* file extent number */
  ext_dev_off  number,                     /* element offset */
  ext_size     number,                     /* file extent size */
  ext_file_off number,                     /* file offset */
  ext_type     number,                     /* file extent type */
  elem_name    varchar2(2000),             /* element name */
  elem_idx     number                      /* element index */
)
/
create table map_subelement$(
  sub_num      number,                     /* subelement number */
  sub_size     number,                     /* subelement size */
  elem_offset  number,                     /* element offset */
  sub_flags    number,                     /* subelement flags */
  parent_idx   number,                     /* parent element index */
  child_idx    number,                     /* child element index */
  elem_name    varchar2(2000)              /* element name */
)
```

```
/
create table map_element$ (
  elem_name     varchar2(2000),           /* element name */
  elem_cfgid    varchar2(2000),           /* element configuration id */
  elem_type     number,                   /* element type */
  elem_idx      number,                   /* element index */
  elem_size     number,                   /* element size */
  elem_nsubelem number,                   /* number of subelements */
  elem_descr    varchar2(2000),           /* description */
  stripe_size   number,                   /* element stripe size */
  elem_flags    number                    /* flags */
)
/
create table map_extelement$ (
  elem_idx      number,                   /* element index */
  num_attrb     number,                   /* number of attributes */
  attrb1_name   varchar2(30),             /* attribute 1 name */
  attrb1_val    varchar2(30),             /* attribute 1 value */
  attrb2_name   varchar2(30),             /* attribute 2 name */
  attrb2_val    varchar2(30),             /* attribute 2 value */
  attrb3_name   varchar2(30),             /* attribute 3 name */
  attrb3_val    varchar2(30),             /* attribute 3 value */
  attrb4_name   varchar2(30),             /* attribute 4 name */
  attrb4_val    varchar2(30),             /* attribute 4 value */
  attrb5_name   varchar2(30),             /* attribute 5 name */
  attrb5_val    varchar2(30)              /* attribute 5 value */
)
/
create table map_complist$ (
  elem_idx      number,                   /* element index */
  num_comp      number,                   /* number of components */
  comp1_name    varchar2(30),             /* component 1 name */
  comp1_val     varchar2(2000),           /* component 1 value */
  comp2_name    varchar2(30),             /* component 2 name */
  comp2_val     varchar2(2000),           /* component 2 value */
  comp3_name    varchar2(30),             /* component 3 name */
  comp3_val     varchar2(2000),           /* component 3 value */
  comp4_name    varchar2(30),             /* component 4 name */
  comp4_val     varchar2(2000),           /* component 4 value */
  comp5_name    varchar2(30),             /* component 5 name */
  comp5_val     varchar2(2000)            /* component 5 value */
)
/
create global temporary table map_object (
  object_name   varchar2(2000),           /* object name */
  object_owner  varchar2(2000),           /* object owner */
  object_type   varchar2(2000),           /* object type */
  file_map_idx  number,                   /* file index */
  depth         number,                   /* element depth */
  elem_idx      number,                   /* element index */
  cu_size       number,                   /* contiguous unit size */
  stride        number,                   /* stride size */
  num_cu        number,                   /* number of contiguous units */
  elem_offset   number,                   /* element offset */
  file_offset   number,             /* file offset */
  data_type     varchar2(2000),           /* data type */
  parity_pos    number,                   /* parity position */
  parity_period number                    /* parity period */
```

```
) on commit preserve rows
/
create public synonym map_object for sys.map_object
/
grant select on map_object to select_catalog_role
/
grant all on map_object to dba
/

Rem ============================================================
Rem make index on exppkgobj$ unique on two columns
Rem ============================================================

drop index i_objtype
/
create unique index i_objtype on exppkgobj$(type#, class)
/

Rem=========================================================================
Rem  Make changes to the transformations$ table
Rem=========================================================================

ALTER TABLE sys.transformations$ ADD (from_schema  varchar2(30));
ALTER TABLE sys.transformations$ ADD (from_type    varchar2(30));
ALTER TABLE sys.transformations$ ADD (to_schema    varchar2(30));
ALTER TABLE sys.transformations$ ADD (to_type      varchar2(30));


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

ALTER TABLE system.aq$_queues ADD (memory_threshold NUMBER);

Rem ============================================================================
Rem The following block of code upgrades the Object Type System to 9.2.0
Rem It must be done before attempting to create/alter any user-defined
Rem types.
Rem ==================== begin of system type upgrade ====================

Rem initialize kotadx object type

CREATE OR REPLACE LIBRARY UPGRADE_LIB TRUSTED AS STATIC
/

CREATE OR REPLACE PROCEDURE upgrade_system_types_from_901 IS
LANGUAGE C
NAME "UPG_FROM_901"
LIBRARY UPGRADE_LIB;
/

DECLARE
cnt  NUMBER;
objid raw(16);
objnm number;
patch_eoids boolean := FALSE;
BEGIN
```

```
  cnt := 0;
  -- Check if type kotadx exists
  select count(*) into cnt from obj$ o, user$ u where
    o.name = 'KOTADX' and
     o.owner#=u.user# and u.name='SYS' and o.type#=13;

  -- Only run this once
  IF cnt = 0 THEN
    upgrade_system_types_from_901();
  END IF;

END;
/

set serveroutput off;

Rem=========================================================================
Rem END STAGE 1: upgrade from 9.0.1 to 9.2.0
Rem=========================================================================

Rem=========================================================================
Rem BEGIN STAGE 2: Upgrade from 9.2.0 to the new release
Rem=========================================================================

@@c0902000

Rem=========================================================================
Rem END STAGE 2: upgrade from 9.2.0 to the new release
Rem=========================================================================

Rem*************************************************************************
Rem END c0900010.sql
Rem*************************************************************************
```