

## oracle 10g 研究ORACLE\_HOME rdbms admin 下的脚本的功能 (26) c0801070.sql

oracle 10g 研究ORACLE\_HOME rdbms admin 下的脚本的功能 (26) c0801070.sql

```
#upgrade Oracle RDBMS from 8.1.7 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 a0801070.sql
Rem    as catalog.sql and catproc.sql will be run before a0801070.sql
Rem    is invoked.
Rem
Rem    This script is called from u0801070.sql and c0801060.sql
Rem
Rem    This script performs the upgrade in the following stages:
Rem        STAGE 1: upgrade from 8.1.7 to 9.0.1
Rem        STAGE 2: upgrade from 9.0.1 to the new release
#

Rem
Rem $Header: c0801070.sql 02-sep-2004.08:17:06 rburns Exp $
Rem
Rem c0801070.sql
Rem
Rem Copyright (c) 1999, 2004, Oracle. All rights reserved.
Rem
Rem    NAME
Rem        c0801070.sql - upgrade Oracle RDBMS from 8.1.7 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 a0801070.sql
Rem        as catalog.sql and catproc.sql will be run before a0801070.sql
Rem        is invoked.
Rem
Rem        This script is called from u0801070.sql and c0801060.sql
Rem
Rem        This script performs the upgrade in the following stages:
Rem            STAGE 1: upgrade from 8.1.7 to 9.0.1
Rem            STAGE 2: upgrade from 9.0.1 to the new release
Rem
Rem    NOTES
Rem        * This script must be run using SQL*PLUS.
Rem        * You must be connected AS SYSDBA to run this script.
Rem
Rem    MODIFIED    (MM/DD/YY)
Rem    rburns      09/02/04 - remove serveroutput
Rem    rburns      07/15/04 - remove dbms_output compiles
Rem    rburns      02/09/04 - move inserts from i0801070.sql
Rem    avaliani    06/12/03 - update mast_method for resource_plan$
Rem    skabraha    03/31/03 - set index prop as created by constraint
Rem    srtata      02/08/03 - change DDL and DML stmts on aud
Rem    rburns      09/09/02 - revoke all dbsnmp system privileges
Rem    jaysmith    08/16/02 - suppress errors while revoking dbsnmp
Rem    jaysmith    08/16/02 - remove pre-9i DBSNMP priveleges
Rem    prakumar    08/08/02 - bug 2328821 recreate index on sumkey$
```

Rem	ssubrama	06/10/02 - bug 2385207 move delete from dependency\$
Rem	ssubrama	06/02/02 - bug 2385207 delete dependency for no-existent
Rem		x\$ tables from dependency\$
Rem	wesmith	04/23/02 - bug 2338675
Rem	rburns	03/27/02 - add plspur
Rem	rburns	01/06/02 - suppress update errors
Rem	rburns	10/26/01 - wrap drop index statements
Rem	porangas	09/17/01 - Fix bug#1931735
Rem	yzhu	09/10/01 - Fix the spare3 values for NCHAR columns.
Rem	rburns	08/06/01 - remove NCHAR length calculations
Rem	rburns	07/11/01 - bug 1871365
Rem	rburns	06/04/01 - add 9.0.1 upgrade
Rem	rburns	05/02/01 - add nclob columns and remove fixed view
Rem	gmurphy	04/26/01 - only set flags=0 if null, for OLS
Rem	rxgovind	04/26/01 - add dbms_output message for EOID patchup
Rem	arajara	04/18/01 - caching async_updatable_table is not supported
Rem	rxgovind	04/17/01 - fixup Type metadata in migrated dbs from 7.3
Rem	rburns	04/16/01 - drop template_objects_ul constraint
Rem	rburns	04/09/01 - NCHAR fixes
Rem	nshodhan	04/09/01 - bug1725230 : drop views containing "snap"
Rem	jingliu	03/29/01 - remove repcat\$_template_targets
Rem	bpanchap	04/11/01 - Adding index on obj# on tabsubpart
Rem	skabraha	03/27/01 - add for upgrade to 9i
Rem	rburns	03/22/01 - add NCHAR upgrade to UTF8
Rem	tkeefe	03/13/01 - Simplify normalization for n-tier schema.
Rem	rburns	03/08/01 - fix grouped_column_pk
Rem	abrumm	02/22/01 - external_tab\$: use LOBs for storing access params
Rem	apadmana	02/19/01 - Add column timestamp to sumpartlog\$
Rem	rburns	02/08/01 - drop packages no longer used
Rem	arajara	01/10/01 - Move replication specific stmts from a0801070.sql
Rem	rburns	02/07/01 - fix targetrba views
Rem	nshodhan	02/06/01 - Remove sys.exptime\$
Rem	bpanchap	01/12/01 - Adding column to sumpartlog
Rem	nshodhan	12/27/00 - Add repcat\$_template_sites.instantiation_date
Rem	rwessman	01/04/01 - Backed out 9i security enhancements
Rem	twong	01/04/01 - invalidate summary obj after upgrade
Rem	rburns	12/14/00 - remove extraneous DDL & cleanup
Rem	slawande	12/13/00 - Force MAVs to go thru revalidation on upgrade.
Rem	celsbern	12/06/00 - fixing replication upgrade
Rem	dalpern	11/30/00 - privileges for kga debugger
Rem	clei	11/29/00 - add SELECT ANY DICTIONARY privilege
Rem	rasivara	12/13/00 - 1375026:Reset INCREMENT_BY field in IDGEN1\$ to 50
Rem	rburns	11/22/00 - move cdc and viewcon\$ to i0801070.sql
Rem	jingliu	11/20/00 - review comments
Rem	rburns	11/20/00 - drop targetrba views
Rem	bpanchap	11/30/00 - Adding flags column to sumpartlog
Rem	gkulkarn	11/16/00 - Minor correction in spare2 initialization
Rem	nbhatt	11/28/00 - change aq_message_types constraint
Rem	gkulkarn	11/14/00 - Initialize spare2 in OBJ\$ for LogMiner
Rem	kquinn	11/17/00 - 1375879: alter operator -> alter any operator
Rem	rburns	11/09/00 - cleanup
Rem	slawande	11/09/00 - Add sequence# to sumdelta\$.
Rem	liwong	10/29/00 - add def\$_destination.flag
Rem	nshodhan	10/27/00 - upgrade snap_reftime\$: sub_handle, change_view
Rem	lsheng	10/23/00 - add viewcon\$
Rem	celsbern	10/17/00 - updated with 9.0 IAS changes for replication.
Rem	rburns	10/16/00 - move alter aq\$_queues 805->817
Rem	rvissapr	09/05/00 - upgrade aud\$ from 8.1.7 to 9.0

Rem	nbhatt	09/29/00 - upgrade aq_mesage_types table
Rem	rburns	09/19/00 - fix short regress difs
Rem	amganesesh	09/12/00 - dejaview.
Rem	rburns	09/12/00 - fix proxy upgrade
Rem	mthiyaga	09/08/00 - Add dataless col to sumdetail\$
Rem	rburns	09/07/00 - sqlplus fixes
Rem	fputzolu	08/31/00 - upgrade bhiboundval for part. tables & indexes
Rem	liwong	09/01/00 - add master w/o quiesce: fixes
Rem	svivian	09/01/00 - plan stability upgrades
Rem	arrajara	08/31/00 - codepoint semantics: system.repcat\$_repcolumn
Rem	dmwong	08/22/00 - add new column info for fga\$.
Rem	masubram	08/22/00 - add new columns to mlog and snap_reftime
Rem	rburns	08/17/00 - add left out tables and columns
Rem	wesmith	08/01/00 - Materialized views: change version# to hashcode
Rem	mtyulene	08/01/00 - add aux_stats\$ table
Rem	rburns	07/31/00 - move some table creations into i0801070.sql
Rem	liwong	07/12/00 - add total_prop_time_latency
Rem	dmwong	07/07/00 - add fga_log\$ for fine grained audit.
Rem	shihliu	07/21/00 - add resumable privilege
Rem	rwessman	07/05/00 - Added creation of tab_ovf\$
Rem	dmwong	07/07/00 - add fga_log\$ for fine grained audit.
Rem	rguzman	07/26/00 - Adding a column to SEQ\$
Rem	liwong	07/12/00 - add total_prop_time_latency
Rem	elu	06/26/00 - add column ddl_num to template_objects
Rem	elu	06/23/00 - add type hashcode column to repcat tables
Rem	mmorsi	06/29/00 - Adding the external name to procedurejava\$.
Rem	liwong	06/29/00 - add total_txn_count, total_prop_time
Rem	thoang	06/27/00 - add system type upgrade
Rem	elu	06/26/00 - add column ddl_num to template_objects
Rem	elu	06/23/00 - add type hashcode column to repcat tables
Rem	rvenkate	06/23/00 - add username and remove userid from
Rem		repcat\$_repgroup_privs
Rem	awitkows	06/27/00 - extend sumagg with agginfo
Rem	awitkows	06/21/00 - upgrade sumkey\$ for gsets
Rem	rmurthy	06/29/00 - procedureinfo: add impltype columns for
Rem		pipelined & aggr functions
Rem	lbarton	07/05/00 - datapump: remove dictionary table inserts
Rem	mkrishna	06/29/00 - add schemaur1 to the opaque type
Rem	bemeng	06/28/00 - create index on object_stats
Rem	bemeng	06/23/00 - Unused Indexes upgrade: object_stats
Rem	thoang	06/19/00 - add hashcode column to type\$
Rem	rherwadk	06/19/00 - change switch_group parameters
Rem	lbarton	06/13/00 - datapump facility name change
Rem	twotong	06/19/00 - add self join support for summary
Rem	elu	06/12/00 - add ddl_num to repcat\$_ddl
Rem	rwessman	06/08/00 - N-Tier enhancements
Rem	mmorsi	05/15/00 - SQLJ catalog changes.
Rem	dmwong	05/27/00 - add new system privileges in 9.0.
Rem	weiwang	06/14/00 - alter table reg\$
Rem	smuralid	06/06/00 - change opqtype\$
Rem	rvenkate	05/31/00 - Added new index I_SNAP2.
Rem	dmwong	05/27/00 - add new system privileges in 9.0.
Rem	rmurthy	06/06/00 - add short typeid support
Rem	twotong	05/25/00 - add on commit and query rewrite privilege
Rem	jdavison	05/25/00 - Fix update statement for resource_plan_directive
Rem	liwong	05/17/00 - add_master_db w/o quiesce
Rem	liwong	05/16/00 - Add sys.exptime\$
Rem	mkrishna	05/23/00 - add opqtype\$ to the upgrade script

```

Rem    dmwong      05/08/00 - remove i_rls2
Rem    slawande    05/19/00 - Add flag2 to snap$.
Rem    twtong      05/25/00 - add inline# col to sumkey$, sumjoin$, sumpred$,
Rem                                sumdetail$
Rem    bpanchap    05/09/00 - Fixing syntax error
Rem    mmorsi      05/02/00 - adding the spare columns to procedureinfo$.
Rem    mmorsi      05/01/00 - SQLJ Catalog upgrade.
Rem    wixu        05/01/00 - wixu_resman_chg
Rem    bpanchap    04/26/00 - Removing a field from sumpred
Rem    spsundar    05/08/00 - remove not null constraint on dataobj# in indpart
Rem    dmwong      04/27/00 - add fga$ for fine grained auditing
Rem    wesmith     04/24/00 - Add oldest_oid to mlog$
Rem    apadmana    04/20/00 - Replicated Objects MV
Rem    sbodagal    04/26/00 - change the size of user_table_name in ol$hints
Rem    rmurthy     04/24/00 - add inheritance related changes
Rem    tfyu        05/03/00 - initialize spare1 of tabsubpart
Rem    dmwong      04/14/00 - add approle$, rls_ctx$, rls_grp$, context$
Rem    smuralid    04/14/00 - Inheritance: add columns to typed_view$
Rem    jdavison    04/13/00 - Move alter table partobj earlier in upgrade.
Rem    wnorcott    04/12/00 - upgrade/downgrade Change Data Capture
Rem    jdavison    04/11/00 - Modify usage notes for 9.0 changes.
Rem    nagarwal    04/06/00 - remove ustats changes
Rem    sbodagal    03/28/00 - extend outln tables
Rem    bpanchap    03/28/00 - Adding new system table sumpred used for material
Rem    bemeng      04/13/00 - insert default temp tablespace number into props$
Rem    tfyu        04/10/00 - alter tabpart spare1 to scn
Rem    wnorcott    03/08/00 - Change Data Capture metadata
Rem    nagarwal    03/09/00 - add extensible optimizer changes
Rem    tfyu        03/23/00 - alter sum, sumdetail and sumkey adding columns
Rem    narora      02/21/00 - add setnum to sys.snap_refop
Rem    amozes      01/27/00 - bitmap join index
Rem    wixu        01/24/00 - Changes_for_RES_MANAGER_extensions
Rem    spsundar    02/21/00 - add table indpart_param$
Rem    elu         01/24/00 - add column apply_init to def$_destination
Rem    amozes      02/02/00 - add col_usage
Rem    gclaborn    11/30/99 - Add Metadata API stuff
Rem    rshaikh     10/29/99 - Created

```

```
Rem
```

```
Rem=====
```

```
Rem BEGIN STAGE 1: upgrade from 8.1.7 to 9.0.1
```

```
Rem=====
```

```
REM =====
```

```
REM INSERT into CDC tables (moved from i0801070.sql)
```

```
REM =====
```

```

insert into cdc_change_sources$
    (source_name,dbid,logfile_location,logfile_suffix,source_description,created)
    values('SYNC_SOURCE',NULL,'N/A',NULL,'SYNCHRONOUS CHANGE SOURCE',SYSDATE)
/

```

```

insert into cdc_change_sets$
    (set_name,change_source_name,begin_date,end_date,begin_scn,end_scn,
    freshness_date,freshness_scn,advance_enabled,ignore_ddl,created,
    rollback_segment_name,advancing,purging,lowest_scn,tablespace,
    lm_session_id,partial_tx_detected,last_advance,last_purge)
    values('SYNC_SET','SYNC_SOURCE',SYSDATE,NULL,NULL,0,
    NULL,NULL,'N','Y',SYSDATE,NULL,'N',0,'N/A',NULL,'N',NULL,NULL)

```

/

```
Rem =====
Rem BEGIN add audit rows to procedure$ (required for recompile)
Rem =====
```

```
insert into procedure$
  select obj#, '-----', NULL, 1
  from obj$ where namespace = 3;
commit;
```

```
Rem =====
Rem END add audit rows to procedure$ (required for recompile)
Rem =====
```

```
Rem=====
Rem Add new system privileges here !!
Rem=====
```

```
update SYSTEM_PRIVILEGE_MAP set name='UNDER ANY TYPE'
where privilege=-186;
```

```
update SYSTEM_PRIVILEGE_MAP set name='UNDER ANY VIEW'
where privilege=-209;
```

```
update SYSTEM_PRIVILEGE_MAP set name='ALTER ANY OPERATOR'
where privilege=-202;
```

```
insert into SYSTEM_PRIVILEGE_MAP values (-213, 'UNDER ANY TABLE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-229, 'CREATE SECURITY PROFILE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-230, 'CREATE ANY SECURITY PROFILE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-231, 'DROP ANY SECURITY PROFILE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-232, 'ALTER ANY SECURITY PROFILE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-233, 'ADMINISTER SECURITY', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-234, 'ON COMMIT REFRESH', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-235, 'EXEMPT ACCESS POLICY', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-236, 'RESUMABLE', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-237, 'SELECT ANY DICTIONARY', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-238, 'DEBUG CONNECT SESSION', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-239, 'DEBUG CONNECT USER', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-240, 'DEBUG CONNECT ANY', 0);
insert into SYSTEM_PRIVILEGE_MAP values (-241, 'DEBUG ANY PROCEDURE', 0);
```

```
grant all privileges to dba with admin option;
```

```
Rem=====
Rem Add new object privileges here !!
Rem=====
```

```
insert into TABLE_PRIVILEGE_MAP values (22, 'UNDER');
insert into TABLE_PRIVILEGE_MAP values (23, 'ON COMMIT REFRESH');
insert into TABLE_PRIVILEGE_MAP values (24, 'QUERY REWRITE');
insert into TABLE_PRIVILEGE_MAP values (26, 'DEBUG');
```

```
Rem=====
Rem Add new audit options here !!
Rem=====
```

```

insert into STMT_AUDIT_OPTION_MAP values (229, 'ON COMMIT REFRESH', 0);
insert into STMT_AUDIT_OPTION_MAP values (236, 'RESUMABLE', 0);
insert into STMT_AUDIT_OPTION_MAP values (237, 'SELECT ANY DICTIONARY', 0);
insert into STMT_AUDIT_OPTION_MAP values (238, 'DEBUG CONNECT SESSION', 0);
insert into STMT_AUDIT_OPTION_MAP values (239, 'DEBUG CONNECT USER', 0);
insert into STMT_AUDIT_OPTION_MAP values (240, 'DEBUG CONNECT ANY', 0);
insert into STMT_AUDIT_OPTION_MAP values (241, 'DEBUG ANY PROCEDURE', 0);
insert into STMT_AUDIT_OPTION_MAP values (242, 'DEBUG PROCEDURE', 0);

```

```

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

```

```

delete from dependency$ where d_obj# in (select obj# from obj$ where name in
('V_$TARGETRBA', 'GV_$TARGETRBA',
'V_$RECOVERY_SERVERS', 'GV_$RECOVERY_SERVERS',
'V_$RECOVERY_TRANSACTIONS', 'GV_$RECOVERY_TRANSACTIONS'));

```

```
commit;
```

```
alter system flush shared_pool;
```

```

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

```

```

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

```

```

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

```

```

drop view DBA_CACHEABLE_OBJECTS;
drop public synonym DBA_CACHEABLE_OBJECTS;

```

```

drop view DBA_CACHEABLE_TABLES;
drop public synonym DBA_CACHEABLE_TABLES;

```

```

rename DBA_SNAPSHOT_LOG_FILTER_COLS to DBA_MVIEW_LOG_FILTER_COLS;
create or replace public synonym DBA_SNAPSHOT_LOG_FILTER_COLS
for DBA_MVIEW_LOG_FILTER_COLS;

```

```

rename ALL_SNAPSHOT_REFRESH_TIMES to ALL_MVIEW_REFRESH_TIMES;
create or replace public synonym ALL_SNAPSHOT_REFRESH_TIMES
for ALL_MVIEW_REFRESH_TIMES;

```

```

rename DBA_SNAPSHOT_REFRESH_TIMES to DBA_MVIEW_REFRESH_TIMES;
create or replace public synonym DBA_SNAPSHOT_REFRESH_TIMES
for DBA_MVIEW_REFRESH_TIMES;

```

```

rename USER_SNAPSHOT_REFRESH_TIMES to USER_MVIEW_REFRESH_TIMES;
create or replace public synonym USER_SNAPSHOT_REFRESH_TIMES
for USER_MVIEW_REFRESH_TIMES;

```

```
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;
```

```
Rem=====
```

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

```
Rem=====
```

```
drop package DBMS_SUMREF_CHILD;
```

```
drop package DBMS_SUMREF_PARENT;
```

```
drop package DBMS_SUMREF_UTIL2;
```

```
drop package DBMS_HS_EXTPROC;
```

```
Rem=====
```

```
Rem Add changes to dictionary tables here !!
```

```
Rem=====
```

```
Rem Make sure that the Lob generator sequence is 50
```

```
alter sequence SYS.IDGEN1$ increment by 50;
```

```
Rem=====
```

```
Rem ===== Begin changes for LogMiner Project =====
```

```
Rem=====
```

```
Rem Initailize the SPARE2 column in OBJ$ used for storing the
```

```
Rem "OBJECT VERSION (OBJV#)" for the dictionary object.
```

```
Rem
```

```
Rem      Type#          Intial Value
```

```
Rem      2              1
```

```
Rem      All Others      65535
```

```
Rem
```

```
update sys.obj$ set spare2 = (decode (type#, 2, 1, 65535));
```

```
Rem=====
```

```
Rem ===== End changes for LogMiner Project =====
```

```
Rem=====
```

```
Rem=====
```

```
Rem ===== Begin changes for Type Evolution Project =====
```

```
Rem=====
```

```
ALTER TABLE type$ ADD hashcode      raw(17);
```

```
Rem=====
```

```
Rem ===== End changes for Type Evolution Project =====
```

```
Rem=====
```

```
Rem=====
```

```
Rem BEGIN Security Layer changes
```

```
Rem=====
```

```

BEGIN
    EXECUTE IMMEDIATE 'DROP INDEX i_rls';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -1418 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/

ALTER TABLE rls$
add
(
    gname          VARCHAR2(30),          /* name of policy group */
    ptype          NUMBER                 /* policy type */
)
/

rem setting all 8.i style policies to SYS_DEFAULT group
UPDATE rls$
set gname = 'SYS_DEFAULT'
/

ALTER TABLE rls$
modify
(
    gname          NOT NULL               /* change to not NULL */
)
/

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

create table rls_grp$
(
    obj#           NUMBER NOT NULL,       /* parent object number */
    gname          VARCHAR2(30) NOT NULL  /* name of policy group */
)
/

create index i_rls_grp on rls_grp$(obj#)
/

create table rls_ctx$
(
    obj#           NUMBER NOT NULL,       /* parent object number */
    ns             VARCHAR2(30) NOT NULL, /* namespace */
    attr           VARCHAR2(30) NOT NULL  /* attribute */
)
/

create index i_rls_ctx on rls_ctx$(obj#)
/

alter table context$
add
(
    flags          number                 /* for new ctx type */

```



```

)
/

update context$
set flags=0
where flags is null
/

alter table context$
modify
(
    flags          not null          /* make the column not NULL */
)
/

create table approle$                                /* Application Role */
(
    role#          NUMBER NOT NULL,                                /* role# */
    schema          VARCHAR2(30) NOT NULL,                        /* schema of policy function */
    package         VARCHAR2(30) NOT NULL                        /* policy package name */
)
/

create unique index i_approle on approle$(role#)
/

```

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
    -- clientid column represents client identifier from ksuse
    -- sessioncpu column represents cpu per session
    sql_stmt := 'ALTER TABLE ' || schema_name || '.AUD$ ADD (' ||
        ' clientid  varchar2(64),' ||
        ' sessioncpu number'      ||
        ' )';
    EXECUTE IMMEDIATE sql_stmt;
END;
/

```

```

create table fga$
(
    obj#          NUMBER NOT NULL,
    pname         VARCHAR2(30) NOT NULL,
    ptxt          VARCHAR2(4000) NOT NULL,
    pfschma       VARCHAR2(30),
    ppname        VARCHAR2(30),
    pfname        VARCHAR2(30),

```

```

        pcol          VARCHAR2(30),
        enable_flag    NUMBER NOT NULL
    )
/
create index i_fga on fga$(obj#)
/

```

```

create table fga_log$
(
    sessionid      number not null,
    timestamp#     date not null,
    dbuid          varchar2(30),
    osuid          varchar2(255),
    oshst          varchar2(128),
    clientid       varchar2(64),
    extid          varchar2(4000),
    obj$schema     varchar2(30),
    obj$name       varchar2(128),
    policyname     varchar2(30),
    scn            number,
    sqltext        varchar2(4000),
    sqlbind        varchar2(4000),
    comment$text   varchar2(4000)
)
/

```

```

Rem =====
Rem END security layer changes
Rem =====

```

```

Rem =====
Rem resource_plan$ and resource_plan_directive$ column additions
Rem =====

```

```

ALTER TABLE resource_plan$
add
(
    que_method      varchar2(30)          /* queueing method */
)
/

```

```

Rem =====
Rem update the columns to the original default values in case the client
Rem does upgrade, downgrade and then upgrade again..
Rem
update resource_plan$ set que_method='FIFO_TIMEOUT';
commit
/

```

```

update resource_plan$ set mast_method='ACTIVE_SESS_POOL_ABSOLUTE';
commit
/

```

```

ALTER TABLE resource_plan_directive$
add
(
    active_sess_pool_p1  number,          /* NEW mast param */

```

```

queueing_pl          number,                /* queue timeout in sec */
switch_group         varchar2(30),          /* group to switch to */
switch_time          number,                /* time limit for execution within a group */
switch_estimate       number,                /* use execution time estimate? */
max_est_exec_time     number,                /* max. estimate time in sec */
undo_pool            number                 /* max. cumulative undo allocation */
)
/

```

```

update resource_plan_directive$
set active_sess_pool_pl=1000000,
    queueing_pl=1000000,
    switch_group=NULL,
    switch_time=1000000,
    switch_estimate=0,
    max_est_exec_time=1000000,
    undo_pool=1000000;
commit
/

```

```

Rem =====
Rem Inheritance related changes
Rem =====

```

```

alter table type$
add
(
    local_attrs    number,                /* Number of local attributes */
    local_methods  number,                /* Number of local methods */
    typeid         raw(16),                /* short type id */
    roottoid       raw(16)                /* TOID of root type (null if not subtype) */
)
/

```

```

create table typehierarchy$
( toid            raw(16) not null,        /* TOID of the root type */
  next_typeid     raw(16) not null,        /* next available typeid */
  spare1          number,                 /* reserved */
  spare2          number)                 /* reserved */
/
create unique index i_typehierarchy$ on typehierarchy$(toid)
/

```

Rem moved superobj\$ creation to i0801070.sql - required for ALTER

```

alter table typed_view$
add
(
    undertextlength number,                /* length of under clause text for sub-views */
    undertext        varchar2(4000)        /* under clause text for sub-views */
)
/

```

```

alter table attribute$
add
(
    xflags    number,                /* flags not stored in TDO */
    spare4    number,                /* spare column - reserved for future use */

```

```

    spare5    number                /* spare column - reserved for future use */
)
/

alter table method$
add
(
    xflags    number                /* flags not stored in TDO */
)
/

alter table coltype$ add (TYPIDCOL# number)
/

Rem moved creation of subcoltype$ to i0801070.sql

Rem =====
Rem END OF Inheritance related changes
Rem =====

Rem =====
Rem BEGIN OF Partition related changes
Rem =====

Rem update character limits for partition columns

update partcol$ set spare1 =
    (select max(charsetid) from col$ where charsetform = 1), charsetform = 1
    where charsetform = 0 and type# in (1, 8, 96, 112);
update subpartcol$ set spare1 =
    (select max(charsetid) from col$ where charsetform = 1), charsetform = 1
    where charsetform = 0 and type# in (1, 8, 96, 112);

update partcol$ set spare1 =
    (select max(charsetid) from col$ where charsetform = 1)
    where spare1 = 0 and charsetform = 1;
update subpartcol$ set spare1 =
    (select max(charsetid) from col$ where charsetform = 1)
    where spare1 = 0 and charsetform = 1;

update partcol$ set spare1 =
    (select max(charsetid) from col$ where charsetform = 2)
    where spare1 = 0 and charsetform = 2;
update subpartcol$ set spare1 =
    (select max(charsetid) from col$ where charsetform = 2)
    where spare1 = 0 and charsetform = 2;

commit;

Rem The column spare1 of tabpart$ is nullable in 8.1.6
update tabpart$ set spare1 = 0 where spare1 is null;
commit
/

Rem The column spare1 of tabsubpart$ is nullable in 8.1.6
update tabsubpart$ set spare1 = 0 where spare1 is null;
commit
/

```

```

Rem Add index on obj# on tabsubpart$; this index is used for partition
Rem SCN read and update
create index i_tabsubpart$_obj$ on tabsubpart$(obj#)
/

Rem
Rem sumpartlog$ table
Rem This table has one row per table partition being dropped or its dataobj#
Rem changed
Rem obj# is a key; and so is (bo#, part#)
Rem There is a non-unique index on bo#, obj#
Rem
create table sumpartlog$ (
  obj#          number not null,          /* object number of partition */
  /* DO NOT CREATE INDEX ON DATAOBJ# AS IT WILL BE UPDATED IN A SPACE
  * TRANSACTION DURING TRUNCATE */
  dataobj#      number,                  /* data layer object number */
  bo#           number not null,         /* object number of base table */
  newobj#       number,                 /* new object number of partition if any */
  newdataobj#   number,                 /* new data layer object number if any */
  pobj#         number,                 /* partition object number; populated when
                                      TRUNCATE/COALESCE of subpartition */
  hiboundlen    number not null,        /* length of high bound value expression */
  loboundlen    number not null,        /* length of low bound value expression */
  boundvals     long,                  /* concatenated text of low-and high-bound */
                                      /* value expression */
  parttype      number,                 /* partition type */
                                      /* 1=RANGE, 2=COMPOSITE, 3=LIST */
  pmoptype      number,                 /* recorded PMOP type */
  timestamp     date not null,          /* Time when the PMOP occurred. */
  scn           number,                 /* summary sequence number */
  flags         number,                 /* 0x01 It is a table operation */
  /* These spare columns are for future needs, e.g. values for the
  * PARALLEL(degree, instances) parameters. */
  spare1        number,
  spare2        number,
  spare3        number)
/

create index i_sumpartlog$ on sumpartlog$(bo#, obj#)
/

create index i_sumpartlog$_bopart$ on sumpartlog$(bo#, dataobj#)
/

create table indpart_param$ ( /* stores partition specific parameter string */
  obj#          number not null,        /* object number of partition */
  parameters     varchar2(1000)         /* parameter string per index partition */
)
/

create unique index i_indpart_param on indpart_param$(obj#)
/

Rem The column dataobj# of indpart$ and indsubpart$ are nullable in 9.0
alter table indpart$ modify (dataobj# number null);
alter table indsubpart$ modify (dataobj# number null);
update indpart$ set dataobj# = NULL where dataobj# = 4294967295;
commit
/

update indsubpart$ set dataobj# = NULL where dataobj# = 4294967295;
commit

```

```

/

Rem =====
Rem END OF Partion related changes
Rem =====

alter table indtypes$ add(interface_version#    number)
/

alter table association$ add(interface_version#    number)
/

create table secobj$
(
    obj#        number not null,                /* object number of index */
    secobj#     number not null,                /* object number for secondary object */
    spare1      number,
    spare2      number
)
/

Rem=====
Rem BEGIN OF Metadata tables
Rem=====

create table metaview$ /* Used by mdAPI to select which view per object type */
(
    type        varchar2(30) not null,          /* 'TABLE', 'FULL_TYPE', etc */
    flags        number not null, /* Might have mult. views per obj class for */
    /* performance: base rel. tables (fast), part. tbls, object tables, etc. */
    properties   number not null,                /* dict. object type's properties */
    /* 0x0001 = 1 = schema object */
    model        varchar2(30) not null,          /* 'ORACLE', 'ANSI', 'CWM', etc */
    version      number not null,                /* decimal RDBMS version: eg, 0802010000 */
    /* indicates which view to use for client's requested version */
    xmltag       varchar2(30) not null,          /*XML tag to use for each object */
    udt          varchar2(30) not null,          /* UDT name for object view */
    schema       varchar2(30) not null,          /* schema for view */
    viewname     varchar2(30) not null           /* view to use for */
    /* this type, model and version */
)
/
create unique index i_metaview$ on metaview$(type, model, version, flags)
/
create table metafilter$ /* maps filters in mdAPI to UDT attributes */
(
    filter       varchar2(30) not null,          /* documented filter. name */
    type         varchar2(30) not null,          /* dict. obj type: e.g, 'TABLE' */
    model        varchar2(30) not null,          /* model name */
    properties   number not null,                /* filter properties */
    /* 0x01 = boolean filter, 0x02 = expression filter */
    /* 0x04 = custom filter, 0x08 = has default */
    view_attr    number not null,                /* view flag bits (boolean filters only) */
    attrname     varchar2(2000),                /* filtering attribute */
    default_val  number
)
/
create unique index i_metafilter$ on metafilter$(filter, type, model)

```

```

/
create table metaxsl$                                     /* metadata xsl table */
( xmltag          varchar2(30) not null,                  /* xml tag */
  transform        varchar2(30) not null,                  /* transform name */
  model            varchar2(30) not null,                  /* model name */
  script           varchar2(2000) not null)                /* URI of xsl script */
/

create table metaxslparam$      /* legal parameters for mdAPI's XSL scripts */
(
  model            varchar2(30) not null,                  /* model name */
  transform        varchar2(30) not null,                  /* transform name */
  type             varchar2(30) not null,                  /* dict. obj type: e.g, 'TABLE' */
  param            varchar2(30) not null,                  /* documented param. name */
  default_val      varchar2(2000)
)
/

create unique index i_metaxslparam$ on metaxslparam$(model, transform,
type, param)
/

create table metastylesheet     /* Storage for the XSL stylesheets themselves */
( name              varchar2(30) not null,                  /* stylesheet name */
  model             varchar2(30) not null,                  /* model that uses this ss */
  stylesheet         clob)                                   /* stylesheet body */
/

Rem=====
Rem  END OF Metadata tables
Rem=====

Rem=====
Rem  BEGIN Deferred Transactions upgrade
Rem=====

---
--- def$_destination has new column apply_init (for internal use only)
---

alter table system.def$_destination add (apply_init  varchar2(4000))
/

alter table system.def$_destination add (flag      RAW(4) default '00000000')
/

alter table system.def$_origin add(catchup RAW(16) default '00')
/

--- def$_destination has new primary key
--- need to drop and recreate two foreign keys
alter table system.repcat$_repschema drop constraint repcat$_repschema_dest
/

alter table system.def$_calldest add(catchup raw(16) default '00')
/

alter table system.def$_calldest drop constraint def$_call_destination
/

BEGIN

```

```

EXECUTE IMMEDIATE 'drop index system.def$_calldest_n2';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -1418 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/
alter table system.def$_destination add(catchup raw(16) default '00',
                                         alternate char(1) default 'F')
/

alter table system.def$_destination add(total_txn_count      number default 0,
                                         total_prop_time_latency number default 0,
                                         total_prop_time_throughput number default 0,
                                         to_communication_size  number default 0,
                                         from_communication_size number default 0,
                                         spare1                 number default 0,
                                         spare2                 number default 0,
                                         spare3                 number default 0,
                                         spare4                 number default 0)
/

alter table system.def$_destination drop constraint def$_destination_primary
/

alter table system.def$_destination add constraint def$_destination_primary
primary key (dblink, catchup)
/

alter table system.def$_calldest add constraint def$_call_destination
foreign key(dblink, catchup)
references system.def$_destination(dblink, catchup)
/

Rem=====
Rem  END Deferred Transactions upgrade
Rem=====
Rem=====
Rem  BEGIN Replication upgrade
Rem=====

alter table system.repcat$_repschema add(extension_id raw(16) default '00')
/

alter table system.repcat$_repschema add constraint repcat$_repschema_dest
foreign key(dblink, extension_id)
references system.def$_destination(dblink, catchup)
/

rem repcat$_template_objects must store the ddl number to handle multiple ddls
alter table system.repcat$_template_objects add(ddl_num NUMBER DEFAULT 1)
/

alter table system.repcat$_template_objects
add (schema_name varchar2(30))
/

```



```

alter table system.repcat$_template_objects
    drop constraint repcat$_template_objects_u1
/

alter table system.repcat$_template_objects
    add constraint repcat$_template_objects_u1
        unique (object_name,object_type,refresh_template_id,schema_name, ddl_num)
/

alter table system.repcat$_template_sites
    add(instantiation_date date)
/

rem repcat$_ddl must store the ddl number to handle multiple ddls
alter table system.repcat$_ddl add(ddl_num INTEGER DEFAULT 1)
/

rem add column for type hashcode to repcat$_repobject, repcat$_repcolumn
rem and repcat$_flavor_objects
alter table system.repcat$_repobject add(hashcode RAW(17))
/

alter table system.repcat$_repcolumn add(hashcode RAW(17))
/

alter table system.repcat$_flavor_objects add(hashcode RAW(17))
/

REM add new column username to REPCAT$_REPGROUP_PRIVS. populate corresponding
REM data for username from userid. set userid column to NULL
ALTER TABLE system.REPCAT$_REPGROUP_PRIVS ADD (username VARCHAR2(30))
/

BEGIN
    EXECUTE IMMEDIATE 'UPDATE system.REPCAT$_REPGROUP_PRIVS rp set rp.username = ' ||
        '(select u.username from dba_users u where u.user_id = rp.userid)';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -942 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP INDEX system.repcat$_repgroup_privs_n1';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -1418 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/

ALTER TABLE system.REPCAT$_REPGROUP_PRIVS drop constraint
    repcat$_repgroup_privs_uk
/

```

```

ALTER TABLE system.REPCAT$_REPGROUP_PRIVS add constraint
    repcat$_repgroup_privs_uk UNIQUE (username, gname, gowner)
/
ALTER TABLE system.REPCAT$_REPGROUP_PRIVS modify
    (username NOT NULL)
/
ALTER TABLE system.REPCAT$_REPGROUP_PRIVS modify
    (userid NULL)
/
BEGIN
    EXECUTE IMMEDIATE 'CREATE INDEX system.repcat$_repgroup_privs_n1 ON' ||
        ' system.repcat$_repgroup_privs(global_flag, username)';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -942 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/
BEGIN
    EXECUTE IMMEDIATE 'UPDATE system.REPCAT$_REPGROUP_PRIVS rp set rp.userid = NULL';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -942 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/
commit
/

REM repcat$_reproject
ALTER TABLE system.repcat$_reproject
    DROP CONSTRAINT repcat$_reproject_type
/
ALTER TABLE system.repcat$_reproject
    ADD CONSTRAINT repcat$_reproject_type
        CHECK (type IN (-1, 1, 2, 4, 5, 7, 8, 9, 11, 12, -3,
                        -4, 13, 14, 32, 33))
/

ALTER TABLE system.repcat$_reproject
    DROP CONSTRAINT repcat$_reproject_status
/

ALTER TABLE system.repcat$_reproject
    ADD CONSTRAINT repcat$_reproject_status
        CHECK (status IN (0, 1, 2, 3, 4, 5, 6))
/

ALTER TABLE system.repcat$_reproject ADD (version# NUMBER)
/

ALTER TABLE system.repcat$_reproject
    ADD CONSTRAINT repcat$_reproject_version
        CHECK (version# >= 0 AND version# < 65536)
/

```

```

REM repcat$_repcolumn
ALTER TABLE system.recat$_repcolumn
    DROP CONSTRAINT repcat$_repcolumn_uk
/

ALTER TABLE system.recat$_repcolumn ADD
    (clength          NUMBER,
     version#         NUMBER,
     lname            VARCHAR2(4000),
     toid             RAW(16),
     ctype_name       VARCHAR2(30),
     ctype_owner      VARCHAR2(30),
     top              VARCHAR2(30),
     property         RAW(4) default '00000000')
/

ALTER TABLE system.recat$_repcolumn
    ADD CONSTRAINT repcat$_repcolumn_version
        CHECK (version# >= 0 AND version# < 65536)
/

REM system.recat$_repcolumn.pos is nullable
ALTER TABLE system.recat$_repcolumn MODIFY (pos NULL)
/

REM repcat$_parameter_column
ALTER TABLE system.recat$_parameter_column
    drop constraint repcat$_parameter_column_pk
/

ALTER TABLE system.recat$_parameter_column ADD
    (column_pos          NUMBER,
     attribute_sequence_no NUMBER)
/

ALTER TABLE system.recat$_parameter_column
    modify parameter_column_name VARCHAR2(4000)
/

REM repcat$_grouped_column
ALTER TABLE system.recat$_grouped_column
    ADD (pos NUMBER)
/

REM system.recat$_flavor_objects
ALTER TABLE system.recat$_flavor_objects
    ADD (version#    NUMBER)
/

ALTER TABLE system.recat$_flavor_objects
    ADD CONSTRAINT repcat$_flavor_objects_version
        CHECK (version# >= 0 AND version# < 65536)
/

REM system.recat$_template_objects
ALTER TABLE system.recat$_template_objects
    DROP CONSTRAINT repcat$_template_objects_c1
/

```

```

ALTER TABLE system.recat$_template_objects
  ADD CONSTRAINT recat$_template_objects_c1
    CHECK (object_type IN (-1, 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, -5,
                          13, 14, 32, 33))
/

ALTER TABLE system.recat$_template_objects
  ADD (object_version# NUMBER)
/

ALTER TABLE system.recat$_template_objects
  ADD CONSTRAINT recat$_template_objects_ver
    CHECK (object_version# >= 0 AND object_version# < 65536)
/

REM system.recat$_repcatlog
ALTER TABLE system.recat$_repcatlog
  DROP CONSTRAINT recat$_repcatlog_type
/

ALTER TABLE system.recat$_repcatlog
  ADD CONSTRAINT recat$_repcatlog_type
    CHECK (type IN (-1, 0, 1, 2, 4, 5, 7, 8, 9, 11, 12, -3,
                  13, 14, 32, 33))
/

ALTER TABLE system.recat$_repcatlog
  MODIFY (a_comment VARCHAR2(2000))
/

ALTER TABLE system.recat$_repcatlog
  DROP CONSTRAINT recat$_repcatlog_request
/

ALTER TABLE system.recat$_repcatlog
  ADD CONSTRAINT recat$_repcatlog_request
    CHECK (request IN (-1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                     11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
                     21, 22, 23, 24, 25))
/

REM system.recat$_repprop
ALTER TABLE system.recat$_repprop
  ADD (extension_id RAW(16) DEFAULT '00')
/

BEGIN
  EXECUTE IMMEDIATE 'DROP INDEX system.recat$_repprop_dblink_how';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE = -1418 THEN NULL;
    ELSE RAISE;
  END IF;
END;
/

BEGIN

```

```

EXECUTE IMMEDIATE 'DROP INDEX system.repcat$_repprop_dblink_how';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE = -1418 THEN NULL;
    ELSE RAISE;
    END IF;
END;
/

Rem=====
Rem  END Replication upgrade
Rem=====

create table col_usage$
(
  obj#          number,          /* object number */
  intcol#       number,          /* internal column number */
  equality_preds number,          /* equality predicates */
  equijoin_preds number,         /* equijoin predicates */
  nonequijoin_preds number,      /* nonequijoin predicates */
  range_preds   number,          /* range predicates */
  like_preds    number,          /* (not) like predicates */
  null_preds    number,          /* (not) null predicates */
  timestamp     date    /* timestamp of last time this row was changed */
)
/
create unique index i_col_usage$ on col_usage$(obj#,intcol#)
/

Rem =====
Rem BEGIN materialized views upgrade
Rem =====

REM invalidate all summary objects after upgrade
UPDATE obj$ SET status = 5 WHERE type# = 42
/
commit
/

REM add the setnum column to sys.snap_refop$
ALTER TABLE sys.snap_refop$
ADD
(
  setnum          integer default 0 /* the set of queries for a given */
                                          /* table number, used for many-many */
                                          /* subqueries or UNIONS */
)
/

REM add setnum to the unique constraint i_snap_refop1
BEGIN
  EXECUTE IMMEDIATE 'DROP INDEX sys.i_snap_refop1';
EXCEPTION
  WHEN OTHERS THEN
    IF SQLCODE = -1418 THEN NULL;
    ELSE RAISE;
    END IF;
END;

```

/

```
CREATE UNIQUE INDEX sys.i_snap_refop1 ON
  sys.snap_refop$(sowner, vname, instsite, operation#, tabnum, setnum)
```

/

REM Add xpflags, numinlines, numwhrnodes, numhavnodes to sys.sum\$

ALTER TABLE sys.sum\$

ADD

```
(
  xpflags      number,          /* extension to pflags */
  numinlines   integer,        /* number of inline views in summary */
  numwhrnodes  integer,        /* number of nodes in where tree */
  numhavnodes  integer         /* number of nodes in having tree */
)
```

/

REM add inline# to sumdep\$

ALTER TABLE sys.sumdep\$

ADD

```
(
  inline#      number          /* inline view number for summary dependent obj */
)
```

/

REM add inline# to sumdetail\$

ALTER TABLE sys.sumdetail\$

ADD

```
(
  inline#      number,          /* inline view identifier */
  instance#    number,          /* instance # for duplicate table */
  dataless     number          /* detail table is dataless */
)
```

/

REM add inline# to sumkey\$

ALTER TABLE sys.sumkey\$

ADD

```
(
  inline#      number,          /* inline view identifier */
  instance#    number          /* instance # for duplicate table */
)
```

/

REM add inline1 and inline2 to sumjoin\$

ALTER TABLE sys.sumjoin\$

ADD

```
(
  inline1#     number,          /* left inline view number */
  inline2#     number,          /* right inline view number */
  instance1#   number,          /* instance # for tabobj1 */
  instance2#   number          /* instance # for tabobj2 */
)
```

/

REM Create new sustem table sumpred\$ to store where/having clause tree

create table sumpred\$ /\* summary where/having pred tree \*/

(sumobj# number not null, /\* summary object number \*/

```

nodeid      number not null,      /* id that identifies a tree node */
pnodeid     number not null,      /* parent node id */
clauseid    integer not null,     /* clause type: WHERE, HAVING, EUT... */
nodetype    integer not null,     /* AND, OR, COL_REL_CONST ... */
numchild    integer,              /* num. of children for AND, OR nodes */
relop       integer,              /* <, >, ..., RP, IN-LIST... */
loptype     integer,              /* left operand type: COL, AGG, ... */
roptype     integer,              /* right operand type: COL, AGG, ... */
ldobj#      number,               /* left detail table object number */
rdobj#      number,               /* right detail table object number */
lcolid      number,               /* left column id if loptype=COL */
rcolid      number,               /* right column id if roptype=COL */
laggtype    integer,              /* OPTTYPE for left operand if AGG. */
raggtype    integer,              /* OPTTYPE for right operand if AGG. */
lcanotxt    varchar2(4000),       /* left operand normalized string */
rcanotxt    varchar2(4000),       /* right operand normalized string */
lcanotxtlen integer,              /* left operand string length */
rcanotxtlen integer,              /* right operand string length */
ltx         varchar2(4000),       /* string for left expr */
rtx         varchar2(4000),       /* string for right expr */
ltxlen      integer,              /* left expr length */
rtxlen      integer,              /* right expr length */
value       long,                 /* value of oper. if optype = CONST */
valuelen    integer,              /* value length */
numval      integer,              /* number of values in in-list */
colpos      integer,              /* used for multi-column in-lists */
lflags      number,               /* left operand miscellaneous info */
rflags      number,               /* right operand miscellaneous info */
linline#    number,               /* left inline view number */
rinline#    number,               /* right inline view number */
linstance#  number,               /* instance # for left detail tab */
rinstance#  number                /* instance # for right detail tab */

```

```

)
/
create index i_sumpred$1 on sumpred$(sumobj#, clauseid)
/

```

REM Create new system table suminline\$ to store inline views

```

create table suminline$           /* summary inline view table */
( sumobj#      number not null,   /* object number */
  inline#      number not null,   /* inline view unique identifier */
  textpos      number not null,   /* inline view offset starting position */
  textlen      number not null,   /* inline view text length */
  hashval      number not null,   /* hash value generated from the inline */
                                     /* view text */

  spare1       number,
  spare2       number,
  spare3       varchar2(1000),
  spare4       date,
  instance#    number              /* instance # for duplicate inline view */
)
/

```

```

create index i_suminline$1 on suminline$(sumobj#)
/

```

```

create index i_suminline$2 on suminline$(inline#)
/

```

```

create index i_suminline$3 on suminline$(hashval)
/

```

```

REM add columns to sumdelta$
alter table sumdelta$
add
(
    sequence          number                      /* sequence# */
)
/

```

```

REM add columns for objects MVs/extended MV flags
alter table snap$
add
(
    objflag           number,                      /* object properties of snapshot */
    sna_type_oid      raw(16),                     /* object MV type OID */
    sna_type_hashcode raw(17),                     /* object MV type hashcode */
    sna_type_owner    varchar2(30),                /* object MV type owner */
    sna_type_name     varchar2(30),                /* object MV type name */
    mas_type_oid      raw(16),                     /* master object table type OID */
    mas_type_hashcode raw(17),                     /* master object table type hashcode */
    mas_type_owner    varchar2(30),                /* master object table type owner */
    mas_type_name     varchar2(30),                /* master object table type name */
    parent_sowner     varchar2(30),                /* parent snapshot owner */
    parent_vname      varchar2(30),                /* parent snapshot name */
    rel_query         clob,                        /* relational transformation of query */
    flag2             number                      /* extended snapshot flag */
)
/

```

```

REM set flag2 to zero
UPDATE snap$ SET flag2 = 0
/

```

```

REM set objflag to zero
UPDATE snap$ SET objflag = 0
/
commit
/

```

```

create index i_snap2 on
    snap$(parent_vname, parent_sowner, instsite)
/

```

```

REM add table for objects MVs
create table snap_objcol$                      /* snapshot object column information */
(
    sowner            varchar2(30) not null,      /* snapshot view owner */
    vname             varchar2(30) not null,      /* snapshot view name */
    instsite          integer default 0,          /* instantiating site */
    tabnum            integer not null, /* master table this column belongs to */
    snacol            varchar2(30) not null,      /* snapshot column name */
    mascol            varchar2(30),               /* associated master column name */
    flag              number,                     /* column properties */
    storage_tab_owner varchar2(30),               /* non-image coll/substitutable col */
    storage_tab_name  varchar2(30),               /* non-image coll/substitutable col */
    sna_type_oid      raw(16),                   /* type OID for snapshot column */
    sna_type_hashcode raw(17),                   /* type hashcode for snapshot column */
    sna_type_owner    varchar2(30),               /* type owner for snapshot column */
    sna_type_name     varchar2(30),               /* type name for snapshot column */
)

```



```

mas_type_oid      raw(16),                /* type OID for master column */
mas_type_hashcode raw(17),                /* type hashcode for master column */
mas_type_owner    varchar2(30),           /* type owner for master column */
mas_type_name     varchar2(30)            /* type name for master column*/
)
/

create unique index i_snap_objcoll on
snap_objcoll$(sowner, vname, instsite, tabnum, snacol)
/

REM add sub_handle and change_view to sys.snap_reftime$
alter table sys.snap_reftime$
add
(
sub_handle      number,                  /* subscription handle (if using CDC) */
change_view     varchar2(30)             /* change view name (if using CDC) */
)
/

REM set default values for snap_reftime$.sub_handle and change_view
update sys.snap_reftime$
set sub_handle = 0, change_view = NULL
/

REM add oldest_oid and oldest_new to sys.mlog$
alter table mlog$
add
(
oldest_oid      date,                   /* maximum age of OID information in the log */
oldest_new      date                    /* maximum age of new values in the log */
)
/

REM set default value for mlog$.oldest_oid
update mlog$
set oldest_oid = to_date('4000-01-01:00:00:00', 'YYYY-MM-DD:HH24:MI:SS')
/

REM add the detaileut column to sys.sumdetail$
ALTER TABLE sys.sumdetail$
ADD
(
detaileut      number                    /* detail tablew EUT flag */
)
/

REM add the detailcolfunction column to sys.sumkey$
ALTER TABLE sys.sumkey$
ADD
(
detailcolfunction number                /* 0=regular,1=partition key,2=part. marker */
)
/

REM add the nodetype column to sys.sumkey$. Indicates type of GSet node.
ALTER TABLE sys.sumkey$
ADD
(

```

```

        nodetype number /* 0-none, 1-gset, 2-rollup, 3-cube, 4-ccol, 5-cgset, 6-opn*/
    )
/

REM add the ordinalpos column to sys.sumkey$. Used for grouping sets.
ALTER TABLE sys.sumkey$
ADD
(
    ordinalpos number /* ordinal position within grouping set hierarchy */
)
/

REM add the parentpos column to sys.sumkey$. Used for grouping sets.
ALTER TABLE sys.sumkey$
ADD
(
    parentpos number /* parent position within grouping set hierarchy */
)
/

Rem Bug 2328821 : Recreate index on sumkey$
BEGIN
    EXECUTE IMMEDIATE 'DROP INDEX i_sumkey$_1';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -1418 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'CREATE UNIQUE INDEX i_sumkey$_1 ON ' ||
        'sumkey$(sumobj#, sumcolpos#, groupingpos#, ordinalpos)';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -942 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/

REM add the agginfo column to sys.sumagg$. Records start of arguments, etc.
ALTER TABLE sys.sumagg$
ADD
(
    agginfo          varchar2(2000) /* info about aggs like start of args */
)
/

REM add the agginfolen column to sys.sumagg$. Records start of arguments, etc.
ALTER TABLE sys.sumagg$
ADD
(
    agginfolen      number /* length of agginfo */
)
/

```

REM FOR materialized aggregate views, SET the KKZFUSE bit so that they will  
REM be revalidated during the first refresh.

REM Do this only if KKZFAGG bit (0x00001000) is set and KKZFUSE bit  
REM (0x00000008) is not already set.

```
UPDATE sys.snap$ SET flag = flag + 8
WHERE bitand(flag, 4096) = 4096
      AND bitand(flag, 8) = 0
/
```

```
Rem =====
Rem END materialized views upgrade
Rem =====
```

```
REM
REM ===== BEGIN OF OUTLN tables upgrade =====
REM
```

```
ALTER TABLE outln.ol$
add
(
  hash_value2      number, /* hash value on sql_text stripped of whitespace */
  spare1           number,          /* spare column */
  spare2           varchar2(1000)    /* spare column */
)
/
```

```
ALTER TABLE outln.ol$hints
add
(
  ref_id           number,          /* node id that this hint is referencing */
  user_table_name  varchar2(64),    /* table name to which this hint applies */
                                   /* this field also contains the schema */
                                   /* name to which the table belongs to */
  cost             double precision, /* optimizer estimated cost of the
                                   hinted operation */
  cardinality      double precision, /* optimizer estimated cardinality
                                   of the hinted operation */
  bytes            double precision, /* optimizer estimated byte count
                                   of the hinted operation */
  hint_textoff     number,          /* offset into the SQL statement to
                                   which this hint applies */
  hint_textlen     number,          /* length of SQL to which this hint applies */
  join_pred        varchar2(2000),  /* join predicate (applies only for
                                   join method hints) */
  spare1           number,          /* spare number for future enhancements */
  spare2           number           /* spare number for future enhancements */
)
/
```

```
create table outln.ol$nodes
(
  ol_name          varchar2(30),      /* outline name */
  category         varchar2(30),      /* outline category */
  node_id          number,            /* qbc node identifier */
  parent_id        number,            /* node id of the parent node for current node */
  node_type        number,            /* qbc node type */
  node_textlen     number,            /* length of SQL to which this node applies */
  node_textoff     number             /* offset into the SQL statement to which this
```

```

node applies */

)
/

REM ===== END OF outln tables upgrade =====

REM moved SQLJ upgrades to i0801070.sql

REM =====
REM START OF opaque type updates
REM =====

REM moved Opaque Type creates to i0801070.sql

/* upgrade existing opaque types set the opqtype flag and remove adt flag */
update sys.coltype$ ccol set flags = flags + 16384 - 2
where bitand(flags,2) = 2 and
exists (select null from sys.col$ c where c.obj# = ccol.obj# and
c.intcol# = ccol.intcol# and c.type# = 58);

insert into opqtype$ (obj#,intcol#,type,flags, lobcol, objcol, extracol,
schemaid, elemnum)
select obj#, intcol#, 0, 0, 0, 0, 0, null, 0 from
sys.col$ c where c.type# = 58;

REM =====
REM END OF opaque type updates
REM =====

REM =====
REM BEGIN OF notification table upgrade
REM =====
ALTER TABLE reg$
add
(
status number
)
/
REM =====
REM END of notification table upgrade
REM =====

rem
rem ===== BEGIN Security Features Upgrade
rem
rem Create the proxy_data$ using the data in the 8.1.7 proxy$ table. Since
rem different types of authentication are not allowed in 8.1.7
create table proxy_data$
( 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 credential
* 1 = Certificate
* 2 = Distinguished Name
* 3 = Oracle password
* 4 = Kerberos ticket

```

```

        */
credential_version# NUMBER NOT NULL,    /* Version number of the credential */
        /*
        * Values
        * 0 = no version
        * If certificate:
        * 1 = X.509 V3
        * if Kerberos ticket
        * 1 = Beta 5 release 2
        */
credential_minor# NUMBER NOT NULL,      /* Minor credential version number */
        /*
        * Values
        * 0 = no version
        * If certificate:
        * 1 = V3
        */
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
        */
)
/
rem In 9.0, the client and user data has been split from the role data for each
rem pair.
insert into proxy_data$
select client#, proxy#,
       0 credential_type#,
       0 credential_version#,
       0 credential_minor#,
       flags
from ( select distinct client#, proxy#, flags
      from proxy$ )
/
create unique index i_proxy_data$ on proxy_data$(client#, proxy#)
/

rem Create the role table.
create table proxy_role_data$
as
select distinct client#, proxy#, role#
  from proxy$
/
create index i_proxy_role_data$_1 on
  proxy_role_data$(client#, proxy#)
/
create unique index i_proxy_role_data$_2 on
  proxy_role_data$(client#, proxy#, role#)
/

rem Empty the proxy$ table. It is not possible to drop it because the table
rem is cached.
delete from proxy$
/
COMMIT

```

[illegible]

[illegible]

```

(
    bhiboundval    blob    /* partition key in binary (linear key) format */
)
/
alter table tabcompart$
add
(
    bhiboundval    blob    /* partition key in binary (linear key) format */
)
/
alter table indpart$
add
(
    bhiboundval    blob    /* partition key in binary (linear key) format */
)
/
alter table indcompart$
add
(
    bhiboundval    blob    /* partition key in binary (linear key) format */
)
/
REM ===== end of upgrade for partition feature =====
Rem=====

Rem=====
Rem BEGIN OF external tables meta data
Rem=====
Rem
Rem External tables meta data: external_tab$, external_location$
Rem (extracted from sql.bsq)
Rem
create table external_tab$
( obj#          number not null,          /* base table object number */
  default_dir   varchar2(30) not null,    /* default directory */
  type$         varchar2(30) not null,    /* access driver type */
  nr_locations  number          not null, /* number of locations */
  reject_limit  number          not null, /* reject limit */
  par_type      number not null, /* access parameter type: blob=1, clob=2 */
  param_clob    clob,           /* access parameters in clob form */
  param_blob    blob)          /* access parameters in blob form */
/
create unique index i_external_tab1$ on external_tab$(obj#)
/

create table external_location$
(
    obj#          number not null,          /* base table object number */
    position      number not null,          /* this location index */
    dir           varchar2(30),             /* location directory object */
    name          varchar2(4000)           /* location name */
)
/
create unique index i_external_location1$ on external_location$(obj#, position)
/

Rem=====
Rem END OF external tables meta data

```



Rem=====

Rem =====

Rem seq\$ column addition

Rem =====

ALTER TABLE seq\$

add

(

    flags number /\* sequence bit flags \*/

)

/

Rem=====

Rem                      Temporal Access

Rem=====

create cluster smon\_scn\_to\_time (

    thread number                      /\* the thread number \*/

)

/

create index smon\_scn\_to\_time\_idx on cluster smon\_scn\_to\_time

/

create table smon\_scn\_time (

    thread number,                      /\* the thread number \*/

    time\_mp number,                      /\* time this recent scn represents \*/

    time\_dp date,                      /\* time converted into oracle date \*/

    scn\_wrp number,                      /\* scn.wrp \*/

    scn\_bas number                      /\* scn.bas \*/

) cluster smon\_scn\_to\_time (thread)

/

Rem=====

Rem BEGIN AQ changes

Rem=====

ALTER TABLE sys.aq\$\_message\_types

ADD (properties NUMBER,                      /\* properties \*/

    trans\_name VARCHAR2(61))                      /\* transformation to be applied \*/

/

ALTER TABLE sys.aq\$\_message\_types

DROP CONSTRAINT aq\$\_msgtypes\_primary

/

ALTER TABLE sys.aq\$\_message\_types

ADD CONSTRAINT aq\$\_msgtypes\_unique

UNIQUE (queue\_oid, schema\_name, queue\_name, destination, trans\_name)

/

ALTER TABLE sys.aq\$\_message\_types

MODIFY queue\_oid NOT NULL

/

ALTER TABLE sys.aq\$\_message\_types

MODIFY schema\_name NOT NULL

/

```

ALTER TABLE sys.aq$_message_types
MODIFY queue_name NOT NULL
/

ALTER TABLE sys.aq$_message_types
MODIFY destination NOT NULL
/

-- Reorder the unique constraint index on system.aq$_queues
ALTER TABLE system.aq$_queues
    DROP CONSTRAINT aq$_queues_check
/

ALTER TABLE system.aq$_queues
    ADD CONSTRAINT aq$_queues_check UNIQUE(name, table_objno)
/

Rem=====
Rem  END AQ changes
Rem=====

Rem=====
Rem  BEGIN 9.0 IAS changes to replication tables
Rem=====

create table system.repcat$_template_status
(template_status_id number,
    constraint repcat$_template_status_pk primary key (template_status_id),
    status_type_name varchar2(100) not null)
/

insert into system.repcat$_template_status
(template_status_id,status_type_name)
select 0, 'Modifiable'
from dual
where not exists
(select 1 from system.repcat$_template_status
where template_status_id = 0)
/

insert into system.repcat$_template_status
(template_status_id,status_type_name)
select 1, 'Frozen'
from dual
where not exists
(select 1 from system.repcat$_template_status
where template_status_id = 1)
/

insert into system.repcat$_template_status
(template_status_id,status_type_name)
select 2, 'Deleted'
from dual
where not exists
(select 1 from system.repcat$_template_status
where template_status_id = 2)
/

```

```

create table system.repcat$_template_types
(template_type_id number,
    constraint repcat$_template_types_pk primary key (template_type_id),
template_description varchar2(200),
flags raw(255),
spare1 varchar2(4000))
/

```

```

Rem seed data for repcat$_template_types
insert into system.repcat$_template_types
(template_type_id, template_description, flags)
select 1, 'Deployment template', hextoraw('01')
from dual
where not exists
    (select 1 from system.repcat$_template_types
    where template_type_id = 1)
/

```

```

insert into system.repcat$_template_types
(template_type_id, template_description, flags)
select 2, 'IAS template', hextoraw('02')
from dual
where not exists
    (select 1 from system.repcat$_template_types
    where template_type_id = 2)
/

```

```

alter table system.repcat$_refresh_templates
add (refresh_group_id number default 0 not null,
    template_type_id number default 1 not null,
    constraint repcat$_refresh_templates_fk1 foreign key (template_type_id)
        references system.repcat$_template_types,
template_status_id number default 0 not null,
    constraint repcat$_refresh_templates_fk2 foreign key (template_status_id)
        references system.repcat$_template_status,
flags raw(255),
spare1 varchar2(4000))
/

```

```

BEGIN
EXECUTE IMMEDIATE 'create index system.repcat$_user_authorizations_n1 on ' ||
    ' system.repcat$_user_authorizations(refresh_template_id)';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -942 THEN NULL;
        ELSE RAISE;
    END IF;
END;
/

```

```

create table system.repcat$_object_types
(object_type_id number,
    constraint repcat$_object_type_pk primary key (object_type_id),
object_type_name varchar2(200),
flags raw(255),
spare1 varchar2(4000))
/

```

```
REM seed data for system.repcat$_object_types
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select -1017,'GENERATED DDL',hextoraw('02')
from dual
where not exists
```

```
(select 1 from system.repcat$_object_types
where object_type_id = -1017)
```

```
/
```

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select -1016,'DUMMY MATERIALIZED VIEW',hextoraw('02')
from dual
where not exists
```

```
(select 1 from system.repcat$_object_types
where object_type_id = -1016)
```

```
/
```

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select -1015,'UPDATABLE MATERIALIZED VIEW LOG',hextoraw('02')
from dual
where not exists
```

```
(select 1 from system.repcat$_object_types
where object_type_id = -1015)
```

```
/
```

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select -1014,'REFRESH GROUP',hextoraw('02')
from dual
where not exists
```

```
(select 1 from system.repcat$_object_types
where object_type_id = -1014)
```

```
/
```

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select -1013,'SYNCHRONOUS MASTER REPGROUP',hextoraw('02')
from dual
where not exists
```

```
(select 1 from system.repcat$_object_types
where object_type_id = -1013)
```

```
/
```

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select -1012,'ASYNCHRONOUS MASTER REPGROUP',hextoraw('02')
from dual
where not exists
```

```
(select 1 from system.repcat$_object_types
where object_type_id = -1012)
```

```
/
```

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
```

```

select -1011, 'TEMPORARY TABLE', hextoraw('02')
from dual
where not exists
    (select 1 from system.repcat$_object_types
     where object_type_id = -1011)
/
insert into system.repcat$_object_types
(object_type_id, object_type_name, flags)
select -1005, 'SYNCHRONOUS UPDATABLE TABLE', hextoraw('02')
from dual
where not exists
    (select 1 from system.repcat$_object_types
     where object_type_id = -1005)
/

insert into system.repcat$_object_types
(object_type_id, object_type_name, flags)
select -1004, 'ASYNCHRONOUS UPDATABLE TABLE', hextoraw('00')
from dual
where not exists
    (select 1 from system.repcat$_object_types
     where object_type_id = -1004)
/

insert into system.repcat$_object_types
(object_type_id, object_type_name, flags)
select -1003, 'READ ONLY TABLE', hextoraw('02')
from dual
where not exists
    (select 1 from system.repcat$_object_types
     where object_type_id = -1003)
/

insert into system.repcat$_object_types
(object_type_id, object_type_name, flags)
select -1002, 'SITEOWNER', hextoraw('02')
from dual
where not exists
    (select 1 from system.repcat$_object_types
     where object_type_id = -1002)
/

insert into system.repcat$_object_types
(object_type_id, object_type_name, flags)
select -1001, 'USER', hextoraw('02')
from dual
where not exists
    (select 1 from system.repcat$_object_types
     where object_type_id = -1001)
/

insert into system.repcat$_object_types
(object_type_id, object_type_name, flags)
select -5, 'DATABASE LINK', hextoraw('01')
from dual
where not exists
    (select 1 from system.repcat$_object_types
     where object_type_id = -5)

```

/

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select -1,'MATERIALIZED VIEW',hextoraw('01')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = -1)
```

/

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 1,'INDEX',hextoraw('01')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 1)
```

/

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 2,'TABLE',hextoraw('01')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 2)
```

/

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 4,'VIEW',hextoraw('03')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 4)
```

/

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 5,'SYNONYM',hextoraw('01')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 5)
```

/

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 6,'SEQUENCE',hextoraw('03')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 6)
```

/

```
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
```

```

select 7,'PROCEDURE',hextoraw('03')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 7)
/

```

```

insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 8,'FUNCTION',hextoraw('03')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 8)
/

```

```

insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 9,'PACKAGE',hextoraw('03')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 9)
/

```

```

insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 10,'PACKAGE BODY',hextoraw('01')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 10)
/

```

```

insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 12,'TRIGGER',hextoraw('01')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 12)
/

```

```

insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 13,'TYPE',hextoraw('01')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 13)
/

```

```

insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 14,'TYPE BODY',hextoraw('01')
from dual
where not exists
      (select 1 from system.repcat$_object_types
       where object_type_id = 14)
/

```

```

insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 32,'INDEX TYPE',hextoraw('01')
from dual
where not exists
(select 1 from system.repcat$_object_types
where object_type_id = 32)
/
insert into system.repcat$_object_types
(object_type_id,object_type_name, flags)
select 33,'OPERATOR',hextoraw('01')
from dual
where not exists
(select 1 from system.repcat$_object_types
where object_type_id = 33)
/

create table system.repcat$_template_refgroups
(refresh_group_id number not null,
constraint repcat$_template_refgroups_pk primary key (refresh_group_id),
refresh_group_name varchar2(30) not null,
refresh_template_id number not null,
constraint repcat$_template_refgroups_fkl foreign key (refresh_template_id)
references system.repcat$_refresh_templates on delete cascade,
rollback_seg varchar2(30),
start_date varchar2(200),
interval varchar2(200))
/

create sequence system.repcat$_template_refgroups_s
/

BEGIN
EXECUTE IMMEDIATE 'create index system.repcat$_template_refgroups_n1 on ' ||
' system.repcat$_template_refgroups(refresh_group_name)';
EXCEPTION
WHEN OTHERS THEN
IF SQLCODE = -942 THEN NULL;
ELSE RAISE;
END IF;
END;
/

BEGIN
EXECUTE IMMEDIATE 'create index system.repcat$_template_refgroups_n2 on ' ||
' system.repcat$_template_refgroups(refresh_template_id)';
EXCEPTION
WHEN OTHERS THEN
IF SQLCODE = -942 THEN NULL;
ELSE RAISE;
END IF;
END;
/

alter table system.repcat$_template_objects
add ( constraint repcat$_template_objects_fk3 foreign key (object_type)
references system.repcat$_object_types,
template_refgroup_id number default 0 not null,

```



```

        flags raw(255),
        spare1 varchar2(4000)
    )
/

BEGIN
    EXECUTE IMMEDIATE 'create index system.repcat$_object_parms_n2 on ' ||
        ' system.repcat$_object_parms(template_object_id)';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -942 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/

create table system.repcat$_site_objects
(template_site_id number not null,
    constraint repcat$_site_object_fk2 foreign key (template_site_id)
        references system.repcat$_template_sites on delete cascade,
    sname varchar2(30),
    oname varchar2(30) not null,
    object_type_id number not null,
    constraint repcat$_site_objects_fk1 foreign key (object_type_id)
        references system.repcat$_object_types,
    constraint repcat$_site_objects_u1 unique
        (template_site_id, oname, object_type_id, sname))
/

BEGIN
    EXECUTE IMMEDIATE 'create index system.repcat$_site_objects_n1 on ' ||
        ' system.repcat$_site_objects(template_site_id)';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -942 THEN NULL;
        ELSE RAISE;
        END IF;
END;
/

create table system.repcat$_exceptions
(exception_id NUMBER,
    constraint repcat$_exceptions_pk primary key (exception_id),
    user_name varchar2(30),
    request clob,
    job number,
    error_date date,
    error_number number,
    error_message varchar2(4000),
    line_number number)
/

create sequence system.repcat$_exceptions_s
/

create table system.repcat$_instantiation_ddl
(refresh_template_id number,
    constraint repcat$_instantiation_ddl_fk1 foreign key (refresh_template_id)

```

```

        references system.repcat$_refresh_templates on delete cascade,
ddl_text clob,
ddl_num number,
phase number,
constraint repcat$_instantiation_ddl_pk primary key
    (refresh_template_id, phase, ddl_num))
/

```

```

ALTER TABLE system.repcat$_refresh_templates
    DROP CONSTRAINT refresh_templates_c1
/

```

```

ALTER TABLE system.repcat$_refresh_templates
    ADD CONSTRAINT refresh_templates_c1 CHECK
    ((public_template in ('Y','N')) or public_template is NULL)
/

```

```

ALTER TABLE system.repcat$_template_objects
    DROP CONSTRAINT repcat$_template_objects_c1
/

```

```

ALTER TABLE system.repcat$_template_sites
    DROP CONSTRAINT repcat$_template_sites_c1
/

```

```

ALTER TABLE system.repcat$_template_sites
    ADD CONSTRAINT repcat$_template_sites_c1 CHECK
    (status in (-1,0,1))
/

```

```

Rem=====
Rem  END 9.0 IAS changes to replication tables
Rem=====

```

```

REM =====
REM BEGIN character limit updates
REM =====

```

Rem This script fills in default values for required 9.0 information that was  
Rem left blank in 8.1. In 9.0 every string column must have a character limit  
Rem as well as a byte limit, and string columns must know what character set  
Rem they use. There was no concept of character limits in 8.1. Only nchar,  
Rem nvarchar2, nclob were required to know what character set they used.

```

update col$ set charsetid =
    (select max(charsetid) from col$ where charsetform = 1), charsetform = 1
    where charsetform = 0 and type# in (1, 8, 96, 112);
update argument$ set charsetid =
    (select max(charsetid) from col$ where charsetform = 1), charsetform = 1
    where charsetform = 0 and type# in (1, 8, 96, 112);
update collection$ set charsetid =
    (select max(charsetid) from col$ where charsetform = 1), charsetform = 1
    where charsetform = 0 and elem_toid in
    (select toid from type$ where typecode in (1, 8, 96, 112));
update attribute$ set charsetid =
    (select max(charsetid) from col$ where charsetform = 1), charsetform = 1
    where charsetform = 0 and attr_toid in
    (select toid from type$ where typecode in (1, 8, 96, 112));

```

```

update parameter$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 1)
  where charsetform = 0 and toid in
  (select toid from type$ where typecode in (1, 8, 96, 112));
update result$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 1)
  where charsetform = 0 and toid in
  (select toid from type$ where typecode in (1, 8, 96, 112));

update col$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 1)
  where charsetid = 0 and charsetform = 1;
update argument$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 1)
  where charsetid = 0 and charsetform = 1;
update collection$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 1)
  where charsetid = 0 and charsetform = 1;
update attribute$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 1)
  where charsetid = 0 and charsetform = 1;
update parameter$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 1)
  where charsetid = 0 and charsetform = 1;
update result$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 1)
  where charsetid = 0 and charsetform = 1;

update col$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 2)
  where charsetid = 0 and charsetform = 2;
update argument$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 2)
  where charsetid = 0 and charsetform = 2;
update collection$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 2)
  where charsetid = 0 and charsetform = 2;
update attribute$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 2)
  where charsetid = 0 and charsetform = 2;
update parameter$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 2)
  where charsetid = 0 and charsetform = 2;
update result$ set charsetid =
  (select max(charsetid) from col$ where charsetform = 2)
  where charsetid = 0 and charsetform = 2;

update col$ set spare3 = length
  where charsetform = 1 and length > 0
  and (spare3 is null or spare3 = 0) and type# in (1, 96);
update col$ set spare3 = 1
  where charsetform = 2 and length > 0
  and (spare3 is null or spare3 = 0) and type# in (1, 96);

commit;

```

```

REM =====
REM END character limit updates

```

```

REM =====

REM =====
REM BEGIN Fix NCHAR columns - force to UTF8 or AL16UTF16
REM =====

REM Save original NCHAR character set in props$ NLS_OLD_NCHAR_CS

DECLARE
    nchar_cset      VARCHAR2(30);
    prop_count      NUMBER;
BEGIN

-- Only store/set NCHAR CS value if no value exists in props table
select count(*) into prop_count from sys.props$
where name in ('NLS_OLD_NCHAR_CS', 'NLS_SAVED_NCHAR_CS');

if prop_count = 0 then
-- Get NLS NCHAR CS value and store in props$
select value into nchar_cset from v$nls_parameters
where parameter='NLS_NCHAR_CHARACTERSET';
insert into props$ (name, value$)
values ('NLS_OLD_NCHAR_CS', nchar_cset);

if nchar_cset != 'UTF8' then
    nchar_cset := 'AL16UTF16';
end if;

-- Insert is committed even if ALTER fails, so delete in exception block
begin
    execute immediate
        'ALTER DATABASE NATIONAL CHARACTER SET internal_use ' || nchar_cset;
    commit;
exception
    when others then
        delete from props$ where name= 'NLS_OLD_NCHAR_CS';
        commit;
        raise;
end;
end if;
END;
/

REM Convert two replication columns to new National Character Set

ALTER TABLE system.repcat$_priority MODIFY nchar_value NCHAR(500);
ALTER TABLE system.repcat$_priority MODIFY nvarchar2_value NVARCHAR2(1000);
ALTER TABLE system.def$_lob MODIFY nclob_col NCLOB;
ALTER TABLE system.def$_temp$lob MODIFY temp$nclob NCLOB;

REM =====
REM End Fix NCHAR columns - force to UTF8 or AL16UTF16
REM =====

REM =====
REM BEGIN Revoke pre-9i DBSNMP priveleges
REM =====

```

```

BEGIN
    EXECUTE IMMEDIATE
        'REVOKE ALL PRIVILEGES from DBSNMP';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE IN (-1917, -1918, -1919, -1951, -1952) THEN NULL;
        ELSE RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE
        'REVOKE CONNECT, RESOURCE from DBSNMP';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE IN (-1917, -1918, -1919, -1951, -1952) THEN NULL;
        ELSE RAISE;
        END IF;
END;
/

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

REM =====
REM END Revoke pre-9i DBSNMP priveleges
REM =====

REM =====
REM BEGIN Set the created by constraint property flag for all unique indices
REM =====

update ind$ set property = property+4096 where bitand(property, 4096) = 0
and bitand(property, 1) <> 0;

commit;

REM =====
REM END Set the created by constraint property flag for all unique indices
REM =====

Rem=====
Rem END STAGE 1: upgrade from 8.1.7 to 9.0.1
Rem=====

Rem=====
Rem BEGIN STAGE 2: upgrade from 9.0.1 to the new release
Rem=====

```

@@c0900010

```
Rem=====
Rem END STAGE 2: upgrade from 9.0.1 to the new release
Rem=====
```

```
Rem*****
Rem END c0801070.sql
Rem*****
```