

```

CONNECT / AS SYSDBA
DROP USER MY_ORACLE_USR CASCADE;
CREATE USER MY_ORACLE_USR IDENTIFIED BY "oracle"
  DEFAULT TABLESPACE USERS
  TEMPORARY TABLESPACE TEMP
;
CREATE ROLE MY_ORACLE_ROLE;
GRANT CREATE SESSION TO MY_ORACLE_ROLE;
GRANT CREATE TABLE TO MY_ORACLE_ROLE;
GRANT CREATE SEQUENCE TO MY_ORACLE_ROLE;
GRANT CREATE PROCEDURE TO MY_ORACLE_ROLE;
GRANT CREATE VIEW TO MY_ORACLE_ROLE;
GRANT CREATE TRIGGER TO MY_ORACLE_ROLE;
GRANT CREATE MATERIALIZED VIEW TO MY_ORACLE_ROLE;
GRANT CREATE TYPE TO MY_ORACLE_ROLE;
GRANT CREATE ANY DIRECTORY TO MY_ORACLE_ROLE;
GRANT MY_ORACLE_ROLE TO MY_ORACLE_USR;
ALTER USER MY_ORACLE_USR QUOTA UNLIMITED ON USERS;
CONNECT my_oracle_usr/oracle
-- create a type for referencing it in a column definition later
CREATE OR REPLACE TYPE MY_ORACLE_TYPE AS VARRAY(100) OF VARCHAR2(100);
/
-- create sequences, a master and a detail table, primary keys, a unique constraint,
-- a foreign constraint and a check constraint
-- define columns as NUMBER, VARCHAR2, DATE, TIMESTAMP, VARRAY, CLOB, BLOB, INTERVAL to have
-- a wide range of different column definitions
CREATE SEQUENCE MY_ORACLE_SEQ_1;
CREATE SEQUENCE MY_ORACLE_SEQ_2 NOCACHE;
CREATE TABLE T_MASTER ( MASTER_ID NUMBER NOT NULL
  , MASTER_DESCR VARCHAR2(200) NOT NULL
  , MASTER_DATE DATE NOT NULL
  , MASTER_INT INTERVAL DAY(3) TO SECOND NOT NULL
  , MASTER_TS TIMESTAMP
  , MASTER_AR MY_ORACLE_TYPE
  , CONSTRAINT T_MASTER_PK PRIMARY KEY ( MASTER_ID )
  , CONSTRAINT T_MASTER_CHK1 CHECK ( LENGTH(MASTER_DESCR) > 10 )
  , CONSTRAINT T_MASTER_UK1 UNIQUE ( MASTER_DATE )
) TABLESPACE USERS;
CREATE TABLE T_DETAIL ( DETAIL_ID NUMBER NOT NULL
  , MASTER_ID NUMBER NOT NULL
  , DETAIL_CLOB CLOB
  , DETAIL_BLOB BLOB
  , CHANGE_DATE DATE
  , CONSTRAINT T_DETAIL_PK PRIMARY KEY ( DETAIL_ID )
  , CONSTRAINT T_DETAIL_FK FOREIGN KEY ( MASTER_ID ) REFERENCES T_MASTER(MASTER_ID)
) TABLESPACE USERS;
-- add a btree index
CREATE INDEX MY_ORA_IDX1 ON T_DETAIL(MASTER_ID);
-- add a bitmap index
CREATE BITMAP INDEX MY_ORA_IDX2 ON T_DETAIL(CHANGE_DATE);
-- add a trigger
CREATE TRIGGER TRG_T_DETAIL BEFORE INSERT OR UPDATE ON T_DETAIL FOR EACH ROW
DECLARE
BEGIN
  :new.change_date := sysdate;
END TRG_T_DETAIL;
/
-- prepare a binary file ( cp in this case ) for loading and a directory
!cp $ORACLE_HOME/oc4j/j2ee/oc4j_applications/applications/em/em/images/oracleLogo.gif /tmp
CREATE OR REPLACE DIRECTORY MY_DIR AS '/tmp';
-- add some sample data ( this will generate around )
DECLARE
  ln_master_key t_master.master_id%TYPE;
BEGIN
  FOR i IN 1..10000
  LOOP
    ln_master_key := my_oracle_seq_1.nextval;

```

```

INSERT INTO T_MASTER ( MASTER_ID
                      , MASTER_DESCR
                      , MASTER_DATE
                      , MASTER_INT
                      , MASTER_TS
                      , MASTER_AR
                      )
VALUES ( ln_master_key
       , 'DESCRIPTION '||i
       , SYSDATE+i
       , INTERVAL '10 00:00:00' DAY TO SECOND
       , SYSTIMESTAMP
       , my_oracle_type ( lpad(i,90,'A'), lpad(i,90,'B'), lpad(i,90,'C'), lpad(i,90,'D') )
       );

DECLARE
  src_lob BFILE := BFILENAME('MY_DIR','oracleLogo.gif');
  dest_lob BLOB;
BEGIN
  INSERT INTO T_DETAIL ( DETAIL_ID
                       , MASTER_ID
                       , DETAIL_CLOB
                       , DETAIL_BLOB
                       )
    VALUES ( my_oracle_seq_2.nextval
           , ln_master_key
           , 'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA'
           , EMPTY_BLOB()
           )
    RETURNING DETAIL_BLOB INTO dest_lob;
  dbms_lob.open ( src_lob, dbms_lob.lob_readonly);
  dbms_lob.loadfromfile ( dest_lob => dest_lob
                        , src_lob => src_lob
                        , amount => dbms_lob.getlength(src_lob)
                        );
  dbms_lob.close(src_lob);
  COMMIT;
END;
END LOOP;
COMMIT;
END;
/
-- just to verify the amount the of rows
SELECT COUNT(*)
  FROM T_MASTER;
SELECT COUNT(*)
  FROM T_DETAIL;
-- create a sample package
CREATE OR REPLACE PACKAGE MY_ORA_PKG1
AS
  TYPE rec1 IS RECORD ( a NUMBER
                       , b DATE
                       );
  TYPE tab1 IS TABLE OF rec1 INDEX BY PLS_INTEGER;
  PROCEDURE P1 ( pn_detail_id IN t_detail.detail_id%TYPE
               , pc_detail_clob IN CLOB
               );
  PROCEDURE P2 ( pn_master_id IN PLS_INTEGER
               , pn_master_ar IN MY_ORACLE_TYPE
               );
  FUNCTION F1 RETURN PLS_INTEGER;
END MY_ORA_PKG1;
/
CREATE OR REPLACE PACKAGE BODY MY_ORA_PKG1
IS
  PROCEDURE P1 ( pn_detail_id IN t_detail.detail_id%TYPE
               , pc_detail_clob IN CLOB
               )

```

```

IS
BEGIN
  IF
    pn_detail_id IS NOT NULL
    AND
    pc_detail_clob IS NOT NULL
  THEN
    UPDATE t_detail
      SET detail_clob = pc_detail_clob
      WHERE detail_id = pn_detail_id;
  END IF;
END P1;
PROCEDURE P2 ( pn_master_id IN PLS_INTEGER
              , pn_master_ar IN MY_ORACLE_TYPE
              )
IS
BEGIN
  UPDATE t_master
    SET master_ar = pn_master_ar
      , master_date = SYSDATE
    WHERE master_id = pn_master_id
  ;
END P2;
FUNCTION F1 RETURN PLS_INTEGER
IS
BEGIN
  RETURN MOD(99,4);
END F1;
BEGIN
  NULL;
END MY_ORA_PKG1;
/
-- check if the package works
exec my_ora_pkg1.p1 ( 55, '#####');
exec my_ora_pkg1.p2 ( 44, my_oracle_type('#####', '#####'));
select my_ora_pkg1.f1 from dual;
-- create a sample view
CREATE OR REPLACE VIEW MY_ORA_VIEW ( "MY_DET_ID", "MY_MASTER_ID", "MY_CHANGE_DATE" )
AS SELECT DETAIL_ID, MASTER_ID, CHANGE_DATE
   FROM t_detail
   ORDER BY 2;
SELECT count(*)
   FROM my_ora_view;
-- create a sample mview
CREATE MATERIALIZED VIEW MY_ORA_MVIEW ( "MY_DET_ID", "MY_MASTER_ID", "MY_CHANGE_DATE" )
AS SELECT DETAIL_ID, MASTER_ID, CHANGE_DATE
   FROM t_detail
   ORDER BY 2;
SELECT COUNT(*)
   FROM MY_ORA_MVIEW;

-- partitioning
CREATE TABLE MY_PTAB1 ( id NUMBER
                       , when DATE
                       )
  PARTITION BY RANGE ( when )
    ( PARTITION p1 VALUES LESS THAN ( to_date ( '01.01.2012', 'DD.MM.YYYY' ) )
      , PARTITION p2 VALUES LESS THAN ( to_date ( '01.02.2012', 'DD.MM.YYYY' ) )
      , PARTITION p3 VALUES LESS THAN ( to_date ( '01.03.2012', 'DD.MM.YYYY' ) )
      , PARTITION p4 VALUES LESS THAN ( to_date ( '01.04.2012', 'DD.MM.YYYY' ) )
      , PARTITION p5 VALUES LESS THAN ( to_date ( '01.05.2012', 'DD.MM.YYYY' ) )
      , PARTITION p6 VALUES LESS THAN ( to_date ( '01.06.2012', 'DD.MM.YYYY' ) )
    )
  TABLESPACE USERS
  TABLESPACE USERS
  TABLESPACE USERS
  TABLESPACE USERS
  TABLESPACE USERS
  TABLESPACE USERS

```

```
TABLESPACE USERS
      , PARTITION p7 VALUES LESS THAN ( to_date ( '01.07.2012', 'DD.MM.YYYY' ) )
TABLESPACE USERS
      , PARTITION p8 VALUES LESS THAN ( to_date ( '01.08.2012', 'DD.MM.YYYY' ) )
TABLESPACE USERS
      , PARTITION p9 VALUES LESS THAN ( to_date ( '01.09.2012', 'DD.MM.YYYY' ) )
TABLESPACE USERS
      , PARTITION p10 VALUES LESS THAN ( to_date ( '01.10.2012', 'DD.MM.YYYY' ) )
TABLESPACE USERS
      , PARTITION p11 VALUES LESS THAN ( to_date ( '01.11.2012', 'DD.MM.YYYY' ) )
TABLESPACE USERS
      , PARTITION p12 VALUES LESS THAN ( to_date ( '01.12.2012', 'DD.MM.YYYY' ) )
TABLESPACE USERS
      , PARTITION p13 VALUES LESS THAN ( to_date ( '01.01.2013', 'DD.MM.YYYY' ) )
TABLESPACE USERS
      , PARTITION pmax VALUES LESS THAN ( maxvalue )
      )
TABLESPACE USERS;

DECLARE
  ld_begin DATE := to_date ( '31.12.2011', 'DD.MM.YYYY' );
BEGIN
  FOR i IN 1..465 LOOP
    INSERT INTO my_ptab1 ( id, when )
      VALUES ( i, ld_begin );
    ld_begin := ld_begin + 1;
  END LOOP;
END;
/
SELECT min(when)
      , max(when)
  FROM my_ptab1
;
SET LINES 164 PAGES 99
COL table_name FOR a30
COL partition_name FOR a30
COL tablespace_name FOR a30
SELECT table_name
      , partition_name
      , tablespace_name
  FROM user_tab_partitions
 WHERE table_name = 'MY_PTAB1'
 ORDER BY 1,2
;

EXIT;
```