

crearBD_directorio.sql

```
/* @crearBD_directorio */

SET SERVEROUTPUT ON

/* si ya existen las tablas, eliminarlas (primero la de documentos)
*/
DECLARE
    contador number;
BEGIN
    SELECT count(*) INTO contador FROM TAB WHERE TNAME = 'DOCUMENTO';
    IF contador > 0
        THEN
            DBMS_OUTPUT.PUT_LINE ('existe la tabla Documento -> se elimina');
            EXECUTE IMMEDIATE 'DROP TABLE Documento';
        ELSE
            DBMS_OUTPUT.PUT_LINE ('la tabla Documento no existe');
    END IF;
    SELECT count(*) INTO contador FROM TAB WHERE TNAME = 'DIRECTORIO';
    IF contador > 0
        THEN
            DBMS_OUTPUT.PUT_LINE ('existe la tabla Directorio -> se elimina');
            EXECUTE IMMEDIATE 'DROP TABLE Directorio';
        ELSE
            DBMS_OUTPUT.PUT_LINE ('la tabla "Directorio" no existe');
    END IF;
END;
/

CREATE TABLE Directorio (
    idDir    number(4) CONSTRAINT Directorio_PK PRIMARY KEY,
    nombre   varchar2(256) NOT NULL,
    dirPadre number(4));

/* la restriccion de clave ajena hay que añadirla después de que se haya creado la tabla */
ALTER TABLE Directorio
    ADD CONSTRAINT dirPadreDir_FK FOREIGN KEY (dirPadre) REFERENCES Directorio(idDir)
        ON DELETE CASCADE;

CREATE TABLE Documento (
    idDoc    number(4) CONSTRAINT Documento_PK PRIMARY KEY,
    nombre   varchar2(80) NOT NULL,
    dirPadre number(4) CONSTRAINT dirPadreDoc_FK REFERENCES Directorio(idDir)
        ON DELETE CASCADE);

COMMIT;
```

mostrarDatosBD_directorio.sql

```

/* @mostrarDatosBD_directorio */

SET LINESIZE 60
SET PAGESIZE 20

-- para que no muestre la salida generada por un fichero de órdenes
-- SET TERMOUT OFF

-- guardar la fecha en la variable fecha_informe
COLUMN fechaHoy NEW_VALUE fecha_informe

/* mostrar la fecha actual
*/
SELECT TO_CHAR(SYSDATE,'dd-Mon-yyyy') fechaHoy FROM DUAL;

BTITLE LEFT "fecha de informe: " fecha_informe

column idDoc          heading "id|doc"          format 999b;
column idDir          heading "id|dir"          format 999;
column nombre         heading "nombre"         format a30;
column dirPadre       heading "dir|Padre"       format b99;
column Path           heading "Path"           format a80;

/* se pueden mostrar mensajes directamente (desde un procedure, trigger, etc.) :
DBMS_OUTPUT.PUT_LINE('listado de DOCUMENTOS:'); */

TTITLE CENTER 'listado de DOCUMENTOS:' SKIP 2

-- espera confirmacion antes de mostrar una nueva página
-- set pause on

SELECT * FROM documento;

TTITLE LEFT 'listado de DIRECTORIOS:' SKIP 2

SELECT * FROM Directorio;

SET LINESIZE 132
TTITLE LEFT 'listado de todos DOCUMENTOS y su ubicación:' SKIP 2

SELECT idDoc, nombre, (SELECT SYS_CONNECT_BY_PATH(nombre, '/')
                       FROM Directorio WHERE idDir=D.dirPadre
                       START WITH dirPadre IS NULL
                       CONNECT BY dirPadre = PRIOR idDir) "Path"
FROM Documento D;

TTITLE OFF
BTITLE OFF

-- para que no espere al final de cada página
-- set pause off

```

ponerDatosBD_directorio.sql

```
/* @ponerDatosBD_directorio */

/* eliminar todos los datos previos
*/
DELETE FROM Documento;
DELETE FROM Directorio;

-- directorios
INSERT INTO Directorio (idDir, nombre, dirPadre) VALUES (1, 'BD2012', NULL);
INSERT INTO Directorio (idDir, nombre, dirPadre) VALUES (2, 'temas', 1);
INSERT INTO Directorio (idDir, nombre, dirPadre) VALUES (3, 'ejemplos', 1);
INSERT INTO Directorio (idDir, nombre, dirPadre) VALUES (4, 'trabajo', 1);
INSERT INTO Directorio (idDir, nombre, dirPadre) VALUES (5, 'E_R', 3);
INSERT INTO Directorio (idDir, nombre, dirPadre) VALUES (6, 'relacional', 3);
INSERT INTO Directorio (idDir, nombre, dirPadre) VALUES (7, 'SQL', 3);
INSERT INTO Directorio (idDir, nombre, dirPadre) VALUES (8, 'trigger', 7);
INSERT INTO Directorio (idDir, nombre, dirPadre) VALUES (9, 'SQL_PLUS', 7);

-- documentos
INSERT INTO Documento (idDoc, nombre, dirPadre) VALUES (1, 'tema1', 2);
INSERT INTO Documento (idDoc, nombre, dirPadre) VALUES (2, 'tema2', 2);
INSERT INTO Documento (idDoc, nombre, dirPadre) VALUES (3, 'recetas', 5);
INSERT INTO Documento (idDoc, nombre, dirPadre) VALUES (4, 'viajes', 5);
INSERT INTO Documento (idDoc, nombre, dirPadre) VALUES (5, 'autobuses', 6);
INSERT INTO Documento (idDoc, nombre, dirPadre) VALUES (6, 'ej1_trigger', 8);
INSERT INTO Documento (idDoc, nombre, dirPadre) VALUES (7, 'ej1_simple', 9);
INSERT INTO Documento (idDoc, nombre, dirPadre) VALUES (8, 'ej2_procedure', 9);
```

tstBD_directorio.sql

```
/* @tstBD_directorio */

SET ECHO ON
SET PAGESIZE 20
-- SET PAUSE "pulsar <CR> "

/* crea la BD de directorios, añade datos de prueba, y muestra la BD
después se ejecutan algunas operaciones con la BD (eliminar algun
subdirectorio, etc.)
Para terminar, se restaura el estado inicial
*/

-- creacion de la BD de directorios
@crearBD_directorio

PAUSE Press RETURN to continue

-- mostrar información de las tablas creadas
describe Directorio
describe Documento

-- la sentencia ACCEPT se utiliza para asignar valores a variables interactivamente
-- ACCEPT esperar CHAR PROMPT "pulsar <CR>"

-- PAUSE Press RETURN to continue

-- rellenar las tablas con los datos de ejemplo
@ponerDatosBD_directorio

PAUSE Press RETURN to continue

-- mostrar el contenido de la BD
@mostrarDatosBD_directorio

PAUSE Press RETURN to continue

-- eliminar el directorio especificado por el usuario.
DELETE FROM directorio WHERE idDir=&dir_a_borrar;

-- eliminar todo
DELETE FROM directorio WHERE idDir=1;

-- mostrar el contenido de la BD
@mostrarDatosBD_directorio

-- eliminar las tablas de la BD (por si acaso, primero la de Documento)
DROP TABLE Documento;
DROP TABLE Directorio;
```