

Luis Alfonso González de la Calzada y Adrián Díaz Jiménez

-- 0. Una sola vez por sesión (para que se vea salida por consola) --

SET serveroutput ON SIZE 100000;

-- 1. Crear archivos ASCII con los datos manualmente --

-- Están en codigos_postales.txt y domicilios.txt

-- 2. Crear tablas "Domicilios I" y "Códigos postales I" sin restricciones --

CREATE TABLE "Códigos postales I"

(
 "Código postal" CHAR(5),
 Población CHAR(50),
 Provincia CHAR(50)
);

-- Respuesta:

-- Table "Códigos postales I" creado.

CREATE table "Domicilios I"

(
 DNI CHAR(9),
 Calle CHAR(50),
 "Código postal" CHAR(5)
);

-- Respuesta:

-- Table "Domicilios I" creado.

-- 3. Importar con Oracle Loader las tablas del punto 1 --

-- A mano:

INSERT INTO "Códigos postales I"("Código postal", Población, Provincia)
VALUES ('08050', 'Parets', 'Barcelona');

INSERT INTO "Códigos postales I"("Código postal", Población, Provincia)
VALUES ('14200', 'Peñarroya', 'Córdoba');

INSERT INTO "Códigos postales I"("Código postal", Población, Provincia)

```

VALUES ('14900','Lucena','Córdoba');
INSERT INTO "Códigos postales I"("Código postal", Población, Provincia)
VALUES ('','Arganda','Sevilla');
INSERT INTO "Códigos postales I"("Código postal", Población, Provincia)
VALUES ('08050','Zaragoza','Zaragoza');
INSERT INTO "Códigos postales I"("Código postal", Población, Provincia)
VALUES ('28040','Arganda','Madrid');
INSERT INTO "Códigos postales I"("Código postal", Población, Provincia)
VALUES ('28004','Madrid','Madrid');

```

```

INSERT INTO "Domicilios I"(DNI, Calle, "Código postal")
VALUES ('12345678A','Avda. Complutense','28040');
INSERT INTO "Domicilios I"(DNI, Calle, "Código postal")
VALUES ('12345678A','Cántaro','28004');
INSERT INTO "Domicilios I"(DNI, Calle, "Código postal")
VALUES ('12345678P','Diamante','14200');
INSERT INTO "Domicilios I"(DNI, Calle, "Código postal")
VALUES ('12345678P','Carbón','14901');

```

-- Con Oracle Loader:

```

EXP USERID=USU_GII_DIAZGONZALEZ/BBDD1516@BDC owner=(USU_GII_DIAZGONZALEZ) FILE=bbddexp
IMP USERID=USU_GII_DIAZGONZALEZ/BBDD1516@BDC TOUSER=(USU_GII_DIAZGONZALEZ) FILE=bbddexp

```

----- -- 4. Escribir un procedimiento "ComprobarNulos" -- -----

```

CREATE OR REPLACE PROCEDURE ComprobarNulos AS
    excepcion_nulo EXCEPTION;
    varCP "Códigos postales I"."Código postal"%TYPE;
    varPoblacion "Códigos postales I".Población%TYPE;
    varProvincia "Códigos postales I".Provincia%TYPE;
    CURSOR cursorComprobarNulos IS
        SELECT "Código postal", Población, Provincia
        FROM "Códigos postales I"
        WHERE "Código postal" IS NULL OR
              Población IS NULL OR
              Provincia IS NULL;
BEGIN
    OPEN cursorComprobarNulos;
    LOOP
        FETCH cursorComprobarNulos INTO varCP,varPoblacion,varProvincia;
        IF varCP IS NULL OR varPoblacion IS NULL OR varProvincia IS NULL THEN

```

```

                RAISE excepcion_nulo;
            END IF;
            EXIT WHEN cursorComprobarNulos%NOTFOUND;
        END LOOP;
    CLOSE cursorComprobarNulos;
EXCEPTION
    WHEN excepcion_nulo THEN
        DBMS_OUTPUT.PUT_LINE('Algún campo nulo en: '||varCP||','||varPoblacion||','||varProvincia);
END;

EXECUTE ComprobarNulos;
-- Respuesta:
-- Procedure COMPROBARNULOS compilado
-- Procedimiento PL/SQL terminado correctamente.
-- Algún campo nulo en: ,Arganda
,sevilla

```

-- 5. Escribir un procedimiento "ComprobarPK" --

```

create or replace PROCEDURE ComprobarPK AS
    excepcion_PKNula EXCEPTION;
    excepcion_PKRepetida EXCEPTION;
    varCP "Códigos postales I"."Código postal"%TYPE;
    varCPcomp "Códigos postales I"."Código postal"%TYPE;
    CURSOR cursorPKRepetidaA IS
        SELECT "Código postal"
        FROM "Códigos postales I";
    CURSOR cursorPKRepetidaB IS
        SELECT "Código postal"
        FROM "Códigos postales I";
BEGIN
    OPEN cursorPKRepetidaA;
    LOOP
        FETCH cursorPKRepetidaA INTO varCP;
        IF varCP IS NULL THEN
            RAISE excepcion_PKNula;
        END IF;
        OPEN cursorPKRepetidaB;
        LOOP
            FETCH cursorPKRepetidaB INTO varCPcomp;
            IF varCP IS NOT NULL AND varCPcomp IS NOT NULL AND
                varCP=varCPcomp AND

```

```

                cursorPKRepetidaA%ROWCOUNT<>cursorPKRepetidaB%ROWCOUNT THEN
                    RAISE excepcion_PKrepetida;
                END IF;
                EXIT WHEN cursorPKRepetidaB%NOTFOUND;
            END LOOP;
            EXIT WHEN cursorPKRepetidaA%NOTFOUND;
            CLOSE cursorPKRepetidaB;
        END LOOP;
        CLOSE cursorPKRepetidaA;
    EXCEPTION
        WHEN excepcion_PKNu1a THEN
            DBMS_OUTPUT.PUT_LINE('Hay PK nu1a(s)');
        WHEN excepcion_PKRepetida THEN
            DBMS_OUTPUT.PUT_LINE('La PK '||varCP||' está repetida');
    END;

EXECUTE ComprobarPK;
-- Respuesta:
-- Procedure COMPROBARPK compilado
-- Procedimiento PL/SQL terminado correctamente.
-- La PK 08050 está repetida

```

-- 6. Escribir un procedimiento "ComprobarFD" --

```

CREATE OR REPLACE PROCEDURE ComprobarFD AS
    excepcion_FDRepetida EXCEPTION;
    varCP "Códigos postales I"."Código postal"%TYPE;
    varPob "Códigos postales I".Población%TYPE;
    varProv "Códigos postales I".Provincia%TYPE;
    varCPcomp "Códigos postales I"."Código postal"%TYPE;
    varPobcomp "Códigos postales I".Población%TYPE;
    varProvcomp "Códigos postales I".Provincia%TYPE;
    CURSOR cursorFDRepetidaA IS
        SELECT "Código postal", Población, Provincia
        FROM "Códigos postales I";
    CURSOR cursorFDRepetidaB IS
        SELECT "Código postal", Población, Provincia
        FROM "Códigos postales I";
BEGIN
    OPEN cursorFDRepetidaA;
    LOOP

```

```

    FETCH cursorFDRepetidaA INTO varCP, varPob, varProv;
    OPEN cursorFDRepetidaB;
    LOOP
        FETCH cursorFDRepetidaB INTO varCPcomp, varPobcomp, varProvcomp;
        IF (varCP IS NOT NULL) AND
            (varCPcomp IS NOT NULL) AND
            (varPob IS NOT NULL) AND
            (varPobcomp IS NOT NULL) AND
            (varProv IS NOT NULL) AND
            (varProvcomp IS NOT NULL) AND
            (varCP=varCPcomp) AND
            (varPob<>varPobcomp OR varProv<>varProvcomp) AND
            (cursorFDRepetidaA%ROWCOUNT<>cursorFDRepetidaB%ROWCOUNT) THEN
            RAISE excepcion_FDRepetida;
        END IF;
        EXIT WHEN cursorFDRepetidaB%NOTFOUND;
    END LOOP;
    EXIT WHEN cursorFDRepetidaA%NOTFOUND;
    CLOSE cursorFDRepetidaB;
END LOOP;
CLOSE cursorFDRepetidaA;
EXCEPTION
    WHEN excepcion_FDRepetida THEN
        DBMS_OUTPUT.PUT_LINE('Existe violación de dependencia funcional con el CP '||varCP);
END;

EXECUTE ComprobarFD;
-- Respuesta:
-- Procedure COMPROBARFD compilado
-- Procedimiento PL/SQL terminado correctamente.
-- Existe violación de dependencia funcional con el CP 08050

-----
-- 7. Escribir un procedimiento "ComprobarFK" --
-----
CREATE OR REPLACE PROCEDURE ComprobarFK AS
    excepcion_FK7 EXCEPTION;
    varCPD "Domicilios I"."Código postal"%TYPE;
    varCP "Códigos postales I"."Código postal"%TYPE;
    aux INTEGER:=0;
    CURSOR cursorFK7 IS
        SELECT "Código postal"

```

```

        FROM "Domicilios I";
CURSOR cursorFK7b IS
    SELECT "Código postal"
    FROM "Códigos postales I";
BEGIN
    OPEN cursorFK7;
    LOOP
        FETCH cursorFK7 INTO varCPD;
        OPEN cursorFK7b;
        LOOP
            FETCH cursorFK7b INTO varCP;
            IF (varCPD IS NOT NULL) AND (varCP IS NOT NULL) AND (varCPD=varCP) THEN
                aux:=aux+1;
            END IF;
            EXIT WHEN cursorFK7b%NOTFOUND;
        END LOOP;
        IF aux=0 THEN
            RAISE excepcion_FK7;
        ELSE
            aux:=0;
        END IF;
        EXIT WHEN cursorFK7%NOTFOUND;
        CLOSE cursorFK7b;
    END LOOP;
    CLOSE cursorFK7;
EXCEPTION
    WHEN excepcion_FK7 THEN
        DBMS_OUTPUT.PUT_LINE('Existe violación de dependencia referencial con el CP: '||varCPD);
END;

EXECUTE ComprobarFK;
-- Respuesta:
-- Procedure COMPROBARFK compilado
-- Procedimiento PL/SQL terminado correctamente.
-- Existe violación de dependencia referencial con el CP: 14901

```