

Segunda-relacion-PLSQL.pdf



Juandf03



Administración de Bases de Datos



3º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**



Estamos de
Aniversario

De la universidad al
mercado laboral:
especialízate con los posgrados
de EOI y marca la diferencia.



EOI Escuela de
organización
industrial



saber más

Menú King Ahorro

Molar sin gastar mucho,
mola mucho más.



Menú Cheeseburger



Menú Crispy Chicken



Menú Cheddar Lover Double Cheeseburger

PIDE YA

```
--1
CREATE TABLE TB_OBJETOS (
  NOMBRE VARCHAR2(300),
  CODIGO NUMBER,
  FECHA_CREACION DATE,
  FECHA_MODIFICACION DATE,
  TIPO VARCHAR2(30),
  ESQUEMA_ORIGINAL VARCHAR2(30)
);

DECLARE
  CURSOR c_all_objects IS
    SELECT object_name, object_id, created, last_ddl_time, object_type, owner
    FROM all_objects;

BEGIN
  FOR obj IN c_all_objects LOOP
    INSERT INTO TB_OBJETOS (
      NOMBRE, CODIGO, FECHA_CREACION, FECHA_MODIFICACION, TIPO,
      ESQUEMA_ORIGINAL
    ) VALUES (
      obj.object_name, obj.object_id, obj.created, obj.last_ddl_time, obj.object_type,
      obj.owner
    );
  END LOOP;

  COMMIT;
END;
/

--2
CREATE TABLE TB_ESTILO (
  TIPO_OBJETO VARCHAR2(30),
  PREFIJO VARCHAR2(30)
);

--3
ALTER TABLE TB_OBJETOS
ADD (
  ESTADO VARCHAR2(10),
  NOMBRE_CORRECTO VARCHAR2(100)
);
```

WUOLAH

```

CREATE OR REPLACE PROCEDURE PR_COMPROBAR (ESQUEMA IN VARCHAR2
DEFAULT NULL) AS
    CURSOR c_objetos IS
        SELECT NOMBRE, TIPO, ESQUEMA_ORIGINAL FROM TB_OBJETOS
        WHERE ESQUEMA_ORIGINAL = NVL(ESQUEMA, ESQUEMA_ORIGINAL) FOR
UPDATE;

    v_prefijo TB_ESTILO.PREFIJO%TYPE;
    v_nombre_correcto TB_OBJETOS.NOMBRE%TYPE;
BEGIN
    FOR obj IN c_objetos LOOP
        BEGIN
            -- Obtener el prefijo correspondiente al tipo de objeto
            SELECT PREFIJO INTO v_prefijo FROM TB_ESTILO WHERE TIPO_OBJETO =
obj.TIPO;

            -- Construir el nombre correcto
            v_nombre_correcto := v_prefijo || obj.NOMBRE;

            -- Verificar si el nombre correcto excede el tamaño permitido
            IF LENGTH(v_nombre_correcto) > 100 THEN
                v_nombre_correcto := SUBSTR(v_nombre_correcto, 1, 100);
            END IF;

            -- Actualizar la tabla TB_OBJETOS
            IF obj.NOMBRE = v_nombre_correcto THEN
                UPDATE TB_OBJETOS SET ESTADO = 'CORRECTO', NOMBRE_CORRECTO =
NULL WHERE CURRENT OF c_objetos;
            ELSE
                UPDATE TB_OBJETOS SET ESTADO = 'INCORRECTO',
NOMBRE_CORRECTO = v_nombre_correcto WHERE CURRENT OF c_objetos;
            END IF;

            EXCEPTION
                WHEN NO_DATA_FOUND THEN
                    -- No hay prefijo definido para este tipo de objeto
                    UPDATE TB_OBJETOS SET ESTADO = 'SIN NORMA', NOMBRE_CORRECTO =
NULL WHERE CURRENT OF c_objetos;
            END;
        END LOOP;

        COMMIT;
    END PR_COMPROBAR;
/

```

```

BEGIN

```

```
PR_COMPROBAR('SYS');  
END;  
/
```