

## DEPARTAMENTO DE INFORMÁTICA

Ciclo Formativo: DAW

Módulo: BD (Bases de Datos)

Curso escolar: 2023-2024 Ejercicio: Práctica 2 UT6 PL/SQL

Evaluación: 3

Fecha aprobación: 16/04/2024

Fecha realización ejercicio: 19/04/2024

Práctica elaborada y aprobado por: A. Cristina Escudero

PR0602	<b>UNIDAD DE TRABAJO 6. PRÁCTICA 2 PL/SQL. ORACLE</b>	DAW1
--------	---	------

Con la base de datos *jardineria\_orc\_rev.sql* realiza las siguientes tareas. Copia los comandos (código en texto) y los resultados obtenidos en este documento (capturas de pantalla). Prueba los programas.

**Previo:** crea un usuario llamado `c##pr0602<tuNombre>` con clave `pr0602` y dale los permisos necesarios. Carga la base de datos y comprueba que esté todo correcto.

### EJERCICIO 1:

1. Haz una función que cuente los pedidos del año pasado por parámetro que hay. Si no hay ninguno debe devolver un 0, si hay algún error devolverá un -1 y en otro caso debe devolver el número de pedidos. Debes usar gestión de excepciones.  
(2 puntos)

```
create or replace function contarPedidosAnno (anno in
varchar2)
return number as
total number;
begin
    select count(*) into total from pedidos where
extract(year from fechapedido) = anno;

    if total is null then
        return 0;
    else
        return total;
    end if;

exception
    when other then
        return -1;
end;
```

```
set serveroutput on;

-- Ejercicio 01
create or replace function contarPedidosAnno (anno in varchar2)
return number as
total number;
begin
    select count(*) into total from pedidos where extract(year from fechapedido) = anno;

    if total is null then
        return 0;
    else
        return total;
    end if;

exception
    when others then
        return -1;
end;
/

declare
    anno varchar2(1000) := '2007';
begin
    dbms_output.put_line('Pedidos realizados en el año ' || anno || ': ' || contarPedidosAnno(anno));
end;
/
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0,049 segundos

Pedidos realizados en el año hola: -1

Procedimiento PL/SQL terminado correctamente.  
Pedidos realizados en el año 2000: 0

Procedimiento PL/SQL terminado correctamente.  
Pedidos realizados en el año 2007: 6

Procedimiento PL/SQL terminado correctamente.

2. Haz otra función en el que le pasemos como parámetro la gama de producto y nos devuelva el número de productos de esa gama que hay. Si no hay ninguno debe devolver un 0, si hay algún error devolverá un -1 y en otro caso debe devolver el número de productos de esa gama. Debes usar gestión de excepciones.  
(2 puntos)

```
create or replace function contarProductosGama (inputGama
in productos.gama%type)

return number as

total number;

begin

    select count(*) into total from pedidos where gama =
inputGama;

    if total is null then

        return 0;

    else

        return total;

    end if;

exception

    when other then

        return -1;

end;
```

```
set serveroutput on;

create or replace function contarProductosGama (inputGama in productos.gama%type)
return number as
total number;
begin
    select count(*) into total from productos where gama = inputGama;

    if total is null then
        return 0;
    else
        return total;
    end if;

exception
    when others then
        return -1;
end;
/

declare
gama varchar2(1000) := '91832kqhwbkhhbc122oi34e';
begin
    dbms_output.put_line('Productos totales de ' || gama || ': ' || contarProductosGama(gama));
end;
/
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0,042 segundos

Productos totales de Peras: 0

Procedimiento PL/SQL terminado correctamente.

Productos totales de Frutales: 108

Procedimiento PL/SQL terminado correctamente.

Productos totales de 91832kqhwbkhhbc122oi34e: 0

Procedimiento PL/SQL terminado correctamente.

3. Diseña un procedimiento que muestre por pantalla, para cada producto, el margen de beneficio del mismo. [Usamos cursor explícito y gestionamos excepciones: si no hay productos, no hay precio de proveedor, otros errores.]  
(3 puntos)

```
CREATE OR REPLACE PROCEDURE mostrar_margen_beneficio AS
  CURSOR c_productos IS
    SELECT CodigoProducto, Nombre, PrecioVenta, PrecioProveedor
    FROM Productos;

  v_margen_beneficio NUMBER(15,2);
BEGIN
  FOR producto IN c_productos LOOP
    BEGIN
      IF producto.PrecioProveedor IS NULL THEN
        DBMS_OUTPUT.PUT_LINE('El producto ' || producto.Nombre || ' no
tiene precio de proveedor.');
```

```

CREATE OR REPLACE PROCEDURE mostrar_margen_beneficio AS
CURSOR c_productos IS
SELECT CodigoProducto, Nombre, PrecioVenta, PrecioProveedor
FROM Productos;

v_margen_beneficio NUMBER(15,2);
BEGIN
FOR producto IN c_productos LOOP
BEGIN
IF producto.PrecioProveedor IS NULL THEN
DBMS_OUTPUT.PUT_LINE('El producto ' || producto.Nombre || ' no tiene precio de proveedor.');
```

```

ELSE
v_margen_beneficio := producto.PrecioVenta - producto.PrecioProveedor;
DBMS_OUTPUT.PUT_LINE('El margen de beneficio del producto ' || producto.Nombre || ' es ' || v_margen_beneficio);
END IF;
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Error al calcular el margen de beneficio del producto ' || producto.Nombre);
END;
END LOOP;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('No hay productos.');
```

```

END mostrar_margen_beneficio;
/
exec mostrar_margen_beneficio;
```

Salida de Script x

Tarea terminada en 0,072 segundos

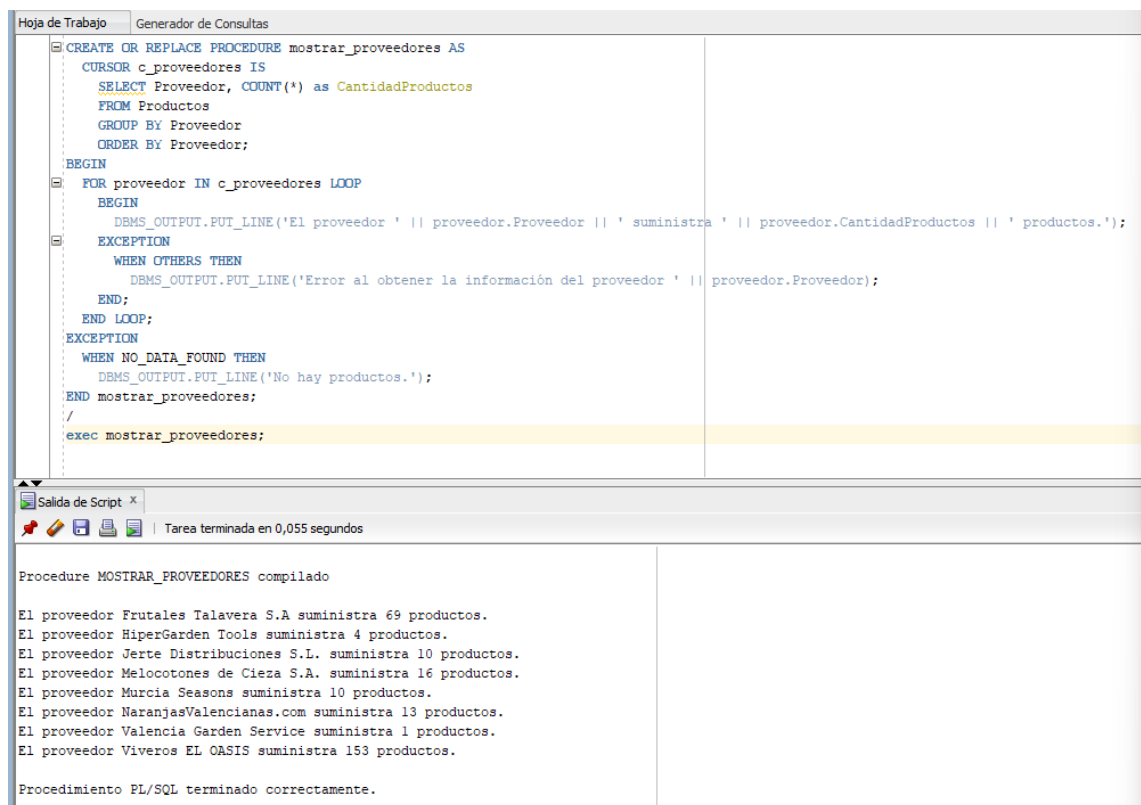
```

El margen de beneficio del producto Peral es 7
El margen de beneficio del producto Limonero 30/40 es 20
El margen de beneficio del producto Kunquat es 5
El margen de beneficio del producto Kunquat EXTRA con FRUTA es 12
El margen de beneficio del producto Calamondin Mini es 2
El margen de beneficio del producto Calamondin Copa es 5
El margen de beneficio del producto Calamondin Copa EXTRA Con FRUTA es 9
El margen de beneficio del producto Rosal bajo 1A* -En maceta-inicio brotación es 1
El margen de beneficio del producto ROSAL TREPADOR es 1
El margen de beneficio del producto Camelia Blanco, Chrysler Rojo, Soraya Naranja, es 1
El margen de beneficio del producto Naranjo -Plantón joven 1 año injerto es 2
El margen de beneficio del producto Landora Amarillo, Rose Gaujard bicolor blanco-rojo es 1
El margen de beneficio del producto Kordes Perfect bicolor rojo-amarillo, Roundelay rojo fuerte es 1
El margen de beneficio del producto Pitimini rojo es 1
El margen de beneficio del producto Rosal copa es 2
El margen de beneficio del producto Sierra de Poda 400MM es 3
El margen de beneficio del producto Pala es 1
El margen de beneficio del producto Rastrillo de Jardín es 1
El margen de beneficio del producto Azadón es 1
El margen de beneficio del producto Ajedrea es 1
El margen de beneficio del producto Lavándula Dentata es 1
El margen de beneficio del producto Mejorana es 1
El margen de beneficio del producto Melissa es 1
El margen de beneficio del producto Mentha Sativa es 1
El margen de beneficio del producto Petrosilium Hortense (Peregil) es 1
El margen de beneficio del producto Salvia Mix es 1
El margen de beneficio del producto Thymus Citriodra (Tomillo limón) es 1
El margen de beneficio del producto Thymus Vulgaris es 1
```

4. Diseña un procedimiento que muestre por pantalla los proveedores de productos, ordenados alfabéticamente y sin repeticiones, y cuántos productos nos suministran. [Usamos cursores y gestionamos excepciones: no hay productos, otros errores.]  
(3 puntos)

```
CREATE OR REPLACE PROCEDURE mostrar_proveedores AS
    CURSOR c_proveedores IS
        SELECT Proveedor, COUNT(*) as CantidadProductos
        FROM Productos
        GROUP BY Proveedor
        ORDER BY Proveedor;
BEGIN
    FOR proveedor IN c_proveedores LOOP
        BEGIN
            DBMS_OUTPUT.PUT_LINE('El proveedor ' || proveedor.Proveedor || '
suministra ' || proveedor.CantidadProductos || ' productos.');
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Error al obtener la información del
proveedor ' || proveedor.Proveedor);
        END;
    END LOOP;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No hay productos.');
```

END mostrar\_proveedores;  
/



```
CREATE OR REPLACE PROCEDURE mostrar_proveedores AS
    CURSOR c_proveedores IS
        SELECT Proveedor, COUNT(*) as CantidadProductos
        FROM Productos
        GROUP BY Proveedor
        ORDER BY Proveedor;
BEGIN
    FOR proveedor IN c_proveedores LOOP
        BEGIN
            DBMS_OUTPUT.PUT_LINE('El proveedor ' || proveedor.Proveedor || ' suministra ' || proveedor.CantidadProductos || ' productos.');
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Error al obtener la información del proveedor ' || proveedor.Proveedor);
        END;
    END LOOP;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No hay productos.');
```

END mostrar\_proveedores;  
/  
exec mostrar\_proveedores;

Salida de Script x | Tarea terminada en 0,055 segundos

Procedure MOSTRAR\_PROVEEDORES compilado

El proveedor Frutales Talavera S.A suministra 69 productos.  
El proveedor HiperGarden Tools suministra 4 productos.  
El proveedor Jerte Distribuciones S.L. suministra 10 productos.  
El proveedor Melocotones de Cieza S.A. suministra 16 productos.  
El proveedor Murcia Seasons suministra 10 productos.  
El proveedor NaranjasValencianas.com suministra 13 productos.  
El proveedor Valencia Garden Service suministra 1 productos.  
El proveedor Viveros EL OASIS suministra 153 productos.

Procedimiento PL/SQL terminado correctamente.

