

DEPARTAMENTO DE INFORMÁTICA

Ciclo Formativo: DAW

Módulo: BD (Bases de Datos)

Curso escolar: 2023-2024 Ejercicio: Práctica 1 UT6 PL/SQL

Evaluación: 3

Fecha aprobación: 15/04/2024

Fecha realización ejercicio: 17/04/2024

Práctica elaborada y aprobado por: A. Cristina Escudero

PR0601	UNIDAD DE TRABAJO 6. PRÁCTICA 1 PL/SQL. ORACLE	DAW1
--------	-----------------------------------------------------------	------

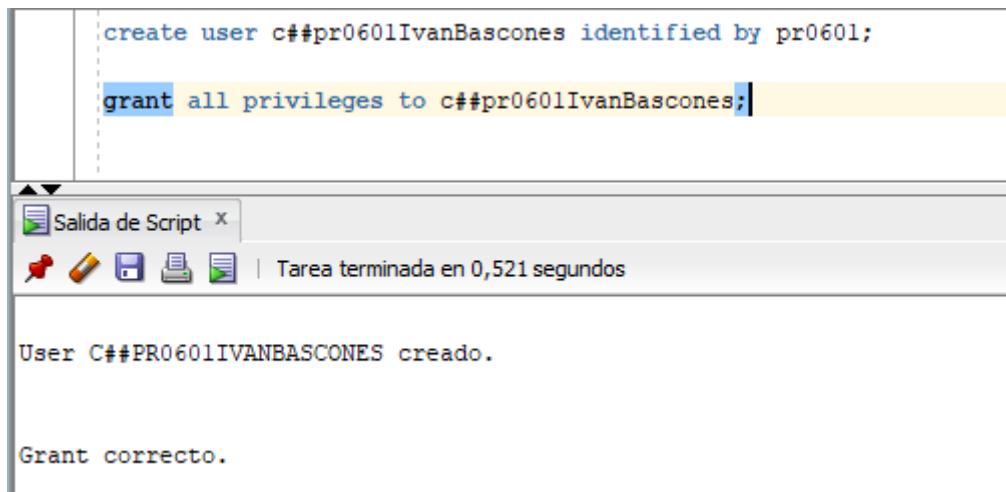
Con la base de datos *jardineria_oracle.sql* realiza las siguientes tareas.
Copia los comandos (código en texto) y los resultados obtenidos en este documento (capturas de pantalla).

EJERCICIO 1:

1. Crea un usuario llamado c##pr0601<tuNombre> con clave pr0601 y dale los permisos necesarios. [Nota: debes crearlo desde el usuario system].

```
create user c##pr0601IvanBascones identified by pr0601;
```

```
grant all privileges to c##pr0601IvanBascones;
```



```
create user c##pr0601IvanBascones identified by pr0601;

grant all privileges to c##pr0601IvanBascones;
```

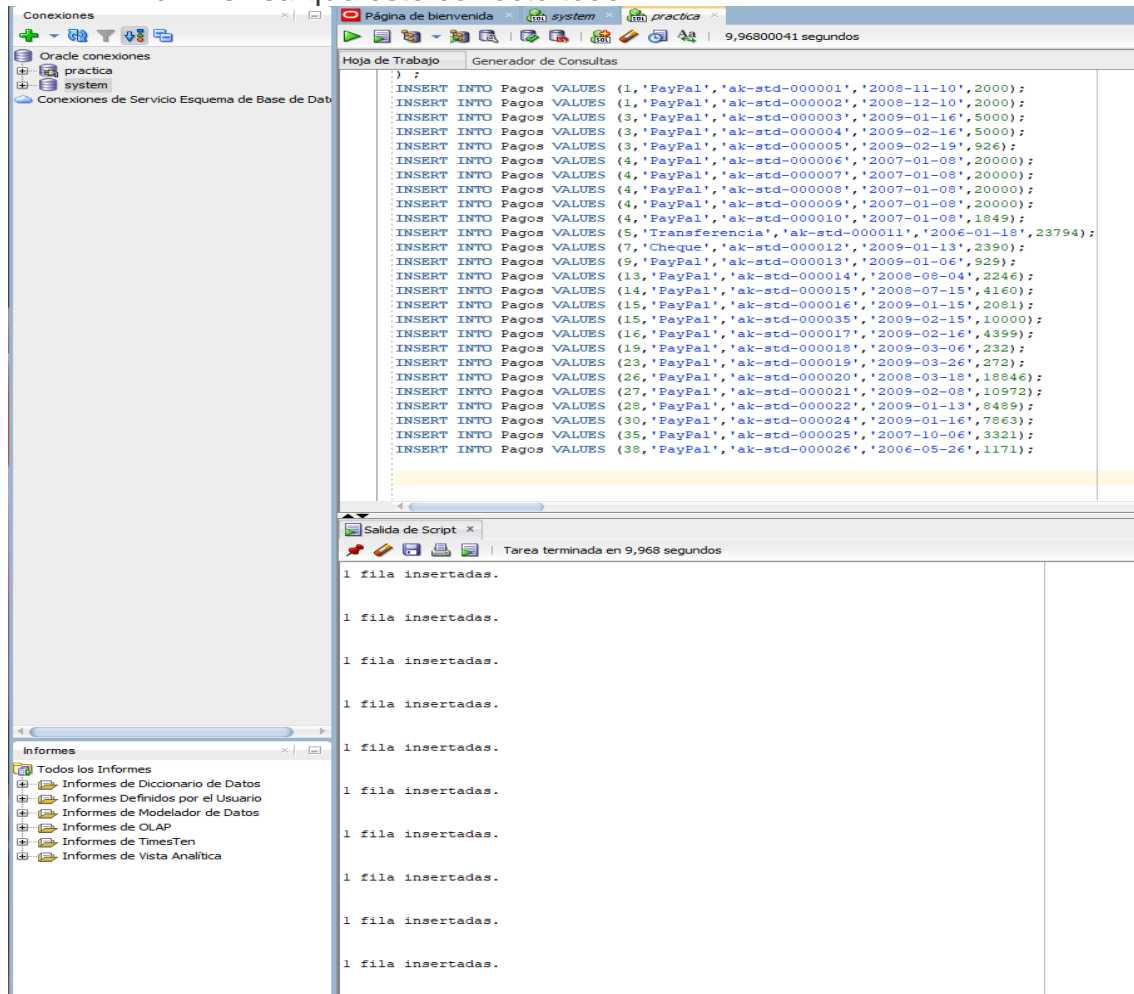
Salida de Script x

Tarea terminada en 0,521 segundos

```
User C##PR0601IVANBASCONES creado.

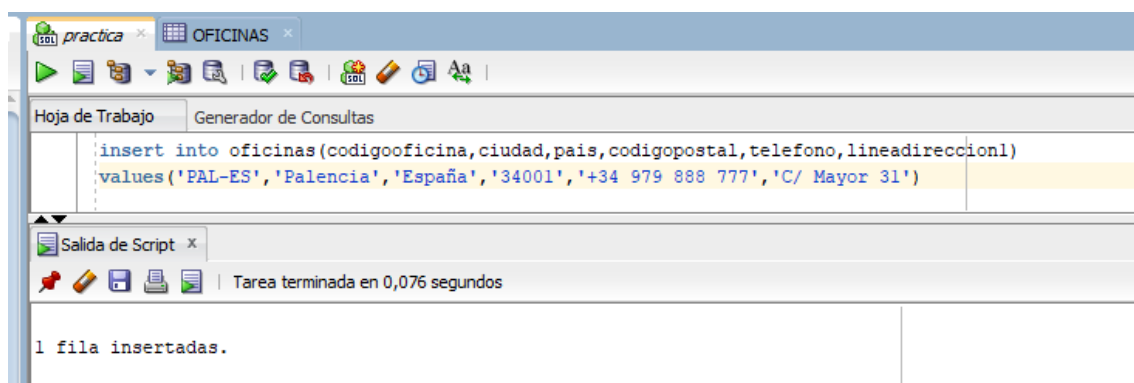
Grant correcto.
```

2. Entra en el usuario creado y realiza las siguientes operaciones:
 - a. Carga la base de datos.
 - b. Revisa que esté correcto todo.



3. Inserta una nueva oficina llamada PAL-ES, situada en Palencia, España, código postal 34001, teléfono: +34979888777, C/ Mayor 31

insert into oficinas (codigooficina, ciudad, pais, codigopostal, telefono, lineadireccion1) values('PAL-ES','Palencia','España','34001','+34 979 888 777','C/ Mayor 31')



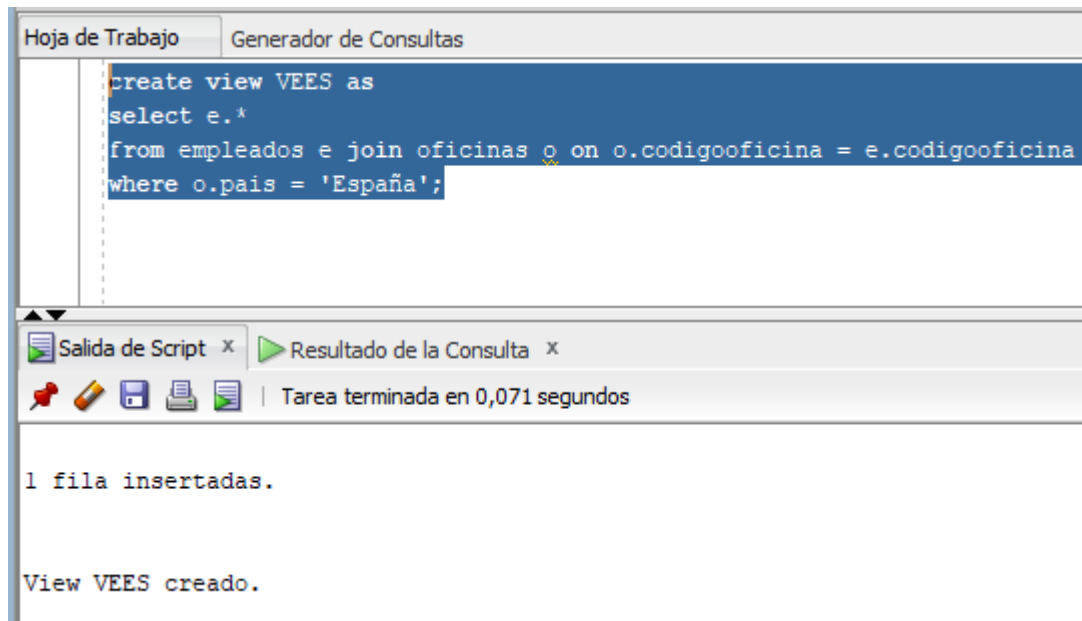
4. Crea una vista llamada VEES con los empleados de España.

create view VEES as

select e.*

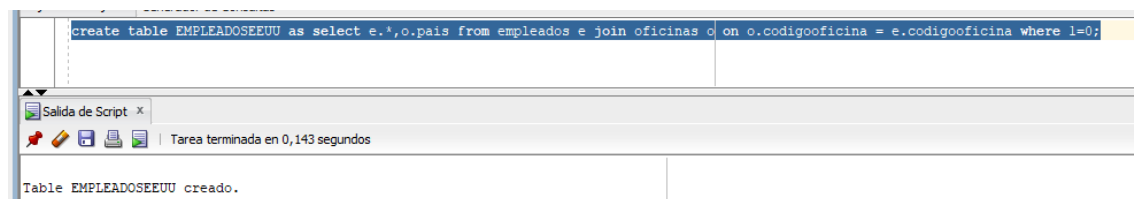
from empleados e join oficinas o on o.codigooficina = e.codigooficina

where o.pais = 'España';



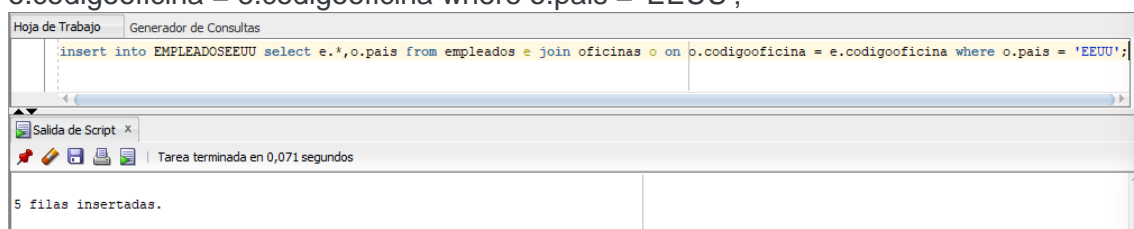
5. Crea una tabla EMPLEADOSEEUU con los mismos atributos de empleados más el país de la oficina.

create table EMPLEADOSEEUU as select e.*,o.pais from empleados e join oficinas o on o.codigooficina = e.codigooficina where 1=0;



6. Insertar los datos de los empleados de Estados Unidos en esa tabla.

insert into EMPLEADOSEEUU select e.*,o.pais from empleados e join oficinas o on o.codigooficina = e.codigooficina where o.pais = 'EEUU';



7. Haz un programa que cuente los empleados que hay y muestre en pantalla el número.

declare

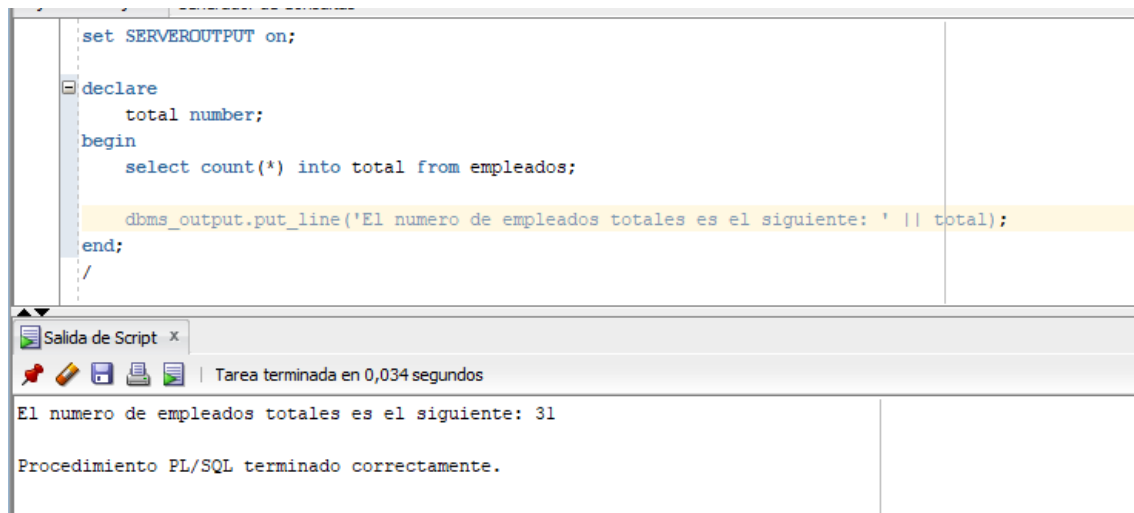
total number;

begin

select count(*) into total from empleados;

dbms_output.put_line('El numero de empleados totales es el siguiente' || total);

end;



The screenshot shows a SQL Developer window with a script editor and a results pane. The script editor contains the following PL/SQL code:

```
set SERVEROUTPUT on;

declare
    total number;
begin
    select count(*) into total from empleados;

    dbms_output.put_line('El numero de empleados totales es el siguiente: ' || total);
end;
/
```

The results pane shows the output of the script:

Tarea terminada en 0,034 segundos

El numero de empleados totales es el siguiente: 31

Procedimiento PL/SQL terminado correctamente.

8. Ahora haz una función que realice lo mismo que el apartado anterior, pero en lugar de mostrarlo en pantalla, devuelva el entero correspondiente. Función numEmpleados.

```
create or replace function numEmpleados return number as
```

```
total number;
```

```
begin
```

```
    select count(*) into total from empleados;
```

```
    return total;
```

```
end;
```

```
/
```

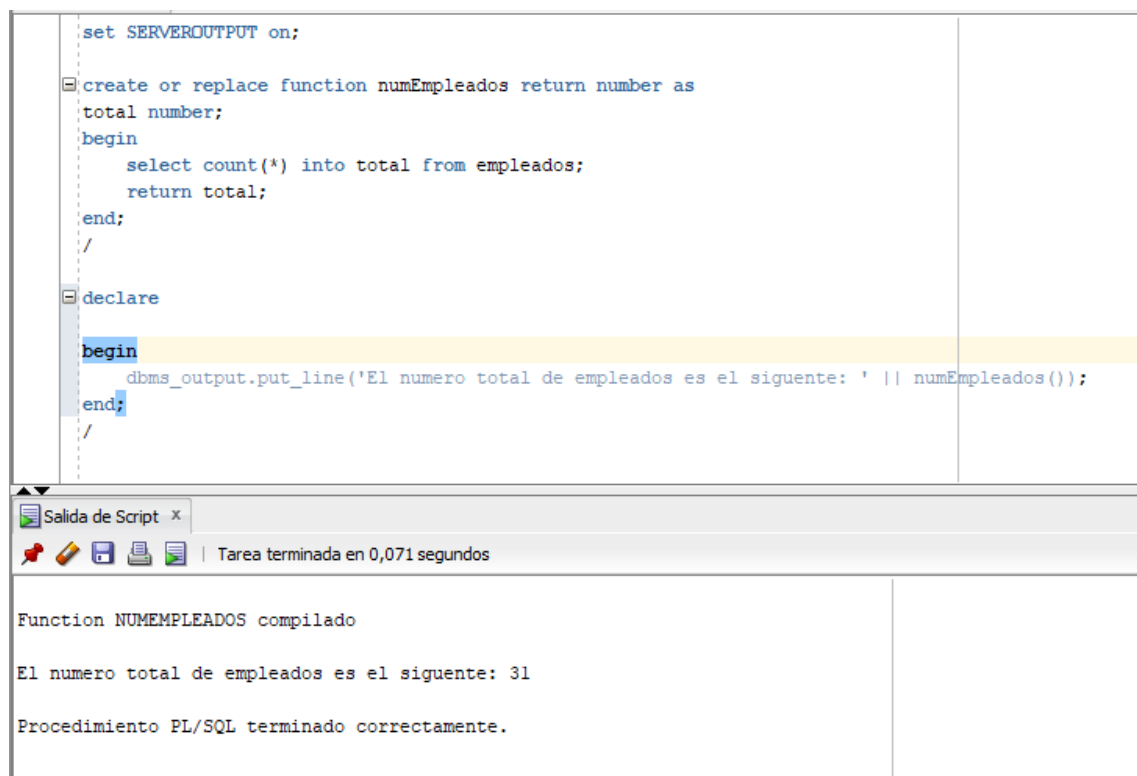
```
declare
```

```
begin
```

```
    dbms_output.put_line('El numero total de empleados es el siguiente: ' ||  
numEmpleados());
```

```
end;
```

```
/
```



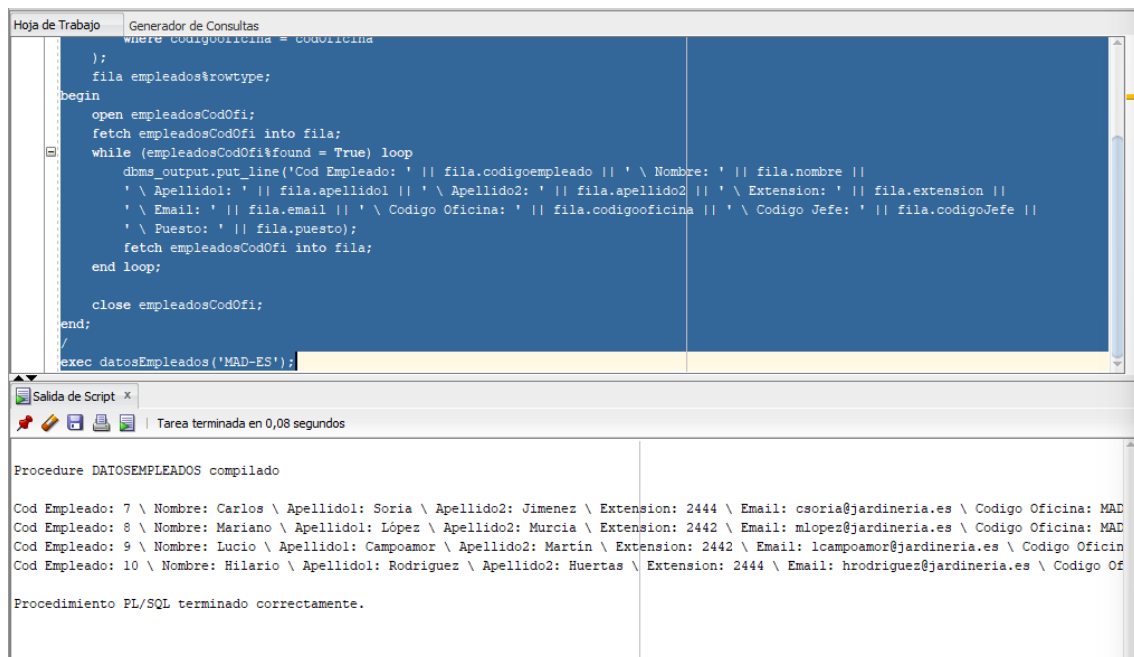
```
set SERVEROUTPUT on;  
  
create or replace function numEmpleados return number as  
total number;  
begin  
    select count(*) into total from empleados;  
    return total;  
end;  
/  
  
declare  
  
begin  
    dbms_output.put_line('El numero total de empleados es el siguiente: ' || numEmpleados());  
end;  
/  
  
-- Output window --  
Salida de Script x  
Tarea terminada en 0,071 segundos  
  
Function NUMEMPLEADOS compilado  
  
El numero total de empleados es el siguiente: 31  
  
Procedimiento PL/SQL terminado correctamente.
```

9. Diseña un procedimiento que muestre por pantalla los datos de los empleados dado el código de oficina donde trabajan. [Usamos cursores]

```
set SERVEROUTPUT on;
```

```
create or replace procedure datosEmpleados(codOficina
oficinas.codigooficina%type) as
  cursor empleadosCodOfi is (
    select *
    from empleados
    where codigooficina = codOficina
  );
  fila empleados%rowtype;
begin
  open empleadosCodOfi;
  fetch empleadosCodOfi into fila;
  while (empleadosCodOfi%found = True) loop
    dbms_output.put_line('Cod Empleado: ' || fila.codigoempleado || ' \
Nombre: ' || fila.nombre ||
      ' \ Apellido1: ' || fila.apellido1 || ' \ Apellido2: ' || fila.apellido2 || ' \ Extension:
' || fila.extension ||
      ' \ Email: ' || fila.email || ' \Codigo Oficina: ' || fila.codigooficina || ' \Codigo
Jefe: ' || fila.codigoJefe ||
      ' \ Puesto: ' || fila.puesto);
    fetch empleadosCodOfi into fila;
  end loop;

  close empleadosCodOfi;
end;
/
exec datosEmpleados('MAD-ES');
```



```
);
  fila empleados%rowtype;
begin
  open empleadosCodOfi;
  fetch empleadosCodOfi into fila;
  while (empleadosCodOfi%found = True) loop
    dbms_output.put_line('Cod Empleado: ' || fila.codigoempleado || ' \ Nombre: ' || fila.nombre ||
      ' \ Apellido1: ' || fila.apellido1 || ' \ Apellido2: ' || fila.apellido2 || ' \ Extension: ' || fila.extension ||
      ' \ Email: ' || fila.email || ' \Codigo Oficina: ' || fila.codigooficina || ' \Codigo Jefe: ' || fila.codigoJefe ||
      ' \ Puesto: ' || fila.puesto);
    fetch empleadosCodOfi into fila;
  end loop;

  close empleadosCodOfi;
end;
/
exec datosEmpleados('MAD-ES');
```

Salida de Script x

Tarea terminada en 0,08 segundos

Procedure DATOSEMPLEADOS compilado

Cod Empleado: 7 \ Nombre: Carlos \ Apellido1: Soria \ Apellido2: Jimenez \ Extension: 2444 \ Email: csoria@jardineria.es \ Codigo Oficina: MAD
Cod Empleado: 8 \ Nombre: Mariano \ Apellido1: López \ Apellido2: Murcia \ Extension: 2442 \ Email: mlopez@jardineria.es \ Codigo Oficina: MAD
Cod Empleado: 9 \ Nombre: Lucio \ Apellido1: Campoamor \ Apellido2: Martín \ Extension: 2442 \ Email: lcampoamor@jardineria.es \ Codigo Oficina: MAD
Cod Empleado: 10 \ Nombre: Hilario \ Apellido1: Rodriguez \ Apellido2: Huertas \ Extension: 2444 \ Email: hrodriguez@jardineria.es \ Codigo Oficina: MAD

Procedimiento PL/SQL terminado correctamente.

10. Diseña un procedimiento que muestre por pantalla los datos de los pedidos (y detalles) que contengan (en su detalle) un producto concreto, dado su nombre de producto. [Usamos cursores. Puede haber varios productos con igual nombre.]

```
create or replace procedure pedidosProducto (nomProducto productos.nombre%type) as
cursor pedidosProducto is(
    SELECT p.codigopedido, p.fechapedido, p.fechaesperada, p.fechaentrega, p.estado, p.comentarios, p.codigocliente, dp.codigoproducto, d
    FROM detallepedidos dp
    JOIN pedidos p ON p.codigopedido = dp.codigopedido
    JOIN productos pr ON pr.codigoproducto = dp.codigoproducto
    WHERE pr.nombre = nomProducto
);
begin
    for pedido in pedidosProducto loop
        dbms_output.put_line(pedido.codigopedido || ' / ' || pedido.fechapedido || ' / ' || pedido.fechaesperada
        || ' / ' || pedido.fechaentrega || ' / ' || pedido.estado || ' / ' || pedido.comentarios || ' / ' ||
        pedido.codigocliente || ' / ' || pedido.codigoproducto || ' / ' || pedido.cantidad
        || ' / ' || pedido.cantidad || ' / ' || pedido.preciounidad || ' / ' || pedido.numerolinea);
    end loop;
end ;
/
exec pedidosProducto('Peral');
```