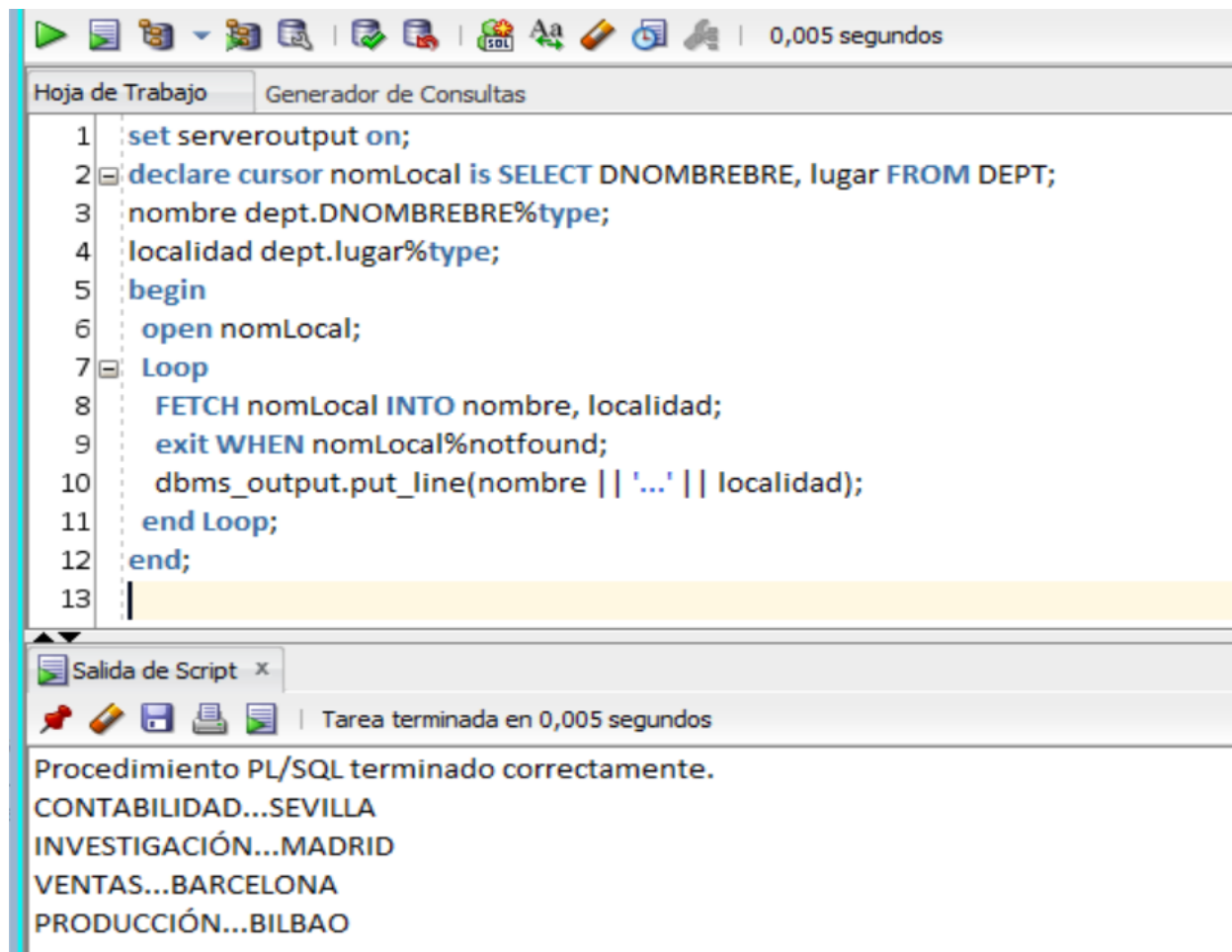


1. Programa anónimo que visualice el nombre y la localidad de todos los departamentos.

```
set serveroutput on;
declare cursor nomLocal is SELECT DNOMBREBRE, lugar FROM DEPT;
nombre dept.DNOMBREBRE%type;
localidad dept.lugar%type;
begin
    open nomLocal;
    Loop
        FETCH nomLocal INTO nombre, localidad;
        exit WHEN nomLocal%notfound;
        dbms_output.put_line(nombre || '...' || localidad);
    end Loop;
end;
```



2. Programa que muestre los apellidos de los empleados que pertenecen al departamento de Ventas.

```
set SERVEROUTPUT ON;
create or replace procedure EmpVentas
as cursor curEmpVentas is select apellido from EMP where emp.dept_no=30;
Apellido emp.apellido%type;
```

```

iterador number;
begin
open curEmpVentas;
iterador:=1;
Loop
  fetch curEmpVentas into Apellido;
  exit when curEmpVentas%notfound;
  DBMS_OUTPUT.PUT_Line(iterador||' - ' || Apellido);
  iterador:=iterador+1;
end loop;
IF iterador=0 then
  DBMS_OUTPUT.PUT_Line('No empleados.');
```

end if;

```

close curEmpVentas;
end;
begin
  EmpVentas;
end;
/
```

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running, saving, and other database operations, along with a timer showing 0,008 segundos. The main window is titled 'Hoja de Trabajo' and 'Generador de Consultas'. It displays a PL/SQL script with line numbers 1 through 19. The script defines a cursor, opens it, iterates through employees in department 30, and prints their names. Below the script, a status bar indicates 'Tarea terminada en 0,008 segundos'. At the bottom, a message states 'Procedure EMPVENTAS compilado'.

```

1  set SERVEROUTPUT ON;
2  create or replace procedure EmpVentas
3  as cursor curEmpVentas is select apellido from EMP where emp.dept_no=30;
4  Apellido emp.apellido%type;
5  iterador number;
6  begin
7  open curEmpVentas;
8  iterador:=1;
9  Loop
10   fetch curEmpVentas into Apellido;
11   exit when curEmpVentas%notfound;
12   DBMS_OUTPUT.PUT_Line(iterador||' - ' || Apellido);
13   iterador:=iterador+1;
14 end loop;
15 IF iterador=0 then
16   DBMS_OUTPUT.PUT_Line('No empleados.');
```

end if;

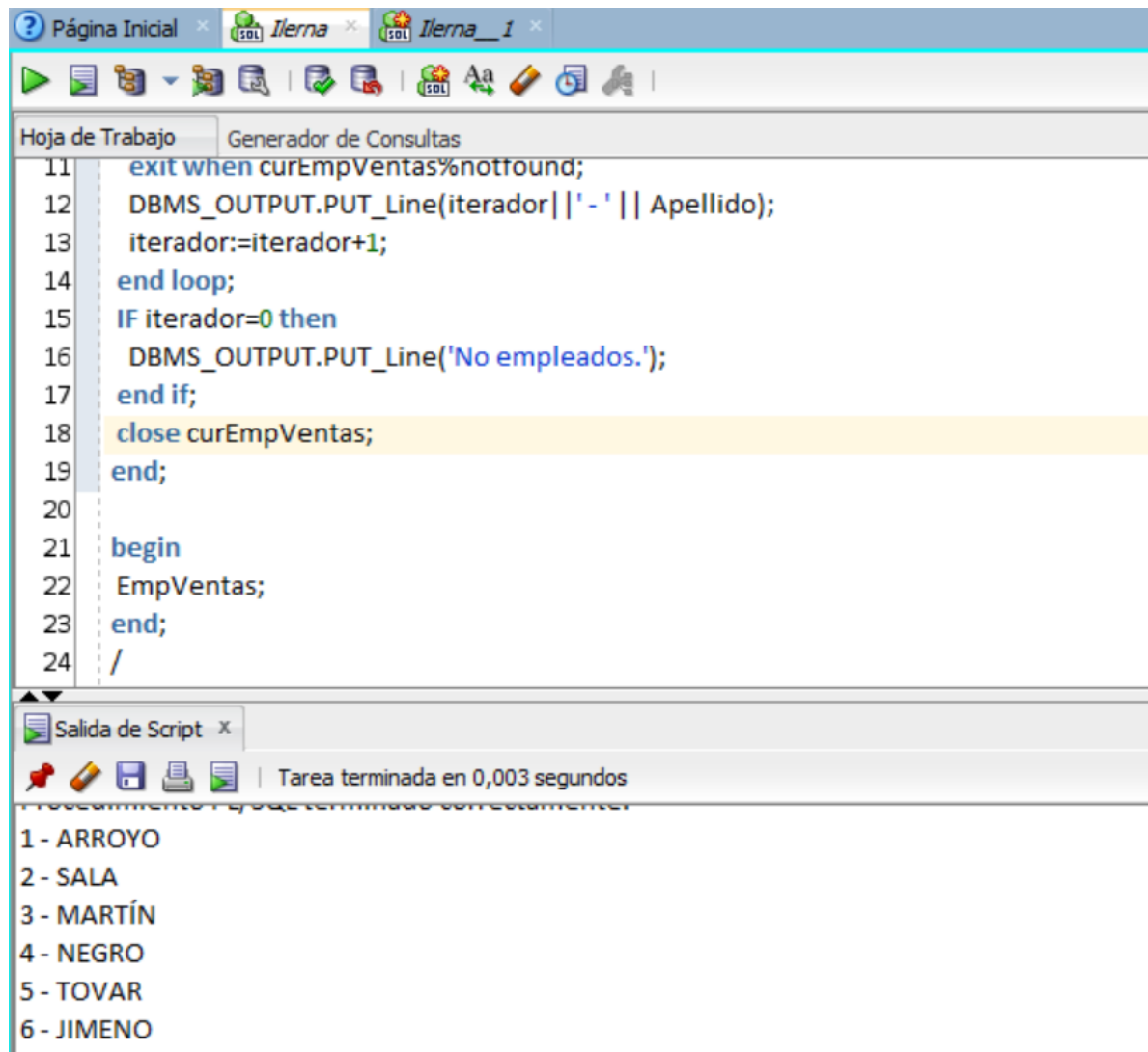
```

18 close curEmpVentas;
19 end;
```

Salida de Script x

Tarea terminada en 0,008 segundos

Procedure EMPVENTAS compilado



3. Crear un procedimiento que visualice los apellidos de los empleados del departamento que se recibe como argumento. Una versión con for y otra sin él.

Versión sin FOR:

```

set SERVEROUTPUT ON;
create or replace procedure EmpVentas (numeroDept number)
as cursor curEmpVentas is select apellido from EMP where emp.dept_no=numeroDept;
Apellido emp.apellido%type;
iterador number;
begin
  open curEmpVentas;
  iterador:=1;
  Loop
    fetch curEmpVentas into Apellido;
    exit when curEmpVentas%notfound;
    DBMS_OUTPUT.PUT_Line(iterador || ' - ' || Apellido);
    iterador:=iterador+1;
  end loop;
  IF iterador=0 then

```

```

DBMS_OUTPUT.PUT_Line('No empleados.');
```

```

end if;
close curEmpVentas;
end;
```

The screenshot shows the SQL Developer interface with the 'Generador de Consultas' (Query Generator) window. The code editor displays the following PL/SQL code:

```

1  set SERVEROUTPUT ON;
2  create or replace procedure EmpDepartamento (numeroDept number)
3  as cursor curEmpVentas is select apellido from EMP where emp.dept_no=numeroDept;
4  Apellido emp.apellido%type;
5  iterador number;
6  begin
7  open curEmpVentas;
8  iterador:=1;
9  Loop
10 fetch curEmpVentas into Apellido;
11 exit when curEmpVentas%notfound;
12 DBMS_OUTPUT.PUT_Line(iterador || ' - ' || Apellido);
13 iterador:=iterador+1;
14 end loop;
15 IF iterador=0 then
16 DBMS_OUTPUT.PUT_Line('No empleados.');
```

```

17 end if;
18 close curEmpVentas;
19 end;
```

Below the code editor, the 'Salida de Script' (Script Output) window shows the message: 'Tarea terminada en 0 segundos' (Task completed in 0 seconds). Below that, the status bar indicates: 'Procedimiento EMPDEPARTAMENTO compilado' (EMPDEPARTAMENTO procedure compiled).

The screenshot shows the SQL Developer interface with the 'Generador de Consultas' (Query Generator) window. The code editor displays the following PL/SQL code:

```

13 iterador:=iterador+1;
14 end loop;
15 IF iterador=0 then
16 DBMS_OUTPUT.PUT_Line('No empleados.');
```

```

17 end if;
18 close curEmpVentas;
19 end;
```

```

20
21 begin
22 EmpDepartamento(20);
23 end;
```

```

24 /
```

Below the code editor, the 'Salida de Script' (Script Output) window shows the message: 'Tarea terminada en 0 segundos' (Task completed in 0 seconds). Below that, the status bar indicates: 'Procedimiento PL/SQL terminado correctamente.' (PL/SQL procedure terminated correctly).

The output window displays the following results:

```

1 - SÁNCHEZ
2 - JIMÉNEZ
3 - GIL
4 - ALONSO
5 - FERNÁNDEZ
```

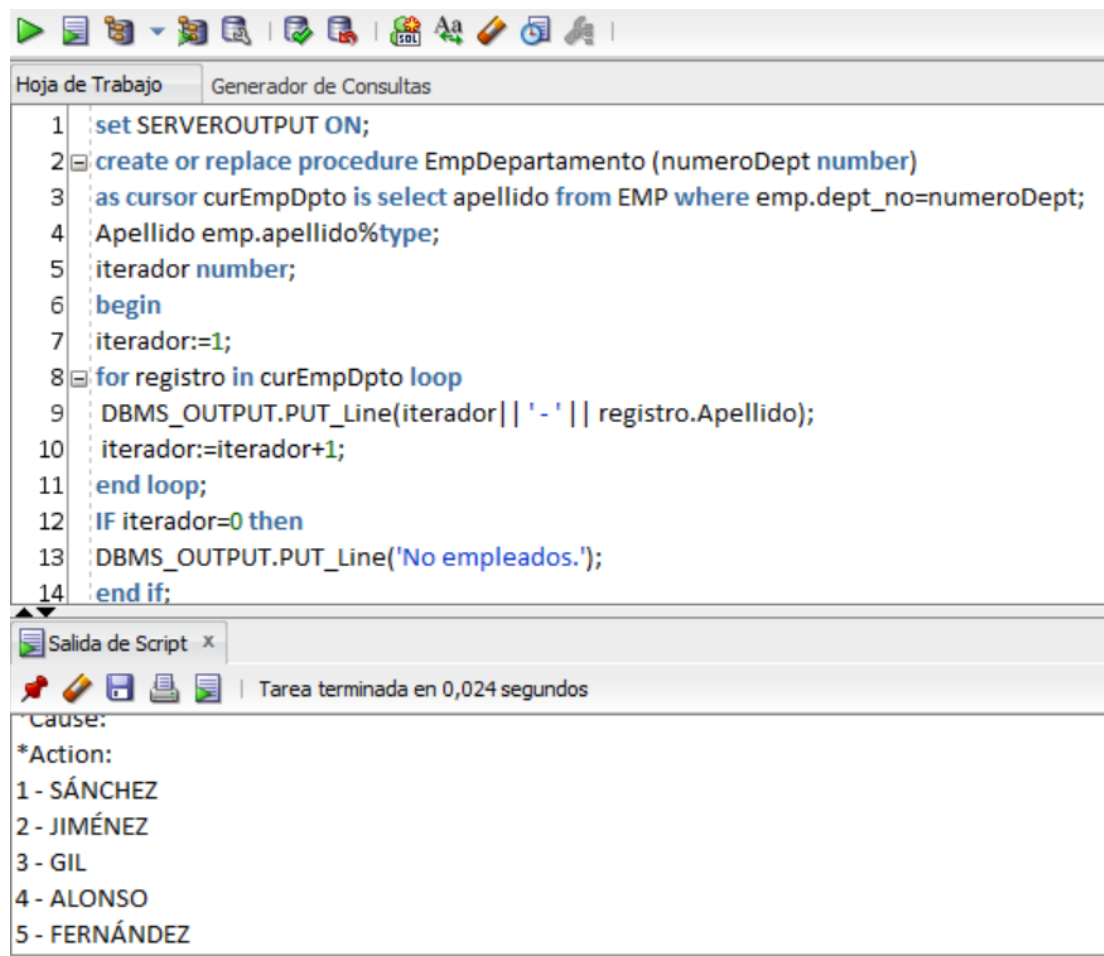
Versión con FOR:

```
set SERVEROUTPUT ON;
create or replace procedure EmpDepartamento (numeroDept number)
as cursor curEmpDpto is select apellido from EMP where emp.dept_no=numeroDept;
Apellido emp.apellido%type;
iterador number;
begin
iterador:=1;
for registro in curEmpDpto loop
  DBMS_OUTPUT.PUT_Line(iterador || ' - ' || registro.Apellido);
  iterador:=iterador+1;
end loop;
IF iterador=0 then
DBMS_OUTPUT.PUT_Line('No empleados.');
```

end if;

```
close curEmpDpto;
end;

begin
EmpDepartamento (20);
end;
/
```



The screenshot shows the SQL Developer interface. The top toolbar includes icons for running, saving, and other database operations. The main window is titled 'Hoja de Trabajo' and 'Generador de Consultas'. It contains a PL/SQL procedure named 'EmpDepartamento' that takes a department number as input and outputs the names of employees in that department. The procedure uses a cursor to fetch employee names and a loop to iterate through them. The output is displayed in the 'Salida de Script' window at the bottom, showing the names of five employees: SÁNCHEZ, JIMÉNEZ, GIL, ALONSO, and FERNÁNDEZ. The task was completed in 0.024 seconds.

```
1 set SERVEROUTPUT ON;
2 create or replace procedure EmpDepartamento (numeroDept number)
3 as cursor curEmpDpto is select apellido from EMP where emp.dept_no=numeroDept;
4 Apellido emp.apellido%type;
5 iterador number;
6 begin
7 iterador:=1;
8 for registro in curEmpDpto loop
9   DBMS_OUTPUT.PUT_Line(iterador || ' - ' || registro.Apellido);
10  iterador:=iterador+1;
11 end loop;
12 IF iterador=0 then
13 DBMS_OUTPUT.PUT_Line('No empleados.');
```

end if;

Salida de Script x

Tarea terminada en 0,024 segundos

Cause:

\*Action:

1 - SÁNCHEZ

2 - JIMÉNEZ

3 - GIL

4 - ALONSO

5 - FERNÁNDEZ

The screenshot shows the Oracle SQL Developer interface. The top toolbar contains various icons for file operations, execution, and formatting. Below the toolbar, there are two tabs: 'Hoja de Trabajo' (Worksheet) and 'Generador de Consultas' (Query Generator). The 'Hoja de Trabajo' tab is active, displaying a PL/SQL script with line numbers 8 through 21. The script is as follows:

```

8 for registro in curEmpDpto loop
9   DBMS_OUTPUT.PUT_Line(iterador || ' - ' || registro.Apellido);
10  iterador:=iterador+1;
11 end loop;
12 IF iterador=0 then
13   DBMS_OUTPUT.PUT_Line('No empleados.');
```

```

14 end if;
15 close curEmpDpto;
16 end;
17
18 begin
19   EmpDepartamento (20);
20 end;
21 /
```

Below the script editor, there is a 'Salida de Script' (Script Output) window. It shows the status 'Tarea terminada en 0,024 segundos' (Task completed in 0.024 seconds). Under the 'Cause:' label, the output of the script is listed:

```

Cause:
*Action:
1 - SÁNCHEZ
2 - JIMÉNEZ
3 - GIL
4 - ALONSO
5 - FERNÁNDEZ
```

4. Crear una función que sirva para calcular cuantos empleados del departamento cobran comisión (comisión >0).

set serveroutput on;

DECLARE

function comision (numDpto number) return number  
is

cursor empleadosConComision( numDpto number ) is SELECT COMISION FROM EMP WHERE  
EMP.DEPT\_NO=numDpto;

empComision EMP.COMISION%type;

contador number;

begin

contador := 0;

open empleadosConComision(numDpto);

Loop

FETCH empleadosConComision INTO empComision;

exit when empleadosConComision%notfound;

if( empComision > 0 ) then

contador := contador + 1;

end if;

end Loop;

close empleadosConComision;

return contador;

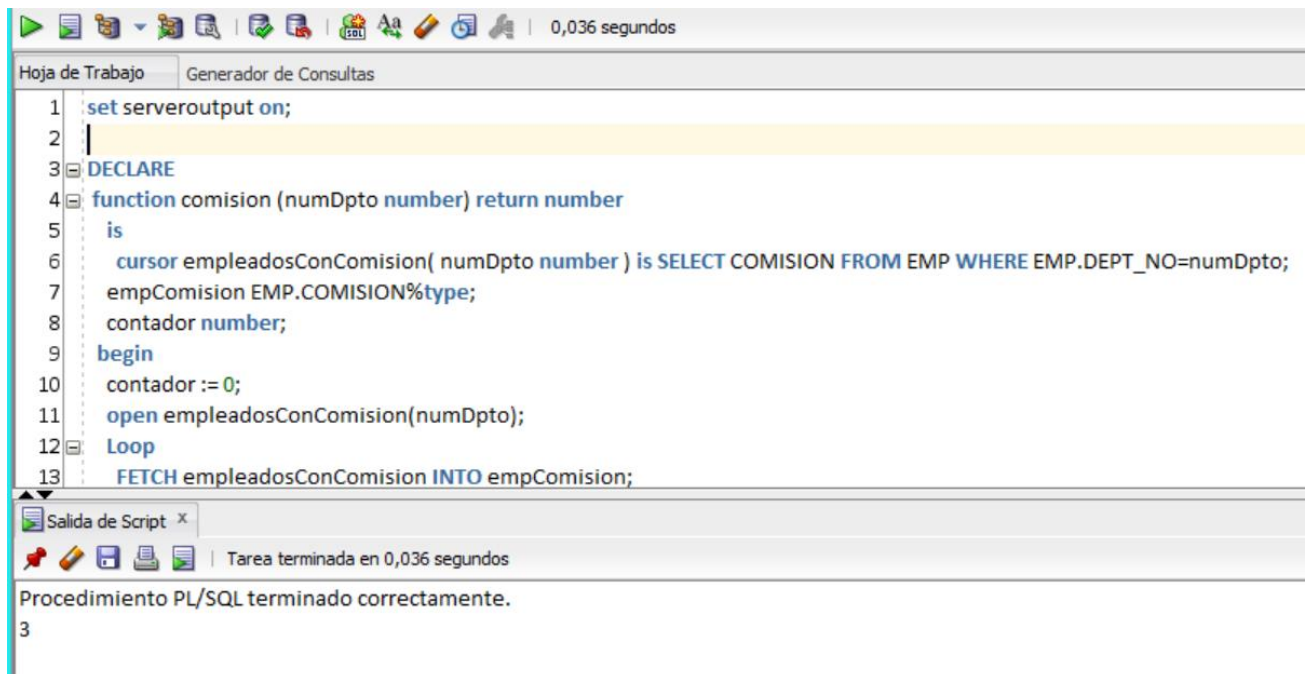
end;

begin

DBMS\_OUTPUT.PUT\_LINE(comision(30));

end;

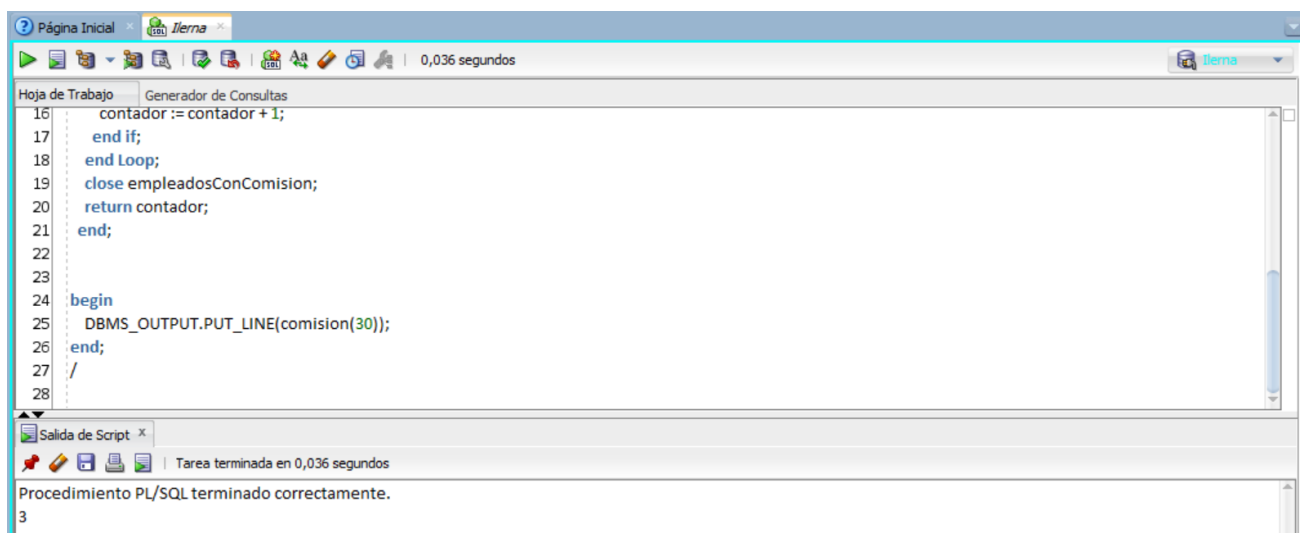
/



The screenshot shows the SQL Developer interface with a script editor. The script contains the following code:

```
1 set serveroutput on;
2
3 DECLARE
4 function comision (numDpto number) return number
5 is
6     cursor empleadosConComision( numDpto number ) is SELECT COMISION FROM EMP WHERE EMP.DEPT_NO=numDpto;
7     empComision EMP.COMISION%type;
8     contador number;
9 begin
10     contador := 0;
11     open empleadosConComision(numDpto);
12 Loop
13     FETCH empleadosConComision INTO empComision;
```

The status bar at the bottom indicates "Procedimiento PL/SQL terminado correctamente." and "3" lines of code.



The screenshot shows the SQL Developer interface with a script editor. The script contains the following code:

```
16     contador := contador + 1;
17 end if;
18 end Loop;
19 close empleadosConComision;
20 return contador;
21 end;
22
23
24 begin
25     DBMS_OUTPUT.PUT_LINE(comision(30));
26 end;
27 /
28
```

The status bar at the bottom indicates "Procedimiento PL/SQL terminado correctamente." and "3" lines of code.

5. Crear un procedimiento que utilizando la función anterior muestre el código y el nombre de los departamentos en los que menos de la mitad de sus empleados cobran comisión.

set serveroutput on;

DECLARE

create or replace procedure menosComisiones as

```

        cursor deptos is SELECT * FROM DEPT;
        filaDpto DEPT%rowtype;
        nEmpleados number;
begin
    for filaDpto in deptos Loop
        SELECT      COUNT(*)      INTO      nEmpleados      from      EMP      WHERE
EMP.DEPT_NO=filaDpto.DEPT_NO;
        if nEmpleados/2 > comision(filaDpto.DEPT_NO) then
            dbms_output.put_line(filaDpto.DEPT_NO      ||      '.....'      ||
filaDpto.DNOMBREBRE);
        end if;
    end loop;
end;

begin
    menosComisiones();
end;

```

The screenshot shows the SQL Developer interface with the following components:

- Top Bar:** Includes a search icon, a tab labeled 'Página Inicial', and a status bar showing '0,015 segundos'.
- Left Panel:** Labeled 'Hoja de Trabajo' and 'Generador de Consultas'.
- Script Editor:** Contains the following PL/SQL code:
 

```

1 set serveroutput on;
2
3 DECLARE
4 function comision (numDpto number) return number
5 is
6 cursor empleadosConComision( numDpto number ) is SELECT COMISION FROM EMP WHERE EMP.DEPT_NO=numDpto;
7 empComision EMP.COMISION%type;
8 contador number;
9 begin
10 contador := 0;
11 open empleadosConComision(numDpto);
      
```
- Output Window:** Labeled 'Salida de Script', it shows the message 'Procedimiento PL/SQL terminado correctamente.' followed by the query results:
 

```

10....CONTABILIDAD
20....INVESTIGACIÓN
      
```



```

20  return contador;
21  end;
22
23  procedure menosComisiones as
24  cursor deptos is SELECT * FROM DEPT;
25  filaDpto DEPT%rowtype;
26  nEmpleados number;
27  begin
28  for filaDpto in deptos Loop
29  SELECT COUNT(*) INTO nEmpleados FROM EMP WHERE EMP.DEPT_NO=filaDpto.DEPT_NO;
30  if nEmpleados/2 > comision(filaDpto.DEPT_NO) then dbms_output.put_line(filaDpto.DEPT_NO || '....' || filaDpto.DNOMBREBRE);

```

Salida de Script x

Tarea terminada en 0,015 segundos

Procedimiento PL/SQL terminado correctamente.  
10....CONTABILIDAD  
20....INVESTIGACIÓN

```

28  for filaDpto in deptos Loop
29  SELECT COUNT(*) INTO nEmpleados FROM EMP WHERE EMP.DEPT_NO=filaDpto.DEPT_NO;
30  if nEmpleados/2 > comision(filaDpto.DEPT_NO) then dbms_output.put_line(filaDpto.DEPT_NO || '....' || filaDpto.DNOMBREBRE);
31  end if;
32  end loop;
33  end;
34
35  begin
36  menosComisiones();
37  end;
38  /

```

Salida de Script x

Tarea terminada en 0,015 segundos

Procedimiento PL/SQL terminado correctamente.  
10....CONTABILIDAD  
20....INVESTIGACIÓN

6. Crear un programa que muestre por cada departamento una línea con el número de empleados y la suma de los salarios del departamento. A continuación, la lista de los apellidos y salarios de los empleados de este departamento.

```
set SERVEROUTPUT ON;
```

```
DECLARE
```

```
procedure listadoEmpleados as
```

```
    cursor departamentos is SELECT * FROM DEPT;
```

```
    departamento DEPT%rowtype;
```

```
    cursor empleados(nDpto number) is SELECT * FROM EMP WHERE EMP.DEPT_NO=nDpto;
```

```
    empleado EMP%rowtype;
```

```
    totalEmpleados number :=0;
```

```
    totalSalarios number :=0;
```

```
    totalEmpleadosDpto number;
```

```

totalSalariosDpto number;

begin
-- totalSalarios:=0;
-- totalEmpleados:=0;

    for departamento in departamentos LOOP
        SELECT COUNT(EMP.EMP_NO), SUM(EMP.SALARIO) INTO totalEmpleadosDpto,
totalSalariosDpto FROM EMP WHERE EMP.DEPT_NO=departamento.DEPT_NO;
        dbms_output.put_line('Depto: ' || departamento.DEPT_NO || 'Total Empleados' ||
totalEmpleadosDpto || 'Total Salarios' || totalSalariosDpto);

        for empleado in empleados(departamento.DEPT_NO) loop
            dbms_output.put_line(empleado.apellido || '...' || empleado.SALARIO);
        end loop;

        totalEmpleados:=totalEmpleados + totalEmpleadosDpto;
        totalSalarios:=totalSalarios + totalSalariosDpto;
    end loop;

    dbms_output.put_line( 'Total Empleados:' || totalEmpleados || 'totalSalarios: ' ||
totalSalarios );

end;

begin
listadoEmpleados();

end;
/

```

0,031 segundos

Hoja de Trabajo    Generador de Consultas

```
1 set SERVEROUTPUT ON;
2 DECLARE
3 procedure listadoEmpleados as
4 cursor departamentos is SELECT * FROM DEPT;
5 departamento DEPT%rowtype;
6 cursor empleados(nDpto number) is SELECT * FROM EMP WHERE EMP.DEPT_NO=nDpto;
7 empleado EMP%rowtype;
8
9 totalEmpleados number :=0;
10 totalSalarios number :=0;
11 totalEmpleadosDpto number;
```

Salida de Script x

Tarea terminada en 0,031 segundos

Procedimiento PL/SQL terminado correctamente.  
Depto: 10Total Empleados3Total Salarios1137500  
CEREZO....318500  
REY....650000  
MUÑOZ....169000  
Depto: 20Total Empleados5Total Salarios1413750  
SÁNCHEZ....104000  
MUÑOZ....169000

Página Inicial x Ilerna x Ilerna\_1 x

0,031 segundos

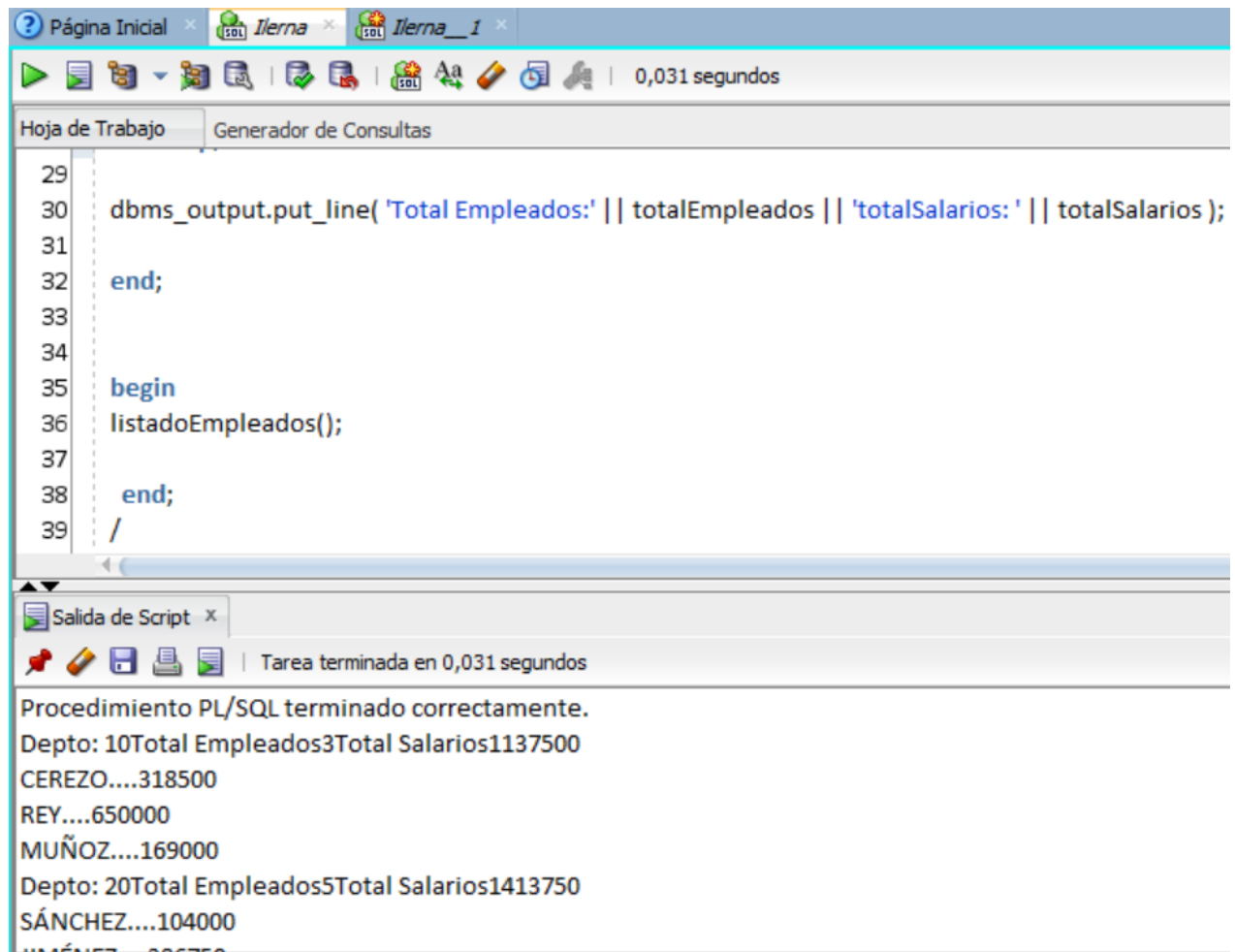
Hoja de Trabajo    Generador de Consultas

```
12 totalSalariosDpto number;
13
14 begin
15 -- totalSalarios:=0;
16 -- totalEmpleados:=0;
17
18 for departamento in departamentos LOOP
19 SELECT COUNT(EMP.EMP_NO), SUM(EMP.SALARIO) INTO totalEmpleadosDpto, totalSalariosDpto FROM EMP WHERE EMP.DEPT_NO=departamento.
20 dbms_output.put_line('Depto: ' || departamento.DEPT_NO || 'Total Empleados' || totalEmpleadosDpto || 'Total Salarios' || totalSalariosDpto);
21
22 for empleado in empleados(departamento.DEPT_NO) loop
23 dbms_output.put_line(empleado.EMP_NO || 'Empleado SALARIO: ' || empleado.SALARIO);
24 end loop;
25 end loop;
```

Salida de Script x

Tarea terminada en 0,031 segundos

Procedimiento PL/SQL terminado correctamente.  
Depto: 10Total Empleados3Total Salarios1137500  
CEREZO....318500  
REY....650000  
MUÑOZ....169000  
Depto: 20Total Empleados5Total Salarios1413750  
SÁNCHEZ....104000  
MUÑOZ....169000



7. Crear un procedimiento que reciba el nombre de un departamento y el porcentaje de subida de salario. El procedimiento incrementara el salario.

```

set SERVEROUTPUT ON;
CREATE or REPLACE
procedure aumentoSueldo
(nomDpto VARCHAR, aumento number) as
    cursor curEmpleados(nombreDpto VARCHAR) is SELECT SALARIO
    FROM EMP WHERE DEPT_NO=(SELECT DEPT_NO FROM DEPT WHERE
    DNOMBREBRE LIKE nomDpto) FOR UPDATE;
    salarioAnterior number;
begin
    for salarioAnterior in curEmpleados(nomDpto) loop
        UPDATE EMP
        set SALARIO=SALARIO*(1+aumento) WHERE current of curEmpleados;
    end loop;
end;

begin
aumentoSueldo ('VENTAS', 2);
END;

```

Página Inicial x Ilerna x Ilerna\_\_1 x

0,18700001 segundos

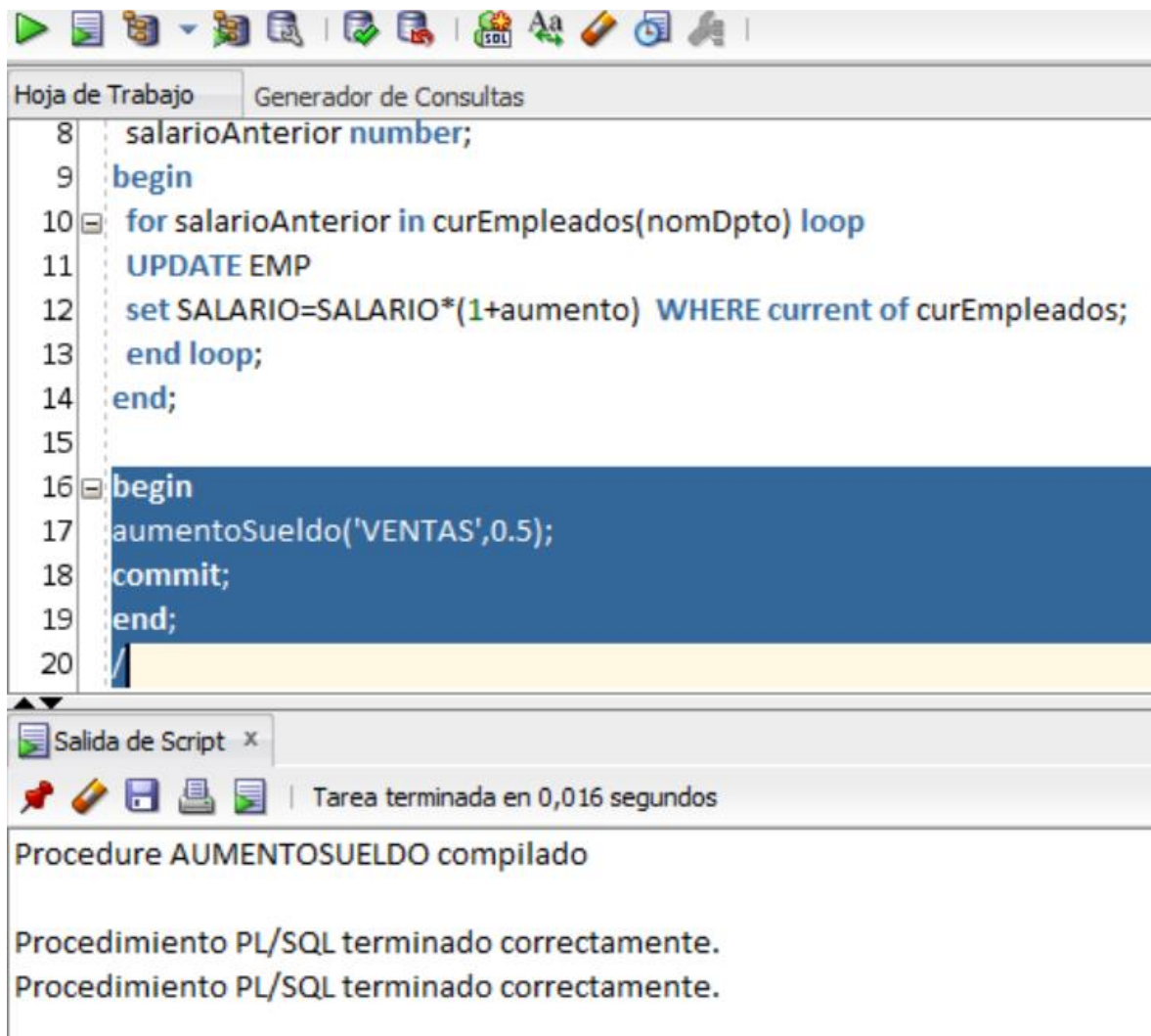
Hoja de Trabajo Generador de Consultas

```
2 CREATE or REPLACE
3 procedure aumentoSueldo
4 (nomDpto VARCHAR, aumento number) as
5     cursor curEmpleados(nombreDpto VARCHAR) is SELECT SALARIO
6     FROM EMP WHERE DEPT_NO=(SELECT DEPT_NO FROM DEPT WHERE
7     DNOMBREBRE LIKE nomDpto) FOR UPDATE;
8     salarioAnterior number;
9     begin
10     for salarioAnterior in curEmpleados(nomDpto) loop
11         UPDATE EMP
12         set SALARIO=SALARIO*(1+aumento) WHERE current of curEmpleados;
13     end loop;
14 end;
```

Salida de Script x

Tarea terminada en 0,187 segundos

Procedure AUMENTOSUELDO compilado



The screenshot displays the Oracle SQL Developer environment. The top toolbar contains various icons for file operations, execution, and formatting. The main window is titled 'Hoja de Trabajo' and 'Generador de Consultas'. It contains a PL/SQL script with the following code:

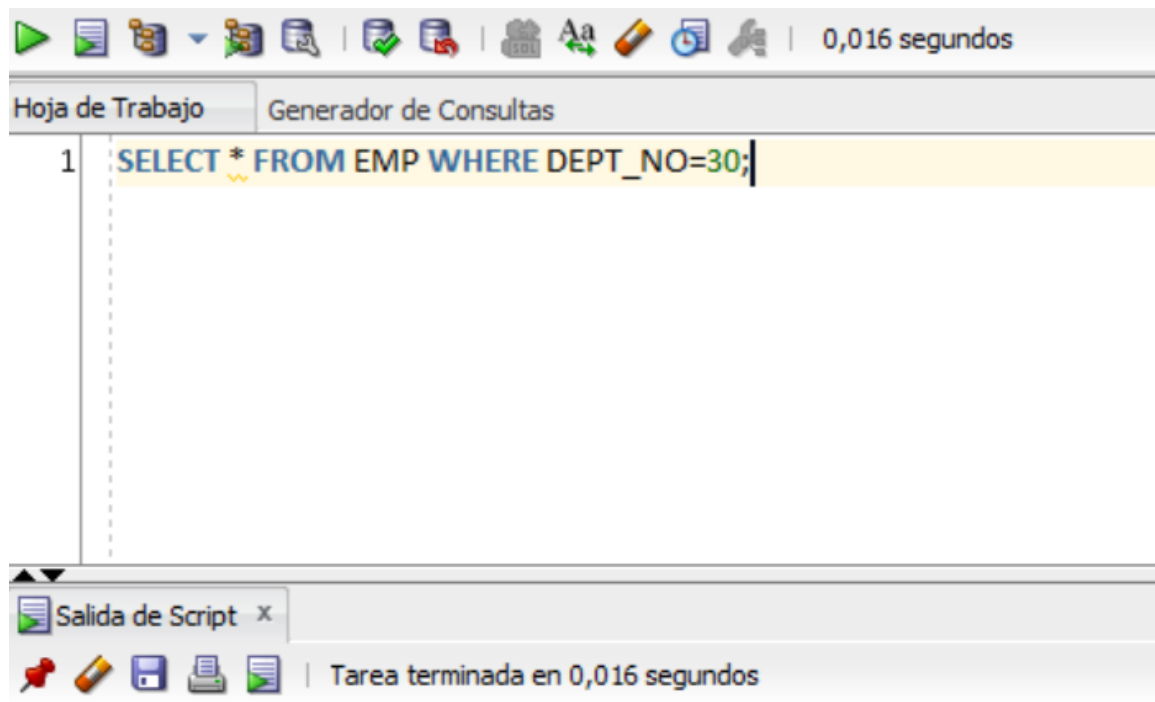
```
8  salarioAnterior number;  
9  begin  
10 for salarioAnterior in curEmpleados(nomDpto) loop  
11  UPDATE EMP  
12  set SALARIO=SALARIO*(1+aumento) WHERE current of curEmpleados;  
13  end loop;  
14 end;  
15  
16 begin  
17 aumentoSueldo('VENTAS',0.5);  
18 commit;  
19 end;  
20 /
```

Below the script editor, a 'Salida de Script' window shows the execution results:

Tarea terminada en 0,016 segundos

Procedure AUMENTOSUELDO compilado

Procedimiento PL/SQL terminado correctamente.  
Procedimiento PL/SQL terminado correctamente.



6 filas seleccionadas

8. Crear un procedimiento que reciba el numero de empleado y la cantidad que se incrementa el salario del empleado correspondiente.

```

set SERVEROUTPUT ON;
create or replace procedure subeSueldo(nEmpleado EMP.EMP_NO%type, aumento number) as
  esCero exception;
  salarioAnterior emp.SALARIO%type;
begin
  SELECT SALARIO INTO salarioAnterior FROM EMP WHERE EMP_NO=nEmpleado;
  if salarioAnterior is null then raise esCero;
  end if;
  UPDATE      EMP      SET      EMP.SALARIO=EMP.SALARIO+aumento      WHERE
EMP.EMP_NO=nEmpleado;

  exception
  when no_data_found then dbms_output.put_line ('no hay datos');
  when esCero then dbms_output.put_line ('salario es cero');
end;

subeSueldo (7369,3);
select * from emp where emp_no=7369;

```



The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running, saving, and other database operations. The main window is titled 'Hoja de Trabajo' and 'Generador de Consultas'. It contains a PL/SQL script with the following code:

```

1 set SERVEROUTPUT ON;
2 create or replace procedure subeSueldo(nEmpleado EMP.EMP_NO%type, aumento number) as
3   esCero exception;
4   salarioAnterior emp.SALARIO%type;
5 begin
6   SELECT SALARIO INTO salarioAnterior FROM EMP WHERE EMP_NO=nEmpleado;
7   if salarioAnterior is null then raise esCero;
8   end if;
9   UPDATE EMP SET EMP.SALARIO=EMP.SALARIO+aumento WHERE EMP.EMP_NO=nEmpleado;
10
11 exception
12 when no_data_found then dbms_output.put_line ('no hay datos');
13 when esCero then dbms_output.put_line ('salario es cero');
14 end;
15
16 subeSueldo (7369,3);
17 select * from emp where emp_no=7369;

```

Below the script, the 'Resultado de la Consulta' tab shows the output of the query. It indicates that 1 row was recovered in 0.015 seconds. The result is displayed in a table with the following columns: EMP\_NO, APELLIDO, OFICIO, JEFE, FECHA\_ALTA, SALARIO, COMISION, and DEPT\_NO.

EMP_NO	APELLIDO	OFICIO	JEFE	FECHA_ALTA	SALARIO	COMISION	DEPT_NO
1	7369 SÁNCHEZ	EMPLEADO	7902	17/12/80	104006	(null)	20

9. Crear un procedimiento que permita subir el sueldo de todos los empleados que ganen menos que el salario medio de su oficio. La subida será del 50% de la diferencia entre el salario del empleado y la media de su oficio.

```

set SERVEROUTPUT ON;
create or replace procedure compensaSueldos as
  cursor miCursor is SELECT EMP.SALARIO, EMP.OFICIO FROM EMP FOR UPDATE;
  salarioMedio number(10,3);
  subida number(10,3);
  fila miCursor%rowtype;
begin
  FOR fila in miCursor
  loop
    SELECT AVG(SALARIO), (AVG(SALARIO)/2) INTO salarioMedio, subida FROM EMP
    WHERE OFICIO=fila.OFICIO;
    --dbms_output.put_line(salarioMedio || ' --- ' || subida);
    if fila.SALARIO < salarioMedio then
      UPDATE EMP SET SALARIO=SALARIO+subida WHERE CURRENT OF miCursor;
    end if;
  end loop;
exception
  WHEN no_data_found then dbms_output.put_line('No hay datos disponibles');
end;

```



```

begin
COMPENSASUELDOS;
commit;
end;
/

```

0,007 segundos

Hoja de Trabajo | Generador de Consultas

```

1  set SERVEROUTPUT ON;
2  create or replace procedure compensaSueLDos as
3  cursor miCursor is SELECT EMP.SALARIO, EMP.OFICIO FROM EMP FOR UPDATE;
4  salarioMedio number(10,3);
5  subida number(10,3);
6  fila miCursor%rowtype;
7  begin
8  FOR fila in miCursor
9  loop
10     SELECT AVG(SALARIO), (AVG(SALARIO)/2) INTO salarioMedio, subida FROM EMP WHERE OFICIO=fila.OFICIO;
11     --dbms_output.put_line(salarioMedio || '---' || subida);
12     if fila.SALARIO < salarioMedio then
13     UPDATE EMP SET SALARIO=SALARIO+subida WHERE CURRENT OF miCursor;
14     end if;
15     end loop;
16     exception
17     WHEN no_data_found then dbms_output.put_line('No hay datos disponibles');
18 end;
19

```

Salida de Script x

Tarea terminada en 0,007 segundos

Procedure COMPENSASUELDOS compilado

Hoja de Trabajo | Generador de Consultas

```

8  FOR fila in miCursor
9  loop
10     SELECT AVG(SALARIO), (AVG(SALARIO)/2) INTO salarioMedio, subida FROM EMP WHERE OFICIO=fila.OFICIO
11     --dbms_output.put_line(salarioMedio || '---' || subida);
12     if fila.SALARIO < salarioMedio then
13     UPDATE EMP SET SALARIO=SALARIO+subida WHERE CURRENT OF miCursor;
14     end if;
15     end loop;
16     exception
17     WHEN no_data_found then dbms_output.put_line('No hay datos disponibles');
18 end;
19
20
21 begin
22 COMPENSASUELDOS;
23 commit;
24 end;
25 /

```

Salida de Script x

Tarea terminada en 0,012 segundos

Procedure COMPENSASUELDOS compilado

Procedimiento PL/SQL terminado correctamente.

