

-----1-----
@ 'C:\hlocal\BBDD\crear_tablas.sql'

-----2-----
declare
 subtype t_sueldo is BINARY_INTEGER range 900 .. 5000 not null;
 v_sueldo t_sueldo := 1000;
begin
 select SUELDO into v_sueldo from EMPLEADOS where DNI = '12345678L';
 --v_sueldo := 4*v_sueldo; --Da error porque se sale del rango de la
variable
 v_sueldo := 1.1*v_sueldo;
 update empleados set sueldo = v_sueldo where DNI = '12345678L';
 rollback;
end;

Procedimiento PL/SQL terminado correctamente.

-----3-----

create or replace procedure pr_empleados_tlf (tlf teléfonos.teléfono%type)
is
 v_nom empleados.nombre%type;
 v_dni empleados.dni%type;
begin
 select nombre, dni into v_nom, v_dni from teléfonos natural join empleados
where teléfono = tlf;
 DBMS_OUTPUT.put_line('El empleado con teléfono '||tlf||' es: '||v_nom||',
con DNI:'||v_dni);
exception
 when no_data_found then
 DBMS_OUTPUT.put_line('No se encontró el empleado con el teléfono
'||tlf);
 when too_many_rows then
 DBMS_OUTPUT.put_line('Hay más de un empleado con el teléfono '||tlf);
end;

SET SERVEROUTPUT ON SIZE 1000000;
declare
 v_tlf teléfonos."TELÉFONO"%type;
begin
 v_tlf := '666666666';
 pr_empleados_tlf(v_tlf);
 v_tlf := '611111111';
 pr_empleados_tlf(v_tlf);
 v_tlf := '913333333';
 pr_empleados_tlf(v_tlf);
end;

No se encontró el empleado con el teléfono 666666666
El empleado con teléfono 611111111 es: Carlota Cerezo, con DNI:12345678C
Hay más de un empleado con el teléfono 913333333

Procedimiento PL/SQL terminado correctamente.

```

create or replace procedure pr_comprobar_poblaciones
is
    v_pobl "Códigos postales".población%type;
    excep exception;
    cont integer;
    cursor c_pob is select población, count(*) from (select distinct población,
provincia from "Códigos postales") group by población;
begin
    open c_pob;
    loop
        fetch c_pob into v_pobl, cont;
        if cont > '1' then
            raise excep;
        end if;
        exit when c_pob%notfound;
    end loop;
    DBMS_OUTPUT.put_line('No hay dos o más provincias que compartan la misma
población.');
```

close c_pob;

```

exception
    when excep then
        DBMS_OUTPUT.put_line('A la población '||v_pobl||' no le corresponde
siempre la misma provincia');
```

end;

SET SERVEROUTPUT ON SIZE 1000000;

```

begin
    pr_comprobar_poblaciones;
    insert into "Códigos postales" values ('41010', 'Arganda', 'Sevilla');
    pr_comprobar_poblaciones;
    delete from "Códigos postales" where provincia = 'Sevilla';
end;
```

No hay dos o más provincias que compartan la misma población.
A la población Arganda no le corresponde siempre la misma provincia

Procedimiento PL/SQL terminado correctamente.

```

create or replace procedure pr_empleados_CP
is
    v_cp domicilios."Código postal"%type;
    v_avg empleados.sueldo%type;
    v_count number;
    cursor c_cp is select "Código postal", avg(sueldo), count(*) from domicilios
natural join empleados group by "Código postal" order by "Código postal";
begin
    open c_cp;
    loop
```

```

        fetch c_cp into v_cp, v_avg, v_count;
        DBMS_OUTPUT.put_line('Código postal: '||v_cp);
        pr_aux(v_cp);
        DBMS_OUTPUT.put_line('          Nº empleados: '||v_count||', Sueldo
medio: '||v_avg);
        exit when c_cp%notfound;
    end loop;
    close c_cp;
exception
    when no_data_found then
        null;
end;

create or replace procedure pr_aux(cp domicilios."Código postal"%type)
is
    v_nombre empleados.nombre%type;
    v_calle domicilios.calle%type;
    v_sueldo empleados.sueldo%type;
    cursor c_emp is select nombre, calle, sueldo from (domicilios join empleados
on domicilios.dni = empleados.dni) where domicilios."Código postal" = cp;
begin
    open c_emp;
    fetch c_emp into v_nombre, v_calle, v_sueldo;
    while c_emp%found loop
        DBMS_OUTPUT.put_line('    '||v_nombre||', '||v_calle||', '||v_sueldo);
        fetch c_emp into v_nombre, v_calle, v_sueldo;
    end loop;
    close c_emp;
exception
    when no_data_found then
        null;
end;

SET SERVEROUTPUT ON SIZE 1000000;

begin
    pr_empleados_CP;
end;

```

```

Código postal: 14200
  Laura López, Diamante, 1500
  Pedro Pérez, Diamante, 2000
    Nº empleados: 2, Sueldo medio: 1750
Código postal: 14900
  Pedro Pérez, Carbón, 2000
    Nº empleados: 1, Sueldo medio: 2000
Código postal: 28004
  Antonio Arjona, Cántaro, 5000
    Nº empleados: 1, Sueldo medio: 5000
Código postal: 28040
  Antonio Arjona, Avda. Complutense, 5000
    Nº empleados: 1, Sueldo medio: 5000
Código postal: 28040
  Antonio Arjona, Avda. Complutense, 5000

```

Nº empleados: 1, Sueldo medio: 5000

Procedimiento PL/SQL terminado correctamente.

-----6-----

```
create or replace function f_c_ba
return number
is
begin
    return 2.920050977316;
end;

create or replace function f_primo(p_n integer)
return number
is
begin
    if p_n = 1 then return f_c_ba;
    else return floor(f_primo(p_n -1))*(f_primo(p_n -1)-floor(f_primo(p_n
-1))+1);
    end if;
end;
```

SET SERVEROUTPUT ON SIZE 1000000;

```
declare
    n integer := 1;
begin
    loop
        DBMS_OUTPUT.put_line(floor(f_primo(n)));
        n := n+1;
        exit when n > 10;
    end loop;
end;
```

2
3
5
7
11
13
17
19
23
29

Procedimiento PL/SQL terminado correctamente.

-----7-----

```
create or replace procedure pr_jefes(p_DNI empleados.dni%type)
is
    v_jefe empleados.jefe%type;
    v_nombre empleados.nombre%type;
```

```
begin
    select jefe into v_jefe from empleados where dni = p_DNI;
    if v_jefe is not null then
        select nombre into v_nombre from empleados where dni = v_jefe;
        DBMS_OUTPUT.put_line(v_nombre||', '||v_jefe);
        pr_jefes(v_jefe);
    end if;
exception
    when no_data_found then
        DBMS_OUTPUT.put('El empleado con el dni proporcionado no existe.');
```

```
end;
```

SET SERVEROUTPUT ON SIZE 1000000;

```
declare
    v_dni empleados.dni%type := '12345678C';
begin
    pr_jefes(v_dni);
end;
```

Laura López, 12345678L
Pedro Pérez, 12345678P
Antonio Arjona, 12345678A

Procedimiento PL/SQL terminado correctamente.