

--Estíbaliz Busto y Lucía Latorre

SET SERVEROUTPUT ON SIZE 100000;

--1) Crear un procedimiento almacenado denominado "empleados_tlf" que tenga un número de teléfono como parámetro de entrada e imprima por pantalla el nombre y DNI del empleado al que pertenezca. Si no se encuentra el número, o si hay más de un empleado para el mismo teléfono, se debe indicar con un mensaje. Usa un cursor en línea para resolver este apartado y pruébalo en tres ejecuciones distintas con los números '666666666', '611111111' y '913333333'. El resultado sería similar a:

CREATE OR REPLACE PROCEDURE empleados_tlf (tel Teléfonos.teléfono%TYPE) AS

name Empleados.nombre%TYPE;

id Empleados.dni%TYPE;

CURSOR cursorTeléfono IS

SELECT nombre, DNI

FROM Empleados natural inner join Teléfonos

WHERE tel = teléfono;

BEGIN

OPEN cursorTeléfono;

LOOP

FETCH cursorTeléfono INTO name, id;

EXIT WHEN cursorTeléfono%NOTFOUND;

END LOOP;

IF cursorTeléfono%ROWCOUNT = 1 THEN

DBMS_OUTPUT.put_line ('El empleado con el teléfono ' || ltrim(rtrim (tel)) || ' es: ' ||
ltrim(rtrim (name)) || ', con DNI: ' || ltrim(rtrim (id)) || '.');

ELSIF cursorTeléfono%ROWCOUNT > 1 THEN

DBMS_OUTPUT.put_line ('Hay más de un empleado con el teléfono ' || ltrim(rtrim (tel))
|| '.');

```

ELSE
    DBMS_OUTPUT.put_line ('No se encontró el empleado con el teléfono ' || ltrim(rtrim
(tel)) || '.');
END IF;
CLOSE cursorTeléfono;

END;

```

--Procedure EMPLEADOS_TLF compilado

```
EXECUTE empleados_tlf('666666666');
```

/*No se encontró el empleado con el teléfono 666666666.

Procedimiento PL/SQL terminado correctamente.*/

```
EXECUTE empleados_tlf('611111111');
```

/*El empleado con el teléfono 611111111 es: Carlota Cerezo, con DNI: 12345678C.

Procedimiento PL/SQL terminado correctamente.*/

```
EXECUTE empleados_tlf('913333333');
```

/*Hay más de un empleado con el teléfono 913333333.

Procedimiento PL/SQL terminado correctamente.*/

--2) Crear un procedimiento almacenado denominado "comprobar_poblaciones" que genere una excepción para la primera población que encuentre a la que le corresponda más de una provincia. Tanto si se encuentra como si no, se emitirá un mensaje con el resultado y se terminará. Para probarlo, ejecútalo con los datos que tenga actualmente la tabla. Añade la tupla ('41008','Arganda','Sevilla') a la tabla "Códigos postales" y ejecútalo de nuevo. Finalmente, borra la tupla añadida. El resultado sería similar a:

/*No hay dos o más provincias que compartan la misma población.

A la población Arganda no le corresponde siempre la misma provincia. */

CREATE OR REPLACE PROCEDURE comprobar_poblaciones AS

CURSOR cursorProvincias IS

SELECT población, COUNT(provincia)

FROM "Códigos postales"

GROUP BY población;

pob "Códigos postales".población%TYPE;

num_prov NUMBER;

MiExcepcion EXCEPTION;

BEGIN

OPEN cursorProvincias;

LOOP

FETCH cursorProvincias INTO pob, num_prov;

EXIT WHEN cursorProvincias%NOTFOUND;

IF num_prov > 1 THEN

RAISE MiExcepcion;

END IF;

END LOOP;

CLOSE cursorProvincias;

DBMS_OUTPUT.put_line('No hay dos o más provincias que compartan la misma población');

EXCEPTION

WHEN MiExcepcion THEN

```
DBMS_OUTPUT.put_line('A la población ' || ltrim(rtrim (pob)) || ' no le corresponde siempre la misma provincia');
```

```
END;
```

```
--Procedure COMPROBAR_POBLACIONES compilado
```

```
EXECUTE comprobar_poblaciones;
```

```
/*No hay dos o más provincias que compartan la misma población
```

```
Procedimiento PL/SQL terminado correctamente.*/
```

```
INSERT INTO "Códigos postales" values ('41008','Arganda','Sevilla');
```

```
--1 fila insertadas.
```

```
EXECUTE comprobar_poblaciones;
```

```
DELETE FROM "Códigos postales" WHERE "Código postal" = '41008' AND Población = 'Arganda' AND Provincia = 'Sevilla';
```

--3) Crear un procedimiento almacenado denominado “empleados_CP” que liste el nombre, calle y sueldo de los empleados agrupados por código postal (solo deben aparecer códigos postales en los que resida al menos un empleado). Por cada código postal se mostrará el listado de los datos de sus empleados. Al final de cada grupo se indicará el número de empleados y el sueldo medio. Si algún empleado tiene más de un domicilio, se contabilizará tantas veces como domicilios tenga. Al final de todo el listado se indicará el total de empleados. El resultado sería similar a:

```
CREATE OR REPLACE PROCEDURE empleados_CP AS
```

```
CP "Códigos postales"."Código postal"%TYPE;
```

```
cod "Códigos postales"."Código postal"%TYPE;
```

```
nom Empleados.nombre%TYPE;
```

```

c Domicilios.calle%TYPE;

s Empleados.sueldo%TYPE;

av NUMBER;

numE NUMBER;

numTotal NUMBER;

cont NUMBER;

CURSOR cursorEmpleados IS

SELECT d."Código postal", e.Nombre, d.Calle, e.Sueldo

FROM Empleados e, Domicilios d, "Códigos postales" c

WHERE e.dni = d.dni AND d."Código postal" = c."Código postal";

CURSOR cursorCodPostales IS

SELECT "Código postal"

FROM Domicilios

GROUP BY "Código postal";


BEGIN

OPEN cursorCodPostales;

numTotal := 0;

LOOP

    FETCH cursorCodPostales INTO CP;

    EXIT WHEN cursorCodPostales%NOTFOUND;


    DBMS_OUTPUT.put_line('Código postal: ' || ltrim(rtrim (CP)));

    cont := 0;

    OPEN cursorEmpleados;

    LOOP

        FETCH cursorEmpleados INTO cod, nom, c, s;

        EXIT WHEN cursorEmpleados%NOTFOUND;


        IF CP = cod THEN

```

```

        DBMS_OUTPUT.put_line(' ' || ltrim(rtrim (nom)) || ', ' || ltrim(rtrim (c)) || ', ' ||
ltrim(rtrim (s)) );

        cont := cont + 1;

    END IF;

END LOOP;

CLOSE cursorEmpleados;

IF cont > 0 THEN

    SELECT COUNT(nombre) INTO numE FROM (SELECT DISTINCT nombre FROM (Empleados
natural join Domicilios) natural join "Códigos postales" WHERE "Código postal" = CP);

    SELECT avg(sueldo) INTO av FROM (SELECT DISTINCT nombre, sueldo FROM (Empleados
natural join Domicilios) natural join "Códigos postales" WHERE "Código postal" = CP);

    DBMS_OUTPUT.put_line(' Nº empleados: ' || ltrim(rtrim (numE)) || ', Sueldo medio: ' ||
ltrim(rtrim (av)));

    numTotal := numTotal + numE;

END IF;

END LOOP;

CLOSE cursorCodPostales;

DBMS_OUTPUT.put_line('Total empleados: ' || ltrim(rtrim (numTotal)));

END;

--Procedure EMPLEADOS_CP compilado

EXECUTE empleados_CP;

/*Código postal: 28004

Antonio Arjona, Cántaro, 5000

Nº empleados: 1, Sueldo medio: 5000

```

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: 28040

Antonio Arjona, Avda. Complutense, 5000

Nº empleados: 1, Sueldo medio: 5000

Total empleados: 5

Procedimiento PL/SQL terminado correctamente.*/