

# Práctica 4: El lenguaje de programación PL/SQL

## Objetivo

Practicar con el lenguaje de programación procedimental PL/SQL:

- Declaración de bloques y procedimientos almacenados.
- Declaración y uso de cursores.
- Uso de instrucciones de recorrido de cursores.
- Uso de instrucciones de control de PL/SQL.
- Uso de instrucciones de entrada/salida por pantalla.
- Manejo de excepciones.

## Introducción

Oracle es un gestor de bases de datos que data de finales de la década de 1970 y ha evolucionado en gran medida. Actualmente se usa en muchas empresas e instituciones (administración pública, banca, operadores de telefonía...). PL/SQL es su lenguaje de programación de cuarta generación, es decir, un lenguaje de propósito general y procedimental que tiene integrados los tipos y sus operaciones asociadas correspondientes a un SGBD concreto; en este caso, a Oracle. Además es posible ejecutar sentencias SQL directamente sin necesidad de usar ningún API. El acceso a los datos de las tablas se hace mediante cursores (un objeto que representa a un conjunto de datos extraídos mediante una instrucción **SELECT**) o incluso directamente en variables a partir de una **SELECT** (son los denominados cursores implícitos o en línea). Las instrucciones de PL/SQL se pueden escribir en determinados contextos: bloques, procedimientos almacenados y disparadores.

## Ejercicios

Como resultado de esta práctica se debe subir al CV un documento PDF con las instrucciones usadas, procedimientos escritos y resultados de las ejecuciones para cada uno de los apartados siguientes. Si incluyes en el mismo script la creación de varios procedimientos y los bloques para probarlos, termina cada uno de ellos con la barra de división "/" (cosas de Oracle...) Inicia sesión en SQL Developer en la instancia de base de datos "SRV tania03- db DBic" con el usuario de DESWeb. La contraseña es la misma que el usuario. Cámbiala a una nueva con:

```
ALTER USER CC2x_mail IDENTIFIED BY NUEVA_CONTRASEÑA;
```

- 1) Ejecutar el script `crear_tablas.sql` en la consola de SQL Developer con:

```
@'ruta\crear_tablas.sql'
```

donde **ruta** es la carpeta en la que hayas descargado el script.

- 2) Crear un procedimiento almacenado denominado **pr\_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. Prueba el procedimiento en tres ejecuciones distintas con los números '666666666', '611111111' y '913333333'. El resultado sería similar a:

Primera ejecución:

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

Segunda ejecución:

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

Tercera ejecución:

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

**Recuerda:** hay que habilitar la salida con: SET SERVEROUTPUT ON SIZE 1000000;

Sugerencia:

No es necesario declarar un cursor. Puedes usar SELECT INTO tratando las excepciones NO\_DATA\_FOUND y TOO\_MANY\_ROWS si la select no da filas o devuelve más de una fila.

- 3) Crear un procedimiento almacenado denominado **pr\_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:

Primera ejecución:

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

Segunda ejecución:

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

- 4) Crear un procedimiento almacenado denominado **pr\_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:

**EXECUTE empleados\_CP;**

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  
Total empleados: 5