

# Unidad VI: SQL procedural

## 6.1 Procedimientos almacenados

Un procedimiento es un subprograma que ejecuta una acción específica y que no devuelve ningún valor. Un procedimiento tiene un nombre, un conjunto de parámetros (opcional) y un bloque de código.

La sintaxis Oracle de un procedimiento almacenado es la siguiente:

```
CREATE [OR REPLACE]
PROCEDURE <nombre_procedure> [(<param1> [IN | OUT | IN OUT] <type>,
                                         <param2> [IN | OUT | IN OUT] <type>, ...)]
IS
-- Declaración de variables locales
BEGIN
-- Sentencias
[EXCEPTION]
-- Sentencias control de excepción
END [<nombre_procedure>];
```

## 6.2 Disparadores

Un disparador (o **trigger**) es un tipo especial de **procedimiento almacenado** asociado a una tabla que se ejecuta al realizar una operación “básica” (**INSERT**, un **DELETE** o un **UPDATE**) sobre ésta. La operación básica que despierta al trigger es conocida como **sentencia disparadora**.

La ejecución del disparador puede ser **antes (before)** o **después (after)** de llevar a cabo la sentencia disparadora. Es posible especificar condiciones adicionales para la ejecución del disparador (**restrictores**).

Dado que una sentencia disparadora puede afectar una o más filas de una tabla, es necesario especificar si se quiere que el disparador se ejecute para cada una de las filas afectadas o para el bloque en general.

Para diseñar un disparador hay que cumplir dos requisitos:

- Especificar las condiciones en las que se va a ejecutar el disparador. Esto se descompone en un **evento** que causa la comprobación del disparador y una **condición** que se debe cumplir para ejecutar el disparador.
- Especificar las acciones que se van a realizar cuando se ejecute el disparador.

Los triggers constituyen una potente herramienta para mantener la integridad de la base de datos, ya que pueden llevar a cabo cualquier acción que sea necesaria para el mantenimiento de dicha integridad.

Los triggers pueden llamar a otros procedimientos y disparar otros triggers, pero no admiten parámetros y no pueden ser invocados desde otros procedimientos.

Los disparadores se activan al crearlos.

Eliminar un disparador: **DROP TRIGGER** nombre\_disparador;

Activar/ Desactivar disparadores: Existen dos opciones.

**ALTER TRIGGER** nombre\_disparador {**DISABLE** | **ENABLE**};

**ALTER TABLE** nombre\_tabla {**ENABLE** | **DISABLE**} **ALL TRIGGERS**;

Temporalidad del Evento: AFTER / BEFORE

**BEFORE**: Ejecutan la acción asociada antes de que la sentencia sea ejecutada

- Decidir si la acción debe realizarse o no
- Utilizar valores alternativos para la sentencia