

Laboratorio 3

- Funciones
- Procedimientos almacenados
- Excepciones



Uno de los desafíos más grandes en nuestra profesión es encontrar la forma de reducir la complejidad en nuestro ambiente, y de esta manera proveer a los usuarios con aplicaciones que sean más fácil de usar y mucho más poderosas que sus predecesoras.

Esto se puede conseguir a través de la modularización del código en procedimientos, funciones y tipos de objetos.

Modularización es el proceso por el cual se separan grandes bloques de código en partes más pequeñas, llamadas módulos, que pueden ser llamadas por otros módulos. Con modularización el código se vuelve:

1. Más reusable.
2. Más legible.
3. Más fácil de administrar o manejar.
4. Más confiable.

Subprogramas

La forma particular de PL/SQL de ofrecer la capacidad de modularizar su código es por medio de subprogramas y estructuras de agrupación. Los subprogramas son bloques de PL/SQL nombrados (Named blocks, en inglés) con un identificador y que normalmente se almacenan en la propia base de datos para su posterior ejecución. Existen diferentes tipos de subprogramas, por ejemplo: procedimientos, funciones, triggers o subprogramas en bloques anónimos.

En esta guía nos centraremos principalmente en las funciones y los procedimientos almacenados.

Procedimientos almacenados

Un procedimiento almacenado es un modulo que realiza una o más acciones y que no devuelve ningún valor. A pesar de que no retorna ningún valor directamente, se puede simular. Un procedimiento puede recibir, o retornar, cero o muchos valores a través de la lista de parámetros.

La sintaxis de un procedimiento es la siguiente:

```
CREATE [OR REPLACE] PROCEDURE <nombre>
  [(<param1> [IN | OUT | IN OUT] <type>,
    (<param2> [IN | OUT | IN OUT] <type>, ...)]
IS
  -- Declaración de variables
BEGIN
  -- Instrucciones de las acciones
[EXCEPTION]
  -- Sentencias de control de excepciones
END;
```

Existen diversas formas para ejecutar un procedimiento. Dentro de una sección PL/SQL ejecutable, el procedimiento es llamado como una sentencia ejecutable independiente.

```
BEGIN

    procedureName(param1);

END
```

Aunque por lo general, se ejecuta de alguna de las siguientes formas:

```
EXEC procedureName(param1);

O bien

CALL procedureName(param1);
```

Funciones

Una función es un módulo que siempre retorna un valor por medio de su cláusula **RETURN**. Como una función retorna un valor, se suele decir que tiene un tipo de dato.

Normalmente son usadas para cálculos, mientras que los procedimientos son usados para ejecutar lógica de negocio. Además de eso, pueden ser utilizadas en instrucciones DML o asignación directa a variables, mientras que los procedimientos no.

La sintaxis para construir funciones es muy similar a la de los procedimientos, excepto que la función también tiene una cláusula **RETURN**. El formato es el siguiente:

```
CREATE [OR REPLACE] FUNCTION <nombre>
    [(<param1> [IN | OUT | IN OUT] <type>,
      (<param2> [IN | OUT | IN OUT] <type>, ...)]
RETURN <return_datatype>
IS
    -- Declaración de variables
BEGIN
    -- Instrucciones de las acciones
[EXCEPTION]
    -- Sentencias de control de excepciones
END;
```

Nota: Puedes encontrar funciones útiles de Oracle [aquí](#).

Excepciones

PL/SQL permite a los desarrolladores levantar y manejar errores (excepciones) en una forma muy flexible y poderosa. Cada bloque PL/SQL puede tener su propia sección para el manejo de excepciones, donde estás pueden ser “atrapadas” y resueltas. Cuando se levanta una excepción en un bloque PL/SQL, su ejecución termina de forma inmediata. Todas las excepciones tienen un número de error y un mensaje; algunas incluso tienen nombre, por ejemplo: TOO_MANY_ROWS, ZERO_DIVIDE, etc.

Por supuesto, también se puede crear excepciones propias, pero estas se deben levantar de forma explícita.

```
DECLARE
    <exception_name> EXCEPTION;
BEGIN
    IF {fail_condition} THEN
        RAISE <exception_name>;
    EXCEPTION
        WHEN <exception_name> THEN
            -- Acción a realizar cuando se detecte esta excepción
END;
```

Como se pudo observar en los apartados de bloques anónimos (guía 2), funciones y procedimientos almacenados, cada uno de ellos cuenta con su propia sección (opcional) para manejar excepciones, complementándose así entre ellas. Es siempre recomendable manejar las excepciones, y hacerlo de forma correcta; se debe recordar que los usuarios finales de la aplicación son personas que pueden tener poco, o mucho, conocimiento técnico sobre informática.

Nota: En el Moodle puedes encontrar una lista con los [códigos de error](#) de Oracle y un resumen de las [excepciones predefinidas](#).

¿Cuál es la principal diferencia entre procedimientos y funciones?

Si no se indica si el parámetro es IN, OUT o IN OUT, ¿Cuál es el tipo por defecto?