

# Introducción a PL/SQL

*Grupo de Ingeniería del Software y Bases de Datos  
Departamento de Lenguajes y Sistemas Informáticos*

*Universidad de Sevilla*

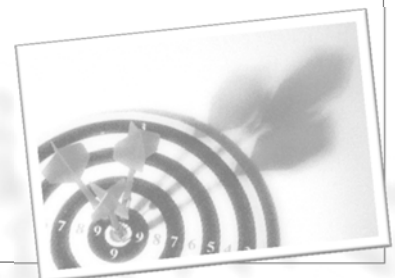
*noviembre 2011*



1. Introducción
2. Tipos
3. Operaciones
4. Control de Flujo
5. Bloques

## PL/SQL

- **Objetivos de este tema**
  - Conocer PL/SQL.
  - Conocer los tipos de datos.
  - Conocer las operaciones que permite.
  - Conocer los bloques PLSQL.



## PL/SQL

- SQL es un lenguaje de consulta para los sistemas de bases de datos relacionales, pero que no posee la potencia de los lenguajes de programación.
- PL/SQL amplía SQL con los elementos característicos de los lenguajes de programación, variables, sentencias de control de flujo, bucles, ...
- Cuando se desea realizar una aplicación completa para el manejo de una base de datos relacional, resulta necesario utilizar alguna herramienta que soporte la capacidad de consulta del SQL y la versatilidad de los lenguajes de programación tradicionales.
- PL/SQL es el lenguaje de programación que proporciona **Oracle** para extender el SQL estándar con otro tipo de instrucciones.

## PL/SQL

- Con PL/SQL vamos a poder programar las unidades de programa de la base de datos ORACLE, están son:
  - Procedimientos almacenados
  - Funciones
  - Triggers
  - Scripts

## Fundamentos de PL/SQL

- PL/SQL no es CASE-SENSITIVE, es decir, no diferencia mayúsculas de minúsculas aunque **ORACLE sea CASE-SENSITIVE en la búsquedas de texto.**
- Una línea en PL/SQL contiene grupos de caracteres conocidos como UNIDADES LÉXICAS, que pueden ser clasificadas como:
  - DELIMITADORES, IDENTIFICADORES, LITERALES, COMENTARIOS y EXPRESIONES
- DELIMITADOR: Es un símbolo simple o compuesto que tiene una función especial en PL/SQL. Estos pueden ser:
  - Operadores Aritméticos, Operadores Lógicos, Operadores Relacionales.
- IDENTIFICADOR: Son empleados para nombrar objetos de programas en PL/SQL así como a unidades dentro del mismo, estas unidades y objetos incluyen:
  - Constantes, Cursores, Variables, Subprogramas, Excepciones, Paquetes
- LITERAL: Es un valor de tipo numérico, carácter, cadena o lógico no representado por un identificador (es un valor explícito).
- COMENTARIO: Es una aclaración que el programador incluye en el código. Son soportados 2 estilos de comentarios, el de línea simple y de multilínea, para lo cual son empleados ciertos caracteres especiales como son:
  - Línea simple
  - /\* conjunto de líneas \*/

## Tipos de Datos

- Cada constante y variable tiene un tipo de dato en el cual se especifica el formato de almacenamiento, restricciones y rango de valores validos.
- PLSQL proporciona una variedad predefinida de tipos de datos. Casi todos los tipos de datos manejados por PL/SQL son similares a los soportados por SQL. Los más utilizados son:
  - **NUMBER** (*Numérico*).
  - **CHAR** (*Carácter*).
  - **VARCHAR2** (*Carácter de longitud variable*).
  - **BOOLEAN** (*lógico*).
  - **DATE** (*Fecha*). Las fechas se almacenan internamente como datos numéricos, por lo que es posible realizar operaciones aritméticas con ellas.
  - Atributos de tipo. Un atributo de tipo PL/SQL es un modificador que puede ser usado para obtener información de un objeto de la base de datos. El atributo **%TYPE** permite conocer el tipo de una variable, constante o campo de la base de datos. El atributo **%ROWTYPE** permite obtener los tipos de todos los campos de una tabla de la base de datos, de una vista o de un cursor.
  - PL/SQL también permite la creación de tipos personalizados (registros) y colecciones(tablas de PL/SQL).

## Operaciones

- **Operador de asignación:** ‘:=‘
- **Operadores aritméticos:** ‘+’ (suma), ‘-’ (resta), ‘\*’ (multiplicación), ‘/’ (división), ‘\*\*’ (exponente)
- **Operadores relacionales o de comparación:** ‘=‘ (igual a), ‘<>’ (distinto de), ‘<’ (menor que), ‘>’ (mayor que), ‘>=‘ (mayor o igual a), ‘<=‘ (menor o igual a)
- **Operadores lógicos:** AND (y lógico), NOT (negación), OR (o lógico)
- **Operador de concatenación** ‘||’

## Estructuras de Control de Flujo

- **Estructura condicional IF**

```
IF (expresión) THEN
    -- Instrucciones
ELSIF (expresión) THEN
    -- Instrucciones
ELSE
    -- Instrucciones
END IF;
```

- **Sentencias GOTO**

```
DECLARE
    flag NUMBER;
BEGIN
    flag :=1 ;
    IF (flag = 1) THEN
        GOTO paso2;
    END IF;
<<paso1>>
    dbms_output.put_line('Ejecución de paso 1');
<<paso2>>
    dbms_output.put_line('Ejecución de paso 2');
END;
```

## Estructuras de Control de Flujo

- LOOP

### LOOP

```
-- Instrucciones  
IF (expresión) THEN  
    -- Instrucciones  
EXIT;  
END IF;  
END LOOP;
```

- WHILE

```
WHILE (expresion) LOOP  
    -- Instrucciones  
END LOOP;
```

- FOR

```
FOR contador IN [REVERSE] inicio..final LOOP  
    -- Instrucciones  
END LOOP;
```

## Bloques PL/SQL

- Un programa de PL/SQL está compuesto por bloques. Un programa está compuesto como mínimo de un bloque

- Estructura de un Bloque

- La sección declarativa en donde se declaran todas las constantes y variables que se van a utilizar en la ejecución del bloque.
- La sección de ejecución que incluye las instrucciones a ejecutar en el bloque PL/SQL.
- La sección de excepciones en donde se definen los manejadores de errores que soportará el bloque PL/SQL.

```
[ declare | is | as ]  
/*Parte declarativa*/  
begin  
    /*Parte de ejecucion*/  
[ exception ]  
    /*Parte de excepciones*/  
end;
```



1. Introducción
2. Tipos
3. Operaciones
4. Control de Flujo
5. Bloques

## Ejemplo

### DECLARE

*/\*Parte declarativa\*/*

nombre\_variable **DATE**;

nombre\_variable [**CONSTANT**]

<tipo\_dato> [**NOT NULL**][:=valor\_inicial]

### BEGIN

*/\*Parte de ejecución /*

**SELECT** SYSDATE **INTO** nombre\_variable  
**FROM** DUAL;

### EXCEPTION

*/\*Parte de excepciones\*/*

**WHEN OTHERS THEN**

dbms\_output.put\_line('Se ha producido un error');

**END;**