

Práctica 4. Introducción a PL/SQL.

Variables, bloques y operadores aritméticos en PL/SQL

Profesor: José M. Luna

Email: jmluna@uco.es

Índice

- Introducción a PL/SQL
- Bloques
- Variables, operadores, etc.

Índice

- **Introducción a PL/SQL**
- Bloques
- Variables, operadores, etc.

Introducción a PL/SQL

- SQL es un lenguaje de consulta para los sistemas de bases de datos relacionales
 - **Inconveniente:** 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, etc)
- PL/SQL es el lenguaje de programación que proporciona Oracle para extender el SQL estándar con otro tipo de instrucciones.

Índice

- Introducción a PL/SQL
- **Bloques**
- Variables, operadores, etc.

Bloques

- Un programa de **PL/SQL** **está compuesto por bloques** (como mínimo debe contener 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.

Bloques

- Un programa de **PL/SQL** está **compuesto por bloques** (como mínimo debe contener un bloque)

- **Estructura de un bloque**

[declare]

/*Parte declarativa*/

begin

/*Parte de ejecución*/

[exception]

/*Parte de excepciones*/

end;

Índice

- Introducción a PL/SQL
- Bloques
- **Variables, operadores, etc.**

Variables, Operadores, etc.

- **Declaración de variables:**

- Dos variables pueden tener el mismo nombre, si están en bloques diferentes.
- El nombre de la variable (identificador) no debería ser el mismo que el de una columna de una tabla utilizada en el bloque.

identificador [CONSTANT] tipo_dato [NOT NULL] [:= | DEFAULT expresión];

- **Asignación de variables:**

- identificador := expresión;

- **Inicialización de variables:**

- identificador tipo_dato := expresión;
- identificador tipo_dato DEFAULT valor;
- identificador tipo_dato NOT NULL := valor;

Variables, Operadores, etc.

- **Operadores aritméticos:** +, -, *, /, **(potencia)
- **Operadores relacionales:** =, <>, <, >, <=, >=
- **Operadores lógicos:** AND, NOT, OR
- **Operadores de concatenación:** ||

Variables, Operadores, etc.

- **Ejemplo:**

```
DECLARE
    variable1 NUMBER := 1;
    variable2 NUMBER := 0;
BEGIN
    variable2 := variable1 +1;
END;
```

Variables, Operadores, etc.

- **Mostrar por pantalla:**

- **Necesario incluir el servidor de salida** set serveroutput on;
- **Mostrar un texto** DBMS_OUTPUT.PUT_LINE('Salida');
- **Mostrar el valor de una variable** DBMS_OUTPUT.PUT_LINE(variable2);

```
set serveroutput on;
```

```
DECLARE
```

```
    variable1 NUMBER := 0;
```

```
    variable2 NUMBER := 0;
```

```
BEGIN
```

```
    variable2 := variable1 +1;
```

```
    DBMS_OUTPUT.PUT_LINE('El valor es: ' || variable2);
```

```
END;
```

Variables, Operadores, etc.

- **Guardar una fila en una variable:**

```
DECLARE
    salida votantes%rowtype;
    mydni NUMBER := 8727847;
BEGIN
    SELECT * INTO salida FROM votantes where dni=mydni;
    dbms_output.put_line(salida.nombrecompleto);
END;
```

- **Guardar un valor en una variable:**

```
DECLARE
    v_nombre votantes.nombrecompleto%type;
    v_correo votantes.email%type;
    mydni NUMBER := 8727847;
BEGIN
    SELECT nombrecompleto, email INTO v_nombre, v_correo FROM votantes where dni=mydni;
    dbms_output.put_line(v_nombre || ' ' || v_correo);
END;
```