



2. Operadors i estructures de flux

RA5.Desenvolupa procediments emmagatzemats
avaluant i utilitzant les sentències del llenguatge
incorporat al sistema gestor de bases de dades.



1. Operadores de PL/SQL
2. Estructuras de control de flujo
3. Ejemplos

2.1. Operadores de PL/SQL

Tipos	Operadores	Descripción
Asignación	:=	Asigna un valor a una variable. edad:=20
Lógicos	AND, OR y NOT	Operadores lógicos Y, O y NO respectivamente.
Concatenación		Une dos o más cadenas. 'Hola' 'soy yo'
Comparación	=, !=, <=, <, >=, >, IS NULL, BETWEEN, LIKE, IN	Operadores de comparación. Funcionan igual que en SQL.
Aritméticos	+, -, *, /, **, ...	Se usan para realizar cálculos.

Prioridad de los operadores de PL/SQL

Prioridad	Operadores	Operación
1	**, NOT	Exponenciación, negación
2	*, /	Multiplicación, división.
3	+, -,	Suma, resta, concatenación
4	=, !=, <=, <, >=, >, IS NULL, BETWEEN, LIKE, IN	Comparación
5	AND	Conjunción
6	OR	Inclusión

2.2. Estructuras de control de flujo

Estructuras de control alternativa

Alternativa simple:

```
IF condición THEN  
    instrucciones;  
...;  
END IF;
```

```
IF edad=18 THEN  
    DBMS_OUTPUT.PUT_LINE('Es mayor de edad');  
END IF;
```

Estructuras de control alternativa

Alternativa doble:

```
IF condición THEN
    instrucciones1;
...; ELSE
    instrucciones2
;

...;
END IF;
```

```
SET SERVEROUTPUT ON
DECLARE
    estado VARCHAR2(1);
BEGIN
    IF edad >= 18 THEN
        DBMS_OUTPUT.PUT_LINE('Es mayor de edad');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Es menor de edad');
    END IF;
END;

/
```

Estructuras de control alternativa

Alternativa múltiples:

```
IF condición1 THEN
    instrucciones1;
...;
ELSIF condición2 THEN
    instrucciones2;
...;
ELSIF condición3 THEN
    instrucciones3;
...;
[ELSE
    Instrucciones4;
...;]
END IF;
```

```
DECLARE
    estado VARCHAR2(1);
BEGIN
    IF estado = 'C' THEN
        DBMS_OUTPUT.PUT_LINE('Casado');
    ELSIF estado = 'S' THEN
        DBMS_OUTPUT.PUT_LINE('Soltero');
    ELSIF estado = 'V' THEN
        DBMS_OUTPUT.PUT_LINE('Viudo');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Desconocido');
    END IF;
END;
```

/

Estructuras de control repetitiva

Mientras...

```
WHILE condición LOOP
    instrucciones;
...;
END LOOP;
```

```
DECLARE
    i NUMBER:=0;
BEGIN
    WHILE i>=5 LOOP
        DBMS_OUTPUT.PUT_LINE('HOLA');
        i:=i+1;
    END LOOP;
END;
/
```


Estructuras de control repetitiva

Iterar ... fin iterar salir si ...

```
LOOP
    instrucciones;
    ...;
    IF condición THEN
        EXIT;
    END IF;
    ...;
END LOOP;
```

```
DECLARE
    i NUMBER:=2;
BEGIN
    LOOP
        DBMS_OUTPUT.PUT_LINE('HOLA');
        IF i=5 THEN
            EXIT;
        END IF;
        i:=i+1;
    END LOOP;
END;
```

Estructuras de control repetitiva

Para...

```
FOR variable IN rango_mínimo..rango_máximo LOOP
    instrucciones;
...;
END LOOP;
```

```
FOR variable IN REVERSE rango_mínimo..rango_máximo LOOP
    instrucciones;
...;
END LOOP;
```

Estructuras de control repetitiva (ejemplo):

```
BEGIN
  FOR i IN 5..10 LOOP
    DBMS_OUTPUT.PUT_LINE ('El valor de i es '||i);
  END LOOP;
END;
```

```
El valor de i es 5
El valor de i es 6
El valor de i es 7
El valor de i es 8
El valor de i es 9
El valor de i es 10
```

```
BEGIN
  FOR i IN REVERSE 5..10 LOOP
    DBMS_OUTPUT.PUT_LINE ('El valor de i es '||i);
  END LOOP;
END;
```

```
El valor de i es 10
El valor de i es 9
El valor de i es 8
El valor de i es 7
El valor de i es 6
El valor de i es 5
```

2.3. Ejemplos

Ejemplo1:

Programar un script en PL/SQL que imprima tres veces la palabra HOLA y una vez ADIOS, para finalizar el procedimiento.

Usando la estructura

```
WHILE  
  
...  
  
END LOOP
```

```
DECLARE  
    i NUMBER;  
BEGIN  
    i:=1;  
    WHILE i!=5 LOOP  
        IF i!=4 THEN  
            DBMS_OUTPUT.PUT_LINE ('HOLA');  
        ELSE  
            DBMS_OUTPUT.PUT_LINE ('ADIOS');  
        END IF;  
        i:=i+1;  
    END LOOP;  
END;
```

Ejemplo2:

Programar un script en PL/SQL que imprima tres veces la palabra HOLA y una vez ADIOS, para finalizar el procedimiento.

Ahora se usará el LOOP ... END LOOP, utilizando IF ... THEN ... END IF y la cláusula EXIT.

```
bloque anónimo terminado
HOLA
HOLA
HOLA
ADIOS
```

```
DECLARE
  i NUMBER;
BEGIN
  I:=1;
  LOOP
    IF i!=4 THEN
      DBMS_OUTPUT.PUT_LINE('HOLA');
    ELSE
      DBMS_OUTPUT.PUT_LINE('ADIOS');
      EXIT;
    END IF;
    i:=i+1;
  END LOOP;
END;
```

Ejemplo3:

Programar un script en PL/SQL que imprima tres veces la palabra HOLA y una vez ADIOS, para finalizar el procedimiento.

Ahora se usará la estructura repetitiva

FOR ... END LOOP.

```
bloque anónimo terminado
HOLA
HOLA
HOLA
ADIOS
```

```
DECLARE
  i NUMBER;
BEGIN
  FOR i IN 1..3 LOOP
    DBMS_OUTPUT.PUT_LINE ('HOLA');
  END LOOP;
  DBMS_OUTPUT.PUT_LINE ('ADIOS');
END;
/
```

Ejemplo4:

Programar un script en PL/SQL que nos devuelva la tabla de multiplicar de un número que es el valor inicial que se asigna a la variable «num».

```
DECLARE
    inicio NUMBER := 0;
    final NUMBER :=10;
    resultado NUMBER :=0;
    num NUMBER := 0;
BEGIN
    FOR ..... LOOP
        .....
        /* DBMS_OUTPUT DEL RESULTAT MOSTRAR ELS
           DOS NUMEROS QUE ES MULTIPIQUEN I EL
           RESULTAT*/
        .....
    END LOOP;
END;
/
```

Preguntes!!!!

