

Programación PL/SQL: Bloques anónimos

1. Procedimiento al que se le pasarán dos números y visualice si uno es divisor de otro (se utilizará la función MOD (dividendo, divisor) que da el resto de dividir el primer número entre el segundo).

```
SET SERVEROUT ON;
```

```
DECLARE
```

```
  V_NUM1 NUMBER(5);
  V_NUM2 NUMBER(5);
  V_RESULTADO NUMBER(5);
```

```
BEGIN
```

```
  V_NUM1 := &Introduce_el_primer_número;
  V_NUM2 := &Introduce_el_segundo_número;
```

```
  V_RESULTADO := MOD(V_NUM1,V_NUM2);
```

```
  IF V_RESULTADO = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Si son divisibles');
```

```
  ELSE
    DBMS_OUTPUT.PUT_LINE('No son divisibles');
```

```
  END IF;
```

```
END;
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script for an anonymous block. The script includes comments, variable declarations, and logic to check if one number is divisible by another using the MOD function. Below the script, the 'Salida de Script' (Script Output) window shows the execution results, indicating that the numbers 4 and 2 are divisible, and the PL/SQL procedure terminated successfully.

```
-- 1. Procedimiento al que se le pasarán dos números y visualice si uno es divisor de otro
-- (se utilizará la función MOD(dividendo,divisor) que da el resto de dividir el primer número entre el segundo).

SET SERVEROUT ON;

DECLARE
  V_NUM1 NUMBER(5);
  V_NUM2 NUMBER(5);
  V_RESULTADO NUMBER(5);

BEGIN
  V_NUM1 := &Introduce_el_primer_número; --030 al pedir las cosas, que si pido un número, nada de frases con espacios en las palabras, una sola cadena (sin comillas)
  V_NUM2 := &Introduce_el_segundo_número;

  V_RESULTADO := MOD(V_NUM1,V_NUM2);

  IF V_RESULTADO = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Si son divisibles');
  ELSE
    DBMS_OUTPUT.PUT_LINE('No son divisibles');
  END IF;

END;
```

Salida de Script

```

BEGIN
  V_NUM1 := &Introduce_el_primer_número; --030 al pedir las cosas, que si pido un número, nada de frases con espacios en las palabras, una sola cadena (sin comillas)
  V_NUM2 := &Introduce_el_segundo_número;

  V_RESULTADO := MOD(V_NUM1,V_NUM2);

  IF V_RESULTADO = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Si son divisibles');
  ELSE
    DBMS_OUTPUT.PUT_LINE('No son divisibles');
  END IF;

END;
Nuevo: DECLARE
  V_NUM1 NUMBER(5);
  V_NUM2 NUMBER(5);
  V_RESULTADO NUMBER(5);

BEGIN
  V_NUM1 := 4; --030 al pedir las cosas, que si pido un número, nada de frases con espacios en las palabras, una sola cadena (sin comillas)
  V_NUM2 := 2;

  V_RESULTADO := MOD(V_NUM1,V_NUM2);

  IF V_RESULTADO = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Si son divisibles');
  ELSE
    DBMS_OUTPUT.PUT_LINE('No son divisibles');
  END IF;

END;
Si son divisibles

Procedimiento PL/SQL terminado correctamente.
```

Salida de DBMS | Tamaño de Buffer: 20000 | Mensajes - Log

2. Visualizar la tabla de multiplicar de un número entre 1 y 10 introducido por teclado.

DECLARE

V_NUM NUMBER; -- no hace falta especificar el tamaño

V_RESULTADO NUMBER; --mejor smp guardar el resultado en una variable resultado

--V_MULTIPLICADOR NUMBER(5);

BEGIN

V_NUM := &Introduzca_un_número;

--V_MULTIPLICADOR := V_NUM;

FOR i IN 1..10 LOOP

V_RESULTADO := V_NUM * i;

--V_MULTIPLICADOR ++;

DBMS_OUTPUT.PUT_LINE(V_RESULTADO);

END LOOP;

END;

The screenshot shows the Oracle SQL Developer interface. The top pane displays a PL/SQL script with the following content:

```
-- 2. Visualizar la tabla de multiplicar de un número entre 1 y 10 introducido por teclado.

SET SERVEROUT ON;

DECLARE
  V_NUM NUMBER; -- no hace falta especificar el tamaño
  V_RESULTADO NUMBER; --mejor smp guardar el resultado en una variable resultado
  --V_MULTIPLICADOR NUMBER(5);

BEGIN
  V_NUM := &Introduzca_un_número;
  --V_MULTIPLICADOR := V_NUM;

  FOR i IN 1..10 LOOP
    V_RESULTADO := V_NUM * i;
    --V_MULTIPLICADOR ++;

    DBMS_OUTPUT.PUT_LINE(V_RESULTADO);
  END LOOP;
END;
```

The bottom pane shows the output of the script execution. It indicates that the task was completed in 8.04 seconds. The output shows the multiplication table for the number 1, with the first line highlighted in red:

```
V_RESULTADO := V_NUM * i;
--V_MULTIPLICADOR ++;

DBMS_OUTPUT.PUT_LINE(V_RESULTADO);

END LOOP;

END;
Nuevo:DECLARE
  V_NUM NUMBER; -- no hace falta especificar el tamaño
  V_RESULTADO NUMBER; --mejor smp guardar el resultado en una variable resultado
  --V_MULTIPLICADOR NUMBER(5);

BEGIN
  V_NUM := 1;
  --V_MULTIPLICADOR := V_NUM;

  FOR i IN 1..10 LOOP
    V_RESULTADO := V_NUM * i;
    --V_MULTIPLICADOR ++;

    DBMS_OUTPUT.PUT_LINE(V_RESULTADO);
  END LOOP;

END;
1
2
3
4
5
6
7
8
9
10

Procedimiento PL/SQL terminado correctamente.
```

3. Obtener las tablas de multiplicar de los números del 1 al 10.

```

DECLARE
  V_RESULTADO NUMBER;
  i NUMBER;
  j NUMBER;

BEGIN
  FOR i IN 1..10 LOOP
    FOR j IN 1..10 LOOP
      V_RESULTADO := i * j;

      DBMS_OUTPUT.PUT_LINE(V_RESULTADO);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(' ');
  END LOOP;
END;

```

The screenshot shows the SQL Developer interface. The top tab is 'Ejer_2_Bloques_Anonimos_2.sql'. Below it, the 'Hoja de Trabajo de SQL' tab shows the script for Exercise 3. The script is as follows:

```

-- 3. Obtener las tablas de multiplicar de los números del 1 al 10.

SET SERVEROUTPUT ON;

DECLARE
  V_RESULTADO NUMBER;
  i NUMBER;
  j NUMBER;

BEGIN
  FOR i IN 1..10 LOOP
    FOR j IN 1..10 LOOP
      V_RESULTADO := i * j;

      DBMS_OUTPUT.PUT_LINE(V_RESULTADO);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(' ');
  END LOOP;
END;

```

The bottom tab, 'Salida de Script', shows the output of the script. It displays a list of numbers from 6 to 80, which are the results of the multiplication table (1*1 to 10*10).

4. Calcular la suma de los números del 1 al 100.

```

DECLARE
  V_SUMA NUMBER(5) := 0;

BEGIN
  FOR i in 1..100 LOOP
    V_SUMA := V_SUMA+i;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE(V_SUMA);

END;

```

The screenshot shows the SQL Developer interface. The top tab is 'Ejer_2_Bloques_Anonimos_2.sql'. Below it, the 'Hoja de Trabajo de SQL' tab shows the script for Exercise 4. The script is as follows:

```

-- 4. Calcular la suma de los números del 1 al 100.

SET SERVEROUTPUT ON;

DECLARE
  V_SUMA NUMBER(5) := 0;

BEGIN
  FOR i in 1..100 LOOP
    V_SUMA := V_SUMA+i;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE(V_SUMA);

END;

```

The bottom tab, 'Salida de Script', shows the output of the script. It displays the number 5050, which is the sum of numbers from 1 to 100.

5. Calcular el producto de los números del 1 al 100.

```

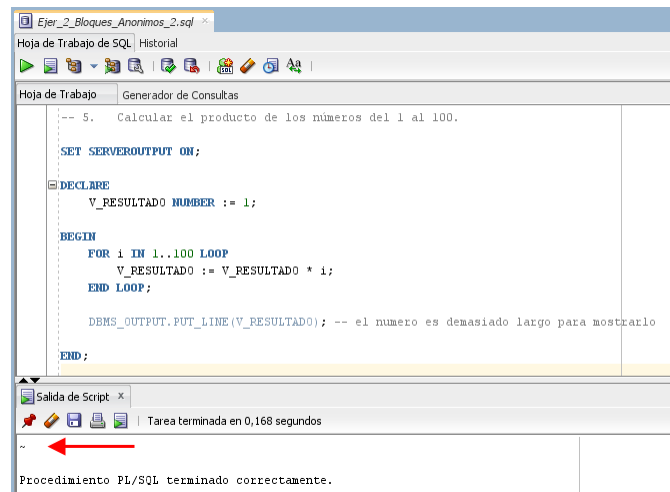
DECLARE
  V_RESULTADO NUMBER := 1;

BEGIN
  FOR i IN 1..100 LOOP
    V_RESULTADO := V_RESULTADO * i;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE(V_RESULTADO); -- el numero es demasiado largo para mostrarlo

END;

```

**6. Imprimir los múltiplos de 3 hasta N, siendo N un valor introducido por teclado.**

```

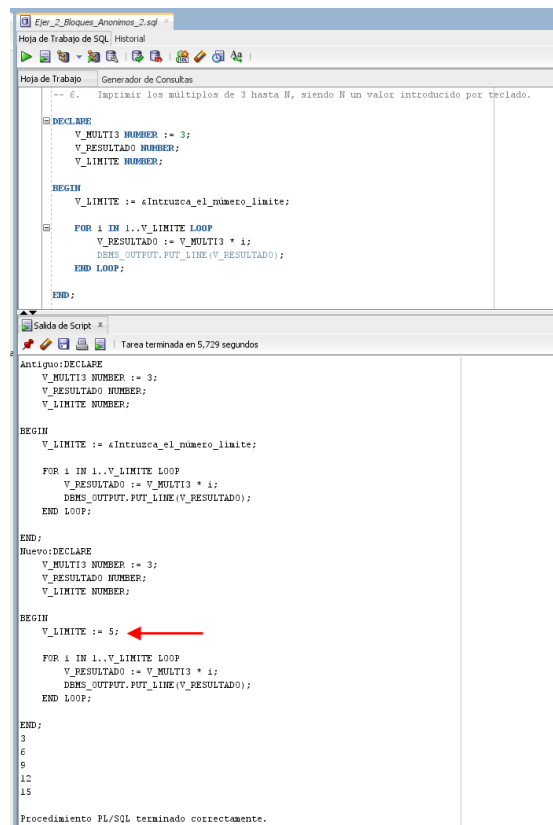
DECLARE
  V_MULT3 NUMBER := 3;
  V_RESULTADO NUMBER;
  V_LIMITE NUMBER;

BEGIN
  V_LIMITE := &Intruzca_el_número_límite;

  FOR i IN 1..V_LIMITE LOOP
    V_RESULTADO := V_MULT3 * i;
    DBMS_OUTPUT.PUT_LINE(V_RESULTADO);
  END LOOP;

END;

```



7. Visualizar el factorial de un número que se pide por teclado.

```

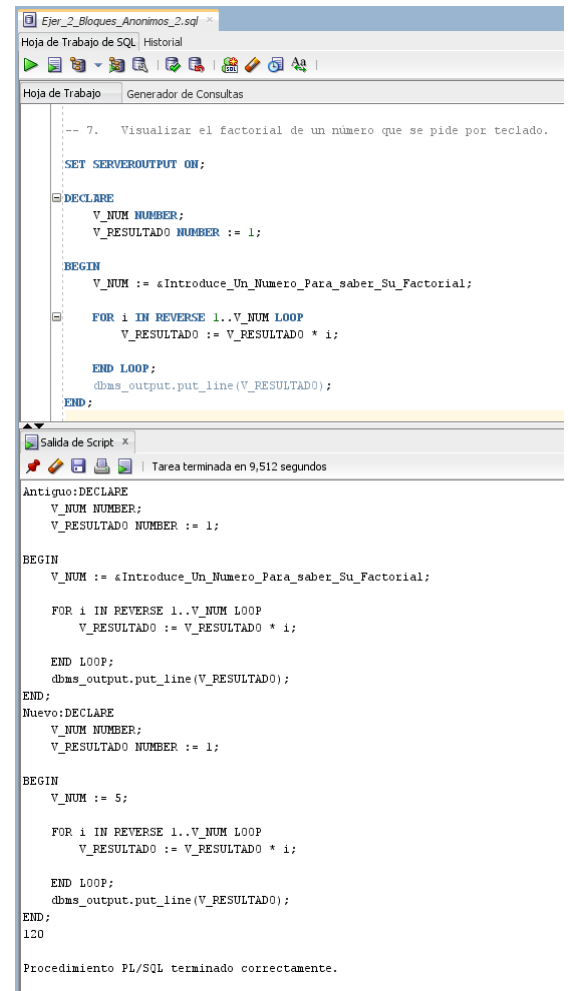
DECLARE
  V_NUM NUMBER;
  V_RESULTADO NUMBER := 1;

BEGIN
  V_NUM := &Introduce_Un_Numero_Para_saber_Su_Factorial;

  FOR i IN REVERSE 1..V_NUM LOOP
    V_RESULTADO := V_RESULTADO * i;

  END LOOP;
  dbms_output.put_line(V_RESULTADO);
END;

```

**8. Visualizar los factoriales de los números del 1 al 10.**

```

DECLARE
  V_NUM NUMBER := 1;
  V_RESULTADO NUMBER := 1;

BEGIN
  FOR i IN 1..10 LOOP
    FOR j IN REVERSE 1..V_NUM LOOP

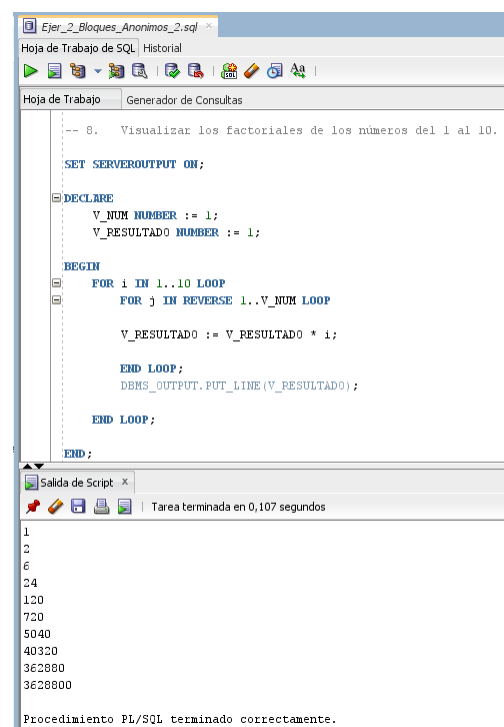
      V_RESULTADO := V_RESULTADO * j;

    END LOOP;
    DBMS_OUTPUT.PUT_LINE(V_RESULTADO);

  END LOOP;

END;

```



9. Visualizar la suma de todos los números existentes entre dos introducidos por teclado.

```

DECLARE
  V_NUM1 NUMBER;
  V_NUM2 NUMBER;
  V_RESULTADO NUMBER := 0;

BEGIN
  V_NUM1 := &Introduzca_el_1º_número;
  V_NUM2 := &Introduzca_el_2º_número;

  FOR i IN V_NUM1..V_NUM2 LOOP
    V_RESULTADO := V_RESULTADO + i;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE(V_RESULTADO);

END;

```

The screenshot shows the SQL Developer interface with a script titled 'Ejer_2_Bloques_Anonimos_2.sql'. The script contains the following PL/SQL code:

```

-- 9. Visualizar la suma de todos los números existentes entre dos introducidos por teclado.

SET SERVEROUTPUT ON;

DECLARE
  V_NUM1 NUMBER;
  V_NUM2 NUMBER;
  V_RESULTADO NUMBER := 0;

BEGIN
  V_NUM1 := &Introduzca_el_1º_número;
  V_NUM2 := &Introduzca_el_2º_número;

  FOR i IN V_NUM1..V_NUM2 LOOP
    V_RESULTADO := V_RESULTADO + i;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE(V_RESULTADO);

END;

```

The 'Salida de Script' (Script Output) window shows the execution results:

```

Antiguo:DECLARE
  V_NUM1 NUMBER;
  V_NUM2 NUMBER;
  V_RESULTADO NUMBER := 0;

BEGIN
  V_NUM1 := &Introduzca_el_1º_número;
  V_NUM2 := &Introduzca_el_2º_número;

  FOR i IN V_NUM1..V_NUM2 LOOP
    V_RESULTADO := V_RESULTADO + i;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE(V_RESULTADO);

END;
Nuevo:DECLARE
  V_NUM1 NUMBER;
  V_NUM2 NUMBER;
  V_RESULTADO NUMBER := 0;

BEGIN
  V_NUM1 := 1;
  V_NUM2 := 5;

  FOR i IN V_NUM1..V_NUM2 LOOP
    V_RESULTADO := V_RESULTADO + i;
  END LOOP;

  DBMS_OUTPUT.PUT_LINE(V_RESULTADO);

END;
15

Procedimiento PL/SQL terminado correctamente.

```

The output shows that the script was executed successfully, and the result '15' is displayed, which is the sum of numbers from 1 to 5.