
	<b>Bases de datos</b> <b>Control Tema 8</b>	
	<b>Procedimientos y funciones en PL/SQL</b>	

Especificar las correspondientes sentencia PL/SQL necesaria para realizar cada uno de los siguientes apartados (usar la fuente **Monaco**) y el mismo color que el del ejemplo.



Ejemplo:

```
SELECT * FROM Personas;
```

1. Diseña un bloque anónimo que solicite el valor de tres variables, las dos primeras variables corresponden a los operandos, y la tercera deber ser un número que significa lo siguiente: si introducen un 1 se deben sumar los operandos, si es un 2 se restan, si es un 3 se multiplican y si es otro número, mostrará un mensaje de operación no permitida. Se debe mostrar el resultado de la operación por pantalla. Si lo haces con la instrucción IF (1p) y con CASE (2p).

Código: (1punto / 2puntos)

```
SET SERVEROUTPUT ON
ACCEPT num1 NUMBER PROMPT 'Introduzca el primer numero';
ACCEPT num2 NUMBER PROMPT 'Introduzca el segundo numero';
ACCEPT num3 NUMBER PROMPT 'Introduzca el tercer numero';
DECLARE
    v_num1 NUMBER := &num1;
    v_num2 NUMBER := &num2;
    v_num3 NUMBER := &num3;
BEGIN
    CASE
        WHEN v_num3 = 1 THEN
            dbms_output.put_line(v_num1 + v_num2);
        WHEN v_num3 = 2 THEN
            dbms_output.put_line(v_num1 - v_num2);
        WHEN v_num3 = 3 THEN
            dbms_output.put_line(v_num1 * v_num2);
        ELSE
            dbms_output.put_line('operacion no permitida');
        END CASE;
END;
```

	<b>Bases de datos</b> <b>Control Tema 8</b>	
	<b>Procedimientos y funciones en PL/SQL</b>	



2. Crear un bloque anónimo que solicite 3 valores. El primero de ellos lo llamaremos **v\_divisible** corresponde al número divisible a comparar, y los otros dos **v\_Inicio** y **v\_Final**, que corresponden al valor inicial y final del rango a buscar. Mostrar por pantalla los números divisibles por **v\_divisible** entre **v\_Inicio** y **v\_Final**.

Código: ( 2.5 puntos)

```

SET SERVEROUTPUT ON
ACCEPT divisible NUMBER PROMPT 'Introduzca el primer numero';
ACCEPT numInicio NUMBER PROMPT 'Introduzca el segundo
numero';
ACCEPT numFinal NUMBER PROMPT 'Introduzca el tercer numero';
DECLARE
    v_divisible NUMBER := &divisible;
    v_inicio NUMBER := &numInicio;
    v_final NUMBER := &numFinal;
    v_numeros VARCHAR2(1000);
BEGIN
    FOR i IN v_inicio..v_final LOOP
        IF MOD(i,v_divisible) = 0 THEN
            v_numeros := v_numeros || ' ' || i;
        END IF;
    END LOOP;
    dbms_output.put_line('Los numeros divisibles por ' ||
v_divisible || ' son: ' ||v_numeros);
END;
/

```

	<b>Bases de datos</b> <b>Control Tema 8</b>	
	<b>Procedimientos y funciones en PL/SQL</b>	



3. Crear un procedimiento llamado **DividirNumero** al que pasaremos como parámetros de entrada: el **dividendo** y el **divisor** y devuelva como parámetros de salida: el **cociente** y el **resto**. La solicitud de los datos y mostrar los resultados se realizará en el bloque anónimo que realiza la llamada.

Código: ( 1 punto)

```
CREATE OR REPLACE PROCEDURE DividirNumero(dividendo IN
NUMBER, divisor IN NUMBER, cociente OUT NUMBER, resto OUT
NUMBER)
AS
BEGIN
    cociente := TRUNC(dividendo/divisor);
    resto := MOD(dividendo, divisor);
END;
/

-----

SET SERVEROUTPUT ON
ACCEPT dividendo NUMBER PROMPT 'Introduzca el dividendo';
ACCEPT divisor NUMBER PROMPT 'Introduzca el divisor';
DECLARE
    v_dividendo NUMBER := &dividendo;
    v_divisor NUMBER := &divisor;
    v_cociente NUMBER;
    v_resto NUMBER;
BEGIN
    dividirnumero(v_dividendo, v_divisor, v_cociente,
v_resto);
    dbms_output.put_line('Los cociente de la division es ' ||
v_cociente || ' y su resto es ' || v_resto);
END;
/
```

	<b>Bases de datos</b> <b>Control Tema 8</b>	
	<b>Procedimientos y funciones en PL/SQL</b>	

4. Crear un función llamada **Factorial** que devuelva el factorial del número pasado como parámetro. Se debe solicitar el número desde un bloque anónimo, además de mostrar el resultado. Por ejemplo, para calcular el factorial del número Factorial(4) --->  $4! = 4 \times 3 \times 2 \times 1 = 24$

Código: ( 2.5 puntos)



```

CREATE OR REPLACE FUNCTION Factorial(num1 IN NUMBER) RETURN
NUMBER
AS
    v_num NUMBER := num1;
BEGIN
    FOR i IN REVERSE 1..num1-1 LOOP
        v_num := v_num * i;
    END LOOP;
    RETURN v_num;
END;
/

-----

SET SERVEROUTPUT ON
DECLARE
    v_fac NUMBER;
BEGIN
    v_fac := Factorial(5);
    dbms_output.put_line(v_fac);
END;
/

```

	<b>Bases de datos Control Tema 8</b>	
	<b>Procedimientos y funciones en PL/SQL</b>	

5. Crear un procedimiento denominado **ContarLocalidades** que reciba como parámetro el id del país (country\_id) y devuelva en otro parámetro la cantidad de localidades del país. Mostrar el resultado en el bloque anónimo que realiza la llamada. *El pais xxxx tiene: xxxx localidades*

Comando: ( 2 puntos)

```
CREATE OR REPLACE PROCEDURE ContarLocalidades(pais_id IN
COUNTRIES.COUNTRY_ID%TYPE, cantidad OUT NUMBER)
AS
BEGIN
    SELECT COUNT(*)
    INTO cantidad
    FROM LOCATIONS
    WHERE country_id = pais_id;
END;
/
```

---

```
DECLARE
    v_cantidad NUMBER;
BEGIN
    contarlocalidades('CA',v_cantidad);
    dbms_output.put_line(v_cantidad);
END;
/
```