

Ejercicio-1

Escribir un procedimiento que escriba dos números y visualice su suma. Después escribir un bloque PL/SQL que introducidos dos números llame a dicho procedimiento para calcular el resultado.

Solución

```
CREATE OR REPLACE PROCEDURE sumarNumeros (nume1 NUMBER, nume2 NUMBER)
IS
    suma NUMBER(6);
BEGIN
    suma := nume1 + nume2;
    DBMS_OUTPUT.PUT_LINE ('Suma: '|| suma);
END sumarNumeros;
```

--Primero compilamos el procedimiento y luego ejecutamos el bloque.

```
BEGIN
    sumarNumeros (7,6);
END;
```

Ejercicio-2

Modificar el bloque del ejercicio anterior para que te pida por teclado los números a sumar.

Solución

```
ACCEPT numero1 PROMPT 'Introduce el primer número'
ACCEPT numero2 PROMPT 'Introduce el segundo número'
BEGIN
    sumarNumeros(&numero1,&numero2);
END;
```

Ejercicio-3

Escribir un procedimiento que permite calcular el doble y la mitad de un número

Solución

```
CREATE OR REPLACE PROCEDURE dobleMitad (num number) IS
    doble number(6);
    mitad number(6,2);
BEGIN
    doble:=num*2;
    mitad:= num/2;
    DBMS_OUTPUT.PUT_LINE ('Número: '|| num);
    DBMS_OUTPUT.PUT_LINE ('Doble: '||doble);
    DBMS_OUTPUT.PUT_LINE ('Mitad: '||mitad);
END dobleMitad;
```

--Bloque para probarlo

BEGIN

dobleMitad(8);

END;

Ejercicio-4

Codificar un procedimiento que reciba una cadena y la visualice al revés, A continuación realizar un bloque para comprobar que funciona.

Solución

CREATE OR REPLACE PROCEDURE cadenaReves (vcadena VARCHAR2)

IS

 vcadReves VARCHAR2(80);

BEGIN

 FOR i IN REVERSE 1..LENGTH (vcadena)

 LOOP vcadReves:= vcadReves || SUBSTR (vcadena,i,1);

 END LOOP;

 DBMS_OUTPUT.PUT_LINE (vcadReves);

END cadenaReves;

--Bloque para probarlo

BEGIN

 cadenaReves ('Hola mundo');

END;

Ejercicio-5

Modificar el bloque del ejercicio anterior para que te pida por teclado la cadena que queremos ver del revés.

Solución

ACCEPT cadena PROMPT 'Introduce la cadena':

BEGIN

 cadenaReves('&cadena');

END;

Ejercicio-6

Escribir un script que una vez introducido el código de departamento visualice utilizando un procedimiento el número de empleados de dicho departamento.

Solución

CREATE OR REPLACE PROCEDURE totalEmple (numedep NUMBER) IS vtotal NUMBER;

BEGIN

 SELECT COUNT (*) INTO vtotal FROM empleado WHERE depnume= numedep;

 DBMS_OUTPUT.PUT_LINE ('El número de empleados en el depto: ' ||numedep||' es '||vtotal||');

END totalEmple;

--Bloque para probarlo

```
ACCEPT vdept PROMPT 'Introduce el número de departamento a consultar: '
```

```
BEGIN
```

```
    totalEmple (&vdept);
```

```
END;
```

Ejercicio-7

Crear un procedimiento que permita visualizar todos los datos de un usuario a partir del número de empleado, visualizar además el nombre de departamento al que pertenece.

Solución

```
CREATE OR REPLACE PROCEDURE verempleado (vemp NUMBER) IS
```

```
    vreg empleado%rowtype;
```

```
    Vdnombre departamento.nombredesp%type;
```

```
BEGIN
```

```
    SELECT * INTO vreg FROM empleado WHERE numemp = vemp;
```

```
    SELECT nombredesp INTO vdnombre FROM departamento WHERE numedep= vreg.depnume;
```

```
    DBMS_OUTPUT.PUT_LINE('Número empleado :'|| vemp );
```

```
    DBMS_OUTPUT.PUT_LINE('Nombre : '|| vreg.apell );
```

```
    DBMS_OUTPUT.PUT_LINE('Oficio : '|| vreg.oficio );
```

```
    DBMS_OUTPUT.PUT_LINE('Su Jefe : '|| vreg.diremp);
```

```
    DBMS_OUTPUT.PUT_LINE('Fecha alta : '|| vreg.fechalt);
```

```
    DBMS_OUTPUT.PUT_LINE('Salario : '|| vreg.salario);
```

```
    DBMS_OUTPUT.PUT_LINE('Comisión : '|| vreg.comision);
```

```
    DBMS_OUTPUT.PUT_LINE('Departamento: '|| vreg.depnume ||' '||vdnombre);
```

```
END verempleado;
```

--Bloque para probarlo

```
ACCEPT vempo PROMPT 'Introduce el número del empleado a consultar: '
```

```
BEGIN
```

```
    verempleado(&vempo);
```

```
END;
```

Ejercicio-8

Crear un procedimiento que permita calcular la nómina de un empleado y visualizarla.

Realizar un bloque que una vez introducido por teclado el código de un empleado, visualice la nómina que cobrará el empleado utilizando el procedimiento anterior.

Solución



```
CREATE OR REPLACE PROCEDURE calculanomina (vnumemp empleado.numemp%TYPE)
```

```
IS
  Vnomina  NUMBER;
  Vreg      empleado%ROWTYPE;
BEGIN
  SELECT * INTO vreg FROM empleado WHERE numemp= vnumemp;
  Vnomina:= vreg.salario + NVL(vreg.comision,0);
  DBMS_OUTPUT.PUT_LINE ('La nomina del empleado'|| vnumemp ||' de nombre ||
                        vreg.apell || ' es '|| Vnomina);
END calculanomina;
```

```
--Bloque para probarlo
```

```
ACCEPT vnumemp PROMPT 'Introduce el número del empleado para calcular su nómina: '
```

```
BEGIN
  calculanomina (&vnumemp);
END;
```