UF3_PAC02

Para el desarrollo de esta PAC necesitarás la base de datos Empresa cuyo Script encontrareis junto a este documento en el campus virtual.

Ejercicios

1. Un procedimiento para mostrar el año actual.

```
create or replace procedure anioactual
as anio varchar2(10);
begin
  select extract(year from sysdate) into anio from dual;
  SYS.DBMS_OUTPUT.PUT_LINE(anio);
end;
```

2. Un procedimiento que sume uno a la variable anterior cada vez que se ejecute.

En este caso el parámetro del procedimiento es de entrada-salida ya que necesitamos su valor para incrementarlo y además necesitamos usarlo después para comprobarlo.

```
create or replace PROCEDURE sumauno(numero IN OUT number)
as
BEGIN
   numero:=numero+1;
END;
```

Para llamar al procedimiento:

```
CALL sumauno(variable);
```

3. Un procedimiento al que se le pasen dos cadenas como parámetros y las muestre concatenadas y en mayúsculas.

```
create or replace procedure concatena(cadena1 varchar2, cadena2
varchar2)
as
begin
  DBMS_OUTPUT.PUT_LINE(CONCAT(UPPER(cadena1), UPPER(cadena2)));

end;
```

Para llamar al procedimiento:

```
concatena('hola','mundo');
```

4. Bloque anónimo que pida un código de empleado y muestre su salario actual, después lo disminuya en un tercio y muestre el nuevo salario.

```
declare
  empleado EMP.EMP_NO%Type;
  salarioVar EMP.Salario%Type;
begin
  empleado:=&Numero_Empleado;

  select salario into salarioVar from emp where emp_no=empleado;
  DBMS_OUTPUT.PUT_LINE('El salario actual es: ' || salarioVar);

  salarioVar:=salarioVar-(salarioVar/3);

  Update emp set emp.salario=salarioVar where emp_no=empleado;

  select salario into salarioVar from emp where emp_no=empleado;
  DBMS_OUTPUT.PUT_LINE('El salario actual es: ' || salarioVar);

end;
```

5. Una función para mostrar el día de la semana según un valor de entrada numérico, 1 para domingo, 2 lunes, etc. utilizando la estructura condicional "IF/IF ELSE".

```
create or replace function diasemana (numero number)
return varchar2
is
  dia varchar2(20);
begin
  IF numero=1 Then dia:='Lunes';
  ELSIF numero=2 Then dia:='Martes';
  ELSIF numero=3 Then dia:='Miercoles';
  ELSIF numero=4 Then dia:='Jueves';
  ELSIF numero=5 Then dia:='Viernes';
  ELSIF numero=6 Then dia:='Sabado';
  ELSIF numero=7 Then dia:='Domingo';
  End if;

  return (dia);
End;
```

Ejemplo de uso:

```
DBMS_OUTPUT.PUT_LINE(diasemana(1));
```

6. La misma función que muestre el día de la semana según un valor de entrada numérico, 1 para domingo, 2 lunes, etc. Pero esta vez utilizando la estructura condicional “CASE”.

```
create or replace function diasemanaCASE (numero number)
return varchar2
is
  dia varchar2(20);
begin
  CASE numero
    WHEN 1 Then dia:='Lunes';
    WHEN 2 Then dia:='Martes';
    WHEN 3 Then dia:='Miercoles';
    WHEN 4 Then dia:='Jueves';
    WHEN 5 Then dia:='Viernes';
    WHEN 6 Then dia:='Sabado';
    WHEN 7 Then dia:='Domingo';
    Else dia:='Error';
  End Case;

  return (dia);
End;
```

Ejemplo de uso:

```
DBMS_OUTPUT.PUT_LINE(diasemanaCASE(1));
```

7. Una función que devuelva el mayor de tres números pasados como parámetros.

```
create or replace function mayorDeTres(numero1 number, numero2
number, numero3 number)
return number
is
  mayor number;
begin
IF(numero1>numero2) Then
  IF (numero1>numero3) Then
   mayor:=numero1;
  Else
    IF (numero3>numero2) Then
     mayor:=numero3;
    else
      mayor:=numero2;
    End if;
  End if;
  ELSIF (numero2>numero3) then
    mayor:=numero2;
    Elsif (numero3>numero1) then
      mayor:=numero3;

End if;
return (mayor);
end;
```

8. Un procedimiento que muestre la suma de los primeros n números enteros, siendo n un parámetro de entrada.

```
create or replace procedure sumaNumeros(numero number)
as
  acumulador number;
begin
  acumulador:=0;
  For iterador in 0 .. numero loop
    acumulador:=acumulador+iterador;
  end loop;

  DBMS_OUTPUT.PUT_LINE(acumulador);
end;
```

Para llamar al procedimiento:

```
sumaNumeros(4);
```

9. Una función que determine si un número es primo devolviendo 0 ó 1.

```
create or replace FUNCTION PRIMO(numero NUMBER)
RETURN number
is
  isPrimo number;
BEGIN
  isPrimo:=0;
  FOR I IN 2..numero-1 LOOP
    IF MOD(numero,I)=0 THEN
      Exit when isPrimo=0;
    ELSE isPrimo:=1;
    END IF;
  END LOOP;
RETURN(isPrimo);
END;
```

Ejemplo de uso:

```
DBMS_OUTPUT.PUT_LINE(PRIMO(1));
```

10.Usando la función anterior crear otra que calcule la suma de los primeros m números primos empezando en el 1.

```
create or replace function sumaPrimo (numero number)
return number
is
  acumulador number;
  contador number;
  iterador number;
begin
  contador:=0;
  acumulador:=0;
  iterador:=0;
  while contador<numero loop
    if(primo(iterador)=1) then
      acumulador:=acumulador+iterador;
      contador:=contador+1;
    end if;
    iterador:=iterador+1;
  end loop;

  return (acumulador);
end;
```

Ejemplo de uso:

```
DBMS_OUTPUT.PUT_Line(sumaprimo(2));
```

11. Crea una función "Nomina1" que reciba como argumento el código de un empleado y calcule el importe a cobrar cada mes tenemos en cuenta que:

- El salario que figura en la tabla "emp" es el sueldo bruto anual.
- Nuestros trabajadores cobran 14 pagas iguales al año.
- Cada mes les pagamos también la comisión correspondiente. La comisión que figura en la tabla es la comisión anual (correspondiente a 12 meses).

```
create or replace function nomina1(codigoEmp EMP.EMP_NO%type)
return float
is
  salarioMensual float;
  salarioBruto emp.salario%type;
  comisionAnual EMP.COMISION%type;

begin
  select salario, comision into salarioBruto, comisionAnual
from emp where emp_no=codigoEmp;
  salarioMensual:=(salarioBruto/14)+(comisionAnual/12);
  return (SalarioMensual);

end;
```

```
DBMS_OUTPUT.PUT_LINE(nomina1(7499));
```