**HOJA 1 DE EJERCICIOS TEMA 12: BLOQUES, FUNCIONES Y PROCEDIMIENTOS**

1. **Escribir un bloque PL/SQL que escriba el texto ‘Hola’**

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Hola');

END;

1. **Escribir un procedimiento que reciba dos números y visualice su suma.**

declare v\_entrada NUMBER(2);

v\_entrada2 NUMBER(2);

begin

v\_entrada:=&entrada\_dato;

v\_entrada2:=&entrada\_dato;

DBMS\_OUTPUT.PUT\_LINE(V\_entrada+v\_entrada2);

END;

1. **Codificar un procedimiento que reciba una cadena y la visualice al revés.**

declare

dato varchar2(24);

begin

dato :='&dato';

select reverse(dato) into dato from depart

where dept\_no=30;

DBMS\_OUTPUT.PUT\_LINE(dato);

end;

1. **Escribir una función que reciba una fecha y devuelva el año, en número, correspondiente a esa fecha.**

declare

fecha date;

begin

fecha :='&fecha';

DBMS\_OUTPUT.PUT\_LINE(extract(year from fecha));

end;

1. **Escribir un bloque PL/SQL que haga uso de la función anterior.**

declare

fecha date;

begin

fecha :='&fecha';

DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(fecha, 'DAY, DD MONTH " a las "YYYY:HH24:MI:SS'));

end;

1. **Dado el siguiente procedimiento:**

**CREATE OR REPLACE PROCEDURE crear\_depart (**

**p\_num\_dept depart.dept\_no%TYPE,**

**p\_dnombre depart.dnombre%TYPE DEFAULT 'PROVISIONAL',**

**p\_loc depart.loc%TYPE DEFAULT 'PROVISIONAL')**

**IS**

**BEGIN**

**INSERT INTO depart**

**VALUES (p\_num\_dept, p\_dnombre, p\_loc);**

**END crear\_depart;**

**Indicar cuáles de las siguientes llamadas son correctas y cuáles incorrectas, en este último caso escribir la llamada correcta usando la notación posicional (en los casos que se pueda):**

**1º. crear\_depart;**

**2º. crear\_depart(50);**

**3º. crear\_depart('COMPRAS');**

**4º. crear\_depart(50,'COMPRAS');**

**5º. crear\_depart('COMPRAS', 50);**

**6º. crear\_depart('COMPRAS', 'VALENCIA');**

**7º. crear\_depart(50, 'COMPRAS', 'VALENCIA');**

**8º. crear\_depart('COMPRAS', 50, 'VALENCIA');**

**9º. crear\_depart('VALENCIA', ‘COMPRAS’);**

**10º. crear\_depart('VALENCIA', 50);**

1. **Desarrollar una función que devuelva el número de años completos que hay entre dos fechas que se pasan como argumentos.**

CREATE OR REPLACE PROCEDURE entre\_fecha(fecha1 date, fecha2 date)

is

begin

DBMS\_OUTPUT.PUT\_LINE(floor(MONTHS\_BETWEEN(fecha1,fecha2)/12));

end;

begin

entre\_fecha('18-12-12','21-01-11');

end;

1. **Escribir una función que, haciendo uso de la función anterior devuelva los trienios que hay entre dos fechas. (Un trienio son tres años completos).**

CREATE OR REPLACE PROCEDURE entre\_fecha(fecha1 date, fecha2 date)

is

begin

DBMS\_OUTPUT.PUT\_LINE(floor(MONTHS\_BETWEEN(fecha1,fecha2)/12/3));

end;

begin entre\_fecha('18-12-15','21-01-11');

end;

1. **Codificar un procedimiento que reciba una lista de hasta 5 números y visualice su suma.**

CREATE OR REPLACE PROCEDURE suma(numero1 number,numero2 number,

numero3 number,numero4 number,numero5 number)

is

begin

DBMS\_OUTPUT.PUT\_LINE(numero1+numero2+numero3+numero4+numero5);

end;

begin suma(1,2,3,4,5);

end;

1. **Escribir una función que devuelva solamente caracteres alfabéticos sustituyendo cualquier otro carácter por blancos a partir de una cadena que se pasará en la llamada.**
2. **Implementar un procedimiento que reciba un importe y visualice el desglose del cambio en unidades monetarias de 1 cent., 2 cents., 5 cents., 10 cents., 20 cents., 50 cents., 1€, 2€, 5€, 10€, 20€, 50€ en orden inverso al que aparecen aquí enumeradas.**
3. **Codificar un procedimiento que permita borrar un empleado cuyo número se pasará en la llamada.**
4. **Escribir un procedimiento que modifique la localidad de un departamento. El procedimiento recibirá como parámetros el número del departamento y la localidad nueva.**
5. **Visualizar todos los procedimientos y funciones del usuario almacenados en la base de datos y su situación (valid o invalid).**