

Actividades T9

Escribe un procedimiento que reciba dos números y visualice su suma.

```
CREATE OR REPLACE PROCEDURE sumar_numeros (  
  num1 NUMBER,  
  num2 NUMBER)  
IS  
  suma NUMBER(6);  
BEGIN
```

Escribe un procedimiento que reciba dos números y visualice su suma.

```
CREATE OR REPLACE PROCEDURE sumar_numeros (  
  num1 NUMBER,  
  num2 NUMB  
  suma := num1 + num2;  
  DBMS_OUTPUT.PUT_LINE('Suma: '|| suma);  
END sumar_numeros;
```

Codifica un procedimiento que reciba una cadena y la visualice al revés.

```

CREATE OR REPLACE PROCEDURE cadena_reves(
vcadena VARCHAR2)
AS
vcad_reves VARCHAR2(80);
BEGIN
FOR i IN REVERSE 1..LENGTH(vcadena) LOOP
vcad_reves := vcad_reves || SUBSTR(vcadena,i,1);
END LOOP;
DBMS_OUTPUT.PUT_LINE(vcad_reves);
END cadena_reves;

```

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

```

CREATE OR REPLACE FUNCTION anio (
fecha DATE)
RETURN NUMBER
AS
v_anio NUMBER(4);
BEGIN
v_anio := TO_NUMBER(TO_CHAR(fecha, 'YYYY'));
RETURN v_anio;
END anio;

```

Desarrolla una función que devuelva el número de años completos que hay entre dos fechas que se pasan como parámetros.

```
CREATE OR REPLACE FUNCTION anios_dif (  
  fecha1 DATE,  
  fecha2 DATE)  
RETURN NUMBER  
  
AS  
  
  v_anios_dif NUMBER(6);  
  
BEGIN  
  
  v_anios_dif := ABS(TRUNC(MONTHS_BETWEEN(fecha2,fecha1)  
    / 12));  
  
  RETURN v_anios_dif;  
  
END anios_dif;
```

Escribe una función que, haciendo uso de la función anterior, devuelva los trienios que hay entre dos fechas (un trienio son tres años).

```
CREATE OR REPLACE FUNCTION trienios (  
  fecha1 DATE,  
  fecha2 DATE)  
  
RETURN NUMBER
```

AS

v_trienios NUMBER(6);

BEGIN

v_trienios := TRUNC(anios_dif(fecha1,fecha2) / 3);

RETURN v_trienios;

END;

Codifica un procedimiento que reciba una lista de hasta cinco números y visualice su suma.

CREATE OR REPLACE PROCEDURE sumar_5numeros (

Num1 NUMBER DEFAULT 0,

Num2 NUMBER DEFAULT 0,

Num3 NUMBER DEFAULT 0,

Num4 NUMBER DEFAULT 0,

Num5 NUMBER DEFAULT 0)

AS

BEGIN

DBMS_OUTPUT.PUT_LINE(Num1 + Num2 + Num3 + Num4 + Num5);

END sumar_5numeros;

Escribe una función que devuelva solamente caracteres alfabéticos sustituyendo cualquier otro carácter por blancos a partir de una cadena que se pasará en la llamada.

```

CREATE OR REPLACE FUNCTION sust_por_blanco(
cad VARCHAR2)
RETURN VARCHAR2
AS
nueva_cad VARCHAR2(30);
car CHARACTER;
BEGIN
FOR i IN 1..LENGTH(cad) LOOP
car:=SUBSTR(cad,i,1);
IF (ASCII(car) NOT BETWEEN 65 AND 90)
AND (ASCII(car) NOT BETWEEN 97 AND 122) THEN
car :=' ';
END IF;
nueva_cad := nueva_cad || car;
END LOOP;
RETURN nueva_cad;
END sust_por_blanco;

```

Codifica un procedimiento que permita borrar un empleado cuyo número se pasará en la llamada.

```

CREATE OR REPLACE PROCEDURE borrar_emple(
num_emple emple.emp_no%TYPE)
AS

```

BEGIN

DELETE FROM emple WHERE emp_no = num_emple;

END borrar_emple;

Escribe un procedimiento que modifique la localidad de un departamento. El procedimiento recibirá como parámetros el número del departamento y la nueva localidad.

CREATE OR REPLACE

PROCEDURE modificar_localidad(

num_depart NUMBER,

localidad VARCHAR2)

AS

BEGIN

UPDATE depart SET loc = localidad

WHERE dept_no = num_depart;

END modificar_localidad;

Visualiza todos los procedimientos y funciones del usuario almacenados en la base de datos y su situación (valid o invalid).

SELECT OBJECT_NAME, OBJECT_TYPE, STATUS

FROM USER_OBJECTS

WHERE OBJECT_TYPE IN ('PROCEDURE','FUNCTION');