

Daniel Gutiérrez – Ejercicios PL/SQL 13, 14 y 15

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

-----BLOQUE ANÓNIMO-----

```
SET SERVEROUTPUT ON;
DECLARE
    V_SUMA NUMBER(6);
BEGIN
    Ej13(1,5,76,4,3, V_SUMA);
    DBMS_OUTPUT.PUT_LINE(V_SUMA);
END;
```

-----PROCEDIMIENTO EJ13-----

```
CREATE OR REPLACE PROCEDURE EJ13
(
    P_NUM1 IN NUMBER
  , P_NUM2 IN NUMBER
  , P_NUM3 IN NUMBER
  , P_NUM4 IN NUMBER
  , P_NUM5 IN NUMBER
  , SUMA OUT NUMBER
) AS
BEGIN
    SUMA:=P_NUM1+P_NUM2+P_NUM3+P_NUM4+P_NUM5;
END EJ13;
```

-----RESULTADO DEL CÓDIGO-----

89

Daniel Gutiérrez – Ejercicios PL/SQL 13, 14 y 15

14. 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.

-----BLOQUE ANÓNIMO-----

```
SET SERVEROUTPUT ON;
BEGIN
  DBMS_OUTPUT.PUT_LINE(Ej14('¡Estoy(cambiando-por@espacios5todos?aquellos_caracteres@que
%no9sean<letras!'));
END;
```

-----FUNCIÓN EJ14-----

```
CREATE OR REPLACE FUNCTION EJ14
(
  P_CADENA IN VARCHAR2
) RETURN VARCHAR2 AS
BEGIN
  RETURN TRANSLATE(P_CADENA,'1234567890!@#$%^&*()=?\|@#~€¬^*Ç;:_[]{}`'+',.-<>|','
');
--El segundo apartado del translate tiene tantos espacios en blanco como caracteres a cambiar
END EJ14;
```

-----FUNCIÓN EJ14 ALTERNATIVA-----

```
create or replace FUNCTION EJ14
(
  P_CADENA IN VARCHAR2
)
RETURN VARCHAR2 AS
  P_AUX VARCHAR2(100);
BEGIN
  FOR contador IN 1..length(P_CADENA) LOOP
    IF ASCII(SUBSTR(P_CADENA,contador,1)) BETWEEN 65 AND 90 OR
    ASCII(SUBSTR(P_CADENA,contador,1)) BETWEEN 97 AND 127 THEN
      P_AUX:=P_AUX||SUBSTR(P_CADENA,contador,1);
      --Si el ascii está entre 65 y 90 (mayúsculas) o 97 y 127 (minúsculas), escribir el caracter de esa
      posición
    ELSE
      P_AUX:=P_AUX||' ';
      --Si no es una letra mayúscula o minúscula, escribir un espacio
    END IF;
  END LOOP;
  RETURN P_AUX;
END;
```

-----RESULTADO DEL CÓDIGO-----

Estoy cambiando por espacios todos aquellos caracteres que no sean letras

Daniel Gutiérrez – Ejercicios PL/SQL 13, 14 y 15

15. Implementar un procedimiento que reciba un importe y visualice el desglose del cambio en unidades monetarias de 1, 5, 10, 25, 50, 100, 200, 500, 1000, 2000, 5000 Ptas. en orden inverso al que aparecen aquí enumeradas.

-----BLOQUE ANÓNIMO-----

```
SET SERVEROUTPUT ON;
BEGIN
    EJ15(18320);
END;
```

-----PROCEDIMIENTO EJ15-----

```
CREATE OR REPLACE PROCEDURE EJ15
(
    P_IMPORTE_INTRODUCIDO IN NUMBER
) AS
    P_IMPORTE NUMBER(5);
    P_NUMERO_ELEMENTOS NUMBER(5);
    P_RESTO NUMBER(5);
    P_UNIDADES_MONETARIAS NUMBER(5);
BEGIN
    P_RESTO:=P_IMPORTE_INTRODUCIDO;

    FOR loop_counter IN 1..11
    LOOP
        P_IMPORTE:=P_RESTO;
        IF loop_counter=1 THEN P_UNIDADES_MONETARIAS:=5000;
        ELSIF loop_counter=2 THEN P_UNIDADES_MONETARIAS:=2000;
        ELSIF loop_counter=3 THEN P_UNIDADES_MONETARIAS:=1000;
        ELSIF loop_counter=4 THEN P_UNIDADES_MONETARIAS:=500;
        ELSIF loop_counter=5 THEN P_UNIDADES_MONETARIAS:=200;
        ELSIF loop_counter=6 THEN P_UNIDADES_MONETARIAS:=100;
        ELSIF loop_counter=7 THEN P_UNIDADES_MONETARIAS:=50;
        ELSIF loop_counter=8 THEN P_UNIDADES_MONETARIAS:=25;
        ELSIF loop_counter=9 THEN P_UNIDADES_MONETARIAS:=10;
        ELSIF loop_counter=10 THEN P_UNIDADES_MONETARIAS:=5;
        ELSIF loop_counter=11 THEN P_UNIDADES_MONETARIAS:=1;
        END IF;

        IF P_IMPORTE>=P_UNIDADES_MONETARIAS THEN
            P_NUMERO_ELEMENTOS:=FLOOR(P_IMPORTE/P_UNIDADES_MONETARIAS);
            P_RESTO:=P_IMPORTE-(P_UNIDADES_MONETARIAS*P_NUMERO_ELEMENTOS);
            DBMS_OUTPUT.PUT_LINE(P_NUMERO_ELEMENTOS||' de '||P_UNIDADES_MONETARIAS);
        ELSE
            P_NUMERO_ELEMENTOS:=0;
        END IF;
    END LOOP;
END EJ15;
```

-----RESULTADO DEL CÓDIGO-----

```
3 de 5000
1 de 2000
1 de 1000
1 de 200
1 de 100
2 de 10
```