

DEPARTAMENTO DE INFORMÁTICA

Ciclo Formativo: DAW

Curso: 2º

Módulo: BD (Bases de Datos)

Curso escolar: 2021-2022 Ejercicio: UT 6 PL/SQL

Evaluación: 2

Fecha aprobación: 28/03/2022

Fecha realización ejercicio: 01/04/2022

Práctica elaborada y aprobado por: A. Cristina Escudero

EJERCICIOS 1 PL/SQL

1. Diseñar un bloque PL/SQL que cuente el número de filas que hay en la tabla articulos, guarde el resultado en la variable v_num, y visualiza su contenido.

```
-- Ejercicio 1
declare
  v_num number(3):=0;
begin
  select count(*) into v_num
  from articulo;

  dbms_output.put_line('El numero de articulos es: ' || v_num);
end;
```

Salida de Script x

Tarea terminada en 0,071 segundos

El numero de articulos es: 10

Procedimiento PL/SQL terminado correctamente.

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

```
/
-- Ejercicio 2
create or replace procedure suma (num1 number, num2 number) is
suma number(5);
begin
  suma := num1 + num2;
  dbms_output.put_line('El resultado de la suma es: ' || suma);
end;
/
-- Ejecucion
exec suma(7,8);
```

Salida de Script x

Tarea terminada en 0,064 segundos

La suma de los siguientes numeros es: 15

Procedimiento PL/SQL terminado correctamente.



3. Codificar un procedimiento que reciba una cadena y la visualice al revés.
[Nota: se pueden usar las funciones de Oracle *LENGTH(cadena)* que calcula el tamaño de la cadena y *SUBSTR(cadena,pos,tamaño)* que obtiene una su cadena, desde la posición dada con el tamaño dado.]

```
-- Ejercicio 3
create or replace procedure reverseString (cadena varchar2) is
reverseCadena varchar2(32767);
longitud INT;
begin
    longitud := length(cadena);

    -- Bucle de for inverso
    for i in reverse 1..longitud loop
        reverseCadena := reverseCadena || substr(cadena,i,1);
    end loop;

    dbms_output.put_line(reverseCadena);
end;
/

-- Prueba del procedimiento
exec reverseString('Buenos dias');
```

Salida de Script x

Tarea terminada en 0,071 segundos

said soneuB

Procedimiento PL/SQL terminado correctamente.

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

```
-- Ejercicio 4
declare
    cadena varchar2(500) := 'Buenos dias';
    reverseCadena varchar2(1000);
    limite int;
begin
    limite := length(cadena);

    for i in reverse 1..limite loop
        reverseCadena := reverseCadena || substr(cadena,i,1);
    end loop;

    dbms_output.put_line(reverseCadena);
end;
```

Salida de Script x

Tarea terminada en 0,05 segundos

said soneuB

Procedimiento PL/SQL terminado correctamente.



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

[Nota: se pueden usar las funciones de Oracle *MONTHS_BETWEEN(fech1,fech2)*, que calcula el número de meses entre dos fechas; *TRUNC(fecha o numero)* que trunca el valor; *ABS(valor)* se queda con el valor absoluto.]

```

end;
/
-- Ejercicio 5
create or replace function cantidadMeses (fech1 date, fech2 date)
return int is
total int;
begin
    total := trunc(months_between(fech1,fech2));
    return total;
end cantidadMeses;
/

-- Prueba de la funcion
declare
total number;
fechal date := to_date('15-12-2022','DD-MM-YYYY');
fecha2 date := to_date('15-05-2023','DD-MM-YYYY');
begin
    total := cantidadMeses(fecha2,fechal);

    dbms_output.put_line(total);
end;
/

```

Salida de Script x

Tarea terminada en 0,042 segundos

5

Procedimiento PL/SQL terminado correctamente.

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

```

-- Ejercicio 6
create or replace function trienios (fech1 date, fech2 date)
return int is
meses int;
total int;
begin
    meses := cantidadMeses(fech1,fech2);
    total := trunc(meses/36);
    return total;
end;
/

--Prueba de la función
declare
total number;
fechal date := to_date('15-12-2020','DD-MM-YYYY');
fecha2 date := to_date('15-05-2023','DD-MM-YYYY');
begin
    total := trienios(fecha2,fechal);

    dbms_output.put_line(total);
end;
/

```

Salida de Script x

Tarea terminada en 0,041 segundos

0

Procedimiento PL/SQL terminado correctamente.



**Junta de
Castilla y León**

Consejería de Educación



UNION EUROPEA
Fondo Social Europeo
El FSE invierte en tu futuro.



Centro Integrado de Formación Profesional

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

```
-- Ejercicio 7
create or replace procedure suma (num1 int default 0, num2 int default 0, num3 int default 0, num4 int default 0, num5 int default 0)
total number;
begin
    total := num1 + num2 + num3 + num4 + num5;

    dbms_output.put_line('La suma de los siguientes numeros es: ' || total);
end;
/

-- Probamos el codigo
exec suma(1,2,3,4);

-- Ejercicio 8
```

Salida de Script x

Tarea terminada en 0,054 segundos

La suma de los siguientes numeros es: 10

Procedimiento PL/SQL terminado correctamente.

8. 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.

```
-- Ejercicio 8
create or replace function letras (cadena varchar2)
return varchar2 is
resultado varchar2(3000);
begin
    for i in 1..length(cadena) loop
        if substr(cadena,i,1) between 'A' and 'Z' or substr(cadena,i,1) between 'a' and 'z' then
            resultado := resultado || substr(cadena,i,1);
        else
            resultado := resultado || ' ';
        end if;
    end loop;

    return resultado;
end;
/

-- Prueba de la funcion
declare
frase varchar2(3000) := 'Buenos dias a las 11 de la mañana';
resultado varchar2(3000);
begin
    resultado := letras(frase);

    dbms_output.put_line(resultado);
end;
/
```

Salida de Script x

Tarea terminada en 0,066 segundos

Buenos dias a las de la ma ana

Procedimiento PL/SQL terminado correctamente.