Diego Viñals Lage

Ing. Informatica UFV

Bases de Datos

APUNTES PSQL

Índice

[1. APUNTES 2](#_Toc69290805)

[1.1 PROCEDIMIENTOS 2](#_Toc69290806)

[1.2 MOSTRAR COSAS POR PANTALLA 2](#_Toc69290807)

[1.3 ESTRUCTURAS DE CONTROL 2](#_Toc69290808)

[1. ALTERNATIVA SIMPLE 2](#_Toc69290809)

[2. ALTERNATIVA DOBLE 2](#_Toc69290810)

[3. ALTERNATIVA MÚLTIPLE 2](#_Toc69290811)

[4. BUCLE 3](#_Toc69290812)

[5. MIENTRAS 3](#_Toc69290813)

[6. PARA 3](#_Toc69290814)

[1.4 FUNCIONES 3](#_Toc69290815)

[2. EJERCICIOS 4](#_Toc69290816)

[2.1 EJERCICIO 1 4](#_Toc69290817)

[2.2 EJERCICIO 2 4](#_Toc69290818)

[2.3 EJERCICIO 3 5](#_Toc69290819)

[2.4 EJERCICIO 4 5](#_Toc69290820)

[2.5 EJERCICIO 5 6](#_Toc69290821)

# APUNTES

## PROCEDIMIENTOS

CREATE OR REPLACE PROCEDURE nombreproc()

AS

//declaración variables

BEGIN

//procesos

EXCEPTION

// Excepciones

END nombreproc;

## MOSTRAR COSAS POR PANTALLA

DBMS\_OUTPUT.PUT\_LINE(‘el … es:’ ||variable);

## ESTRUCTURAS DE CONTROL

### ALTERNATIVA SIMPLE

IF<CONDICIÓN>THEN

INSTRUCCIONES;

…

END IF;

### ALTERNATIVA DOBLE

IF<CONDICIÓN>THEN

INSTRUCCIONES;

…

ELSE

INSTRUCCIONES;

…

END IF;

### ALTERNATIVA MÚLTIPLE

IF<CONDICIÓN>THEN

INSTRUCCIONES;

…

ELSIF<CONDICIÓN>THEN

INSTRUCCIONES;

…

ELSIF<CONDICIÓN>THEN

INSTRUCCIONES;

…

END IF;

### BUCLE

LOOP

INSTRUCCIONES;

IF<CONDICIÓN>THEN

EXIT;

ENDIF;

INSTRUCCIONES;

…;

END LOOP;

### MIENTRAS

WHILE<CONDICIÓN>LOOP

INSTRUCCIONES;

…;

END LOOP;

### PARA

FOR<VARIABLE>IN<VALORINICIO>..<VALORFINAL>LOOP

INSTRUCCIONES;

…;

END LOOP;

## FUNCIONES

Las funciones no se ejecutan en el terminal, estas funciones se usan en procedimientos o en otras funciones

CREATE OR REPLACE FUNCTION nombre() RETURN tipo

AS

BEGIN

RETURN tipo

END nombre;

# EJERCICIOS

## EJERCICIO 1

* Procedimiento tal que recibe un numero de empleado y un salario para:
* 1. Verificar que el salario nuevo es mayor que el actual
* 2. En caso afirmativo, se procede a actualizar el salario del empleado

CREATE OR REPLACE PROCEDURE e01 (VNUMEM NUMBER, VSALAR NUMBER)

IS

VANTSALAR TEMPLE.SALAR%TYPE

BEGIN

VANTSALAR := 0;

SELECT SALAR INTO VANTSALAR FROM TEMPLE WHERE NUMEM = VNUMEM;

IF (VSALAR > VANTSALAR) THEN

UPDATE TEMPLE SET SALAR = VSALAR WHERE NUMEM = VNUMEM;

DBMS\_OUTPUT.PUT\_LINE(‘Se procede a actualizar el salario del empleado con el salario del empleado id: ‘ || VNUMEM || ’ , a salario, ‘ || VSLAR||);

ELSE

DBMS\_OUTPUT.PUT\_LINE(‘No se actualiza nada’);

END IF

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE(‘No se han encontrado datos’);

END e01;

EXECUTE e01(110, 300); //no actualizaria nada

EXECUTE e01(110, 3000);// Se actualiza

Commit; // para actualizar la bbdd

## EJERCICIO 2

* Generar un procedimiento tal que se le faciliten los atributos (valores) de la tabla TCENTR y cree una nueva tupla (fila) en nuestra tabla
* Necesitamos un nuevo NUMCE, NOMCE, SEÑAS

CREATE OR REPLACE PROCEDURE E02 (VNUMCE NUMBER,VNOMCE VARCHAR2 ,VSEGNAS VARCHAR2)

IS

BEGIN

INSERT INTO TCENTR(NUMCE, NOMCE, SEÑAS) VALUES(VNUMCE, VNOMCE, VSEGNAS);

DBMS\_OUTPUT.PUT\_LINE('Se procede a incorporar un nuevo centro en la tabla');

END E02;

## EJERCICIO 3

* Realizar, a modo de sistema experto, un procedimiento que nos muestre las opciones dentro del LMD (select, insert , update, delete) con la munestra de texto de un ejemplo para cada caso

CREATE OR REPLACE PROCEDURE e03()

IS

BEGIN

EXCEPTION

END e03;

## EJERCICIO 4

* Realizar una función que nos devuelva el resultado del departamento que tiene el mayor numero de empleados trabajando.
* El bloque anónimo imprime todos los empleados de ese departamento.

CREATE OR REPLACE FUNCTION P02 RETURN NUMBER

AS

VMAXIMO NUMBER := 0;

VCUENTA NUMBER :=0;

FUNCIÓN

BEGIN

SELECT NUMDE, COUNT (\*) INTO VMAXIMO, VCUENTA FROM TEMPLE GROUP BY NUMDE HAVING COUNT (\*)

IN (SELECT MAX(COUNT(NUMEM)) FROM TEMPLE GROUP BY NUMDE)

RETURN VMAXIMO

END P02

Declare

Depart number:= 0;

Cursor mi\_cursor is

BLOQUE ANÓNIMO

Select NOMEM from TEMPLE where NUMDE = P02

Begin

For vreg in mi\_cursor loop

DBMS\_OUTPUT.PUT\_LINE(vreg.nomen);

End loop;

End;

/

## EJERCICIO 5