

Universidad Nacional Autónoma de Honduras
Bases de datos II

Tarea #3: PLSQL-Condiciones, Ciclos, Cursores, Procedimientos y Funciones

Fecha de Entrega: martes 11 de octubre - 11:59pm

Subir el script SQL a un repositorio en GitHub (Al mismo repo que ya creo para las otras tareas).

Instrucciones: Resolver los siguientes ejercicios:

1. Bloques Anónimos

- a. Construir un bloque anonimo donde se declare un cursor y que imprima el nombre y sueldo de los empleados (utilice la tabla employees). Si durante el recorrido aparece el nombre Peter Tucker (el jefe) se debe genera un RAISE_APPLICATION_ERROR indicando que no se puede ver el sueldo del jefe.
- b. Crear un cursor con parámetros para el parametro id de departamento e imprima el numero de empleados de ese departamento (utilice la cláusula count).
- c. Crear un bloque que tenga un cursor de la tabla employees.
 - i. Por cada fila recuperada, si el salario es mayor de 8000 incrementamos el salario un 2%
 - ii. Si es menor de 8000 incrementamos en un 3%

2. Funciones

- a. Crear una función llamada CREAR_REGION
 - A la función se le debe pasar como parámetro un nombre de región y debe devolver un número, que es el código de región que calculamos dentro de la función.
 - Se debe crear una nueva fila con el nombre de esa REGION
 - El código de la región se debe calcular de forma automática. Para ello se debe averiguar cual es el código de región más alto que tenemos en la tabla en ese momento, le sumamos 1 y el resultado lo ponemos como el código para la nueva región que estamos creando
 - Debemos controlar los errores en caso que se genere un problema
 - La función debe devolver el número/ código que ha asignado a la región.
 - En el script debe colocar la funcion y el bloque para llamar la función.

3. Procedimientos

- a. Construya un procedimiento almacenado que haga las operaciones de una calculadora, por lo que debe recibir tres parametros de entrada, uno que contenga la operación a realizar (SUMA, RESTA, MULTIPLICACION, DIVISION), num1, num2 y declare un parametro de retorno e imprima el resultado de la operación. Maneje posibles excepciones.
- b. Realice una copia de la tabla employee, utilice el siguiente script:

```
CREATE TABLE
EMPLOYEES_COPIA
(EMPLOYEE_ID NUMBER (6,0) PRIMARY KEY,
FIRST_NAME VARCHAR2(20 BYTE),
LAST_NAME VARCHAR2(25 BYTE),
EMAIL VARCHAR2(25 BYTE),
PHONE_NUMBER VARCHAR2(20 BYTE),
HIRE_DATE DATE,
JOB_ID VARCHAR2(10 BYTE),
SALARY NUMBER(8,2),
COMMISSION_PCT NUMBER(2,2),
MANAGER_ID NUMBER(6,0),
DEPARTMENT_ID NUMBER(4,0)
);
```

Rellene la tabla employees_copia utilizando un procedimiento almacenado, el cual no recibirá parametros unicamente ejecutara los inserts en la nueva tabla, imprima el codigo de error en caso de que ocurra y muestre un mensaje por pantalla que diga "Carga Finalizada".