

EJERCICIO RESUELTO

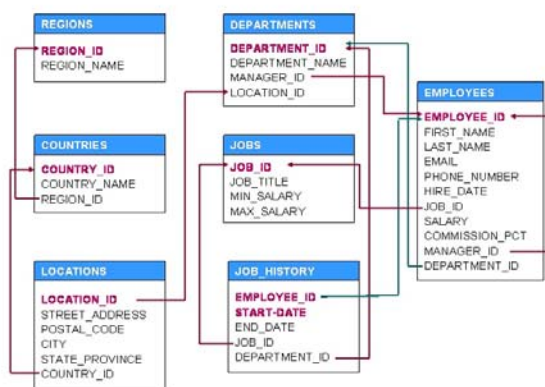
Módulo: BASES DE DATOS

Procedimientos

Descripción:

Sobre el caso práctico de la base de datos del diagrama HR crear las diferentes funciones que se solicitan.

- Creación de procedimiento almacenado que permita mostrar los datos de un empleado y su salario, pasando como parámetro el código de empleado.
- Creación de un procedimiento almacenado que muestre los registros EMPLOYEE_ID, MANAGER_ID, DEPARTMENT_ID de cualquiera de los empleados según el EMPLOYEE_ID, así mismo, debemos mostrar el campo DEPARTMENT_NAME sólo cuando el DEPARTMENT_ID sea igual a 50.



Objetivos:

- Identificar la sintaxis de creación de procedimientos.
- Verificar los diferentes métodos para ejecutar procedimientos.

Recursos:

- Acceso a Internet.
- Software ofimático.
- Oracle



Resolución:

Creación de procedimiento almacenado que permita mostrar los datos de un empleado y su salario, pasando como parámetro el código de empleado.

```
--Declaramos el procedimiento Busca_Empleado
CREATE OR REPLACE PROCEDURE Busca_Empleado (Codigo IN NUMBER)
IS
--Creamos las variables.
V_Cod_Empleado    NUMBER(6);
V_Nombre           VARCHAR2(20);
V_Apellido         EMPLOYEES.LAST_NAME%Type;
V_Salario          EMPLOYEES.SALARY%Type;
BEGIN
    --Realizamos la consulta
    SELECT employee_id, first_name, last_name, salary
        INTO V_Cod_Empleado, V_Nombre, V_Apellido, V_Salario
    FROM employees
    WHERE employee_id=codigo;

    --Mostramos el resultado
    DBMS_OUTPUT.PUT_LINE ('Codigo : '|| V_Cod_Empleado);
    DBMS_OUTPUT.PUT_LINE ('Nombre : '|| V_Nombre);
    DBMS_OUTPUT.PUT_LINE ('Apellido : '|| V_Apellido);
    DBMS_OUTPUT.PUT_LINE ('Salario : '|| V_Salario);
END;
```

Creación de un procedimiento almacenado que muestre los registros EMPLOYEE_ID, MANAGER_ID, DEPARTMENT_ID de cualquiera de los empleados según el EMPLOYEE_ID, así mismo, debemos mostrar el campo DEPARTMENT_NAME sólo cuando el DEPARTMENT_ID sea igual a 50.

```
CREATE OR REPLACE PROCEDURE Ver_Empleados (
--Parámetro de
pIn_Employee_Id    IN NUMBER,--Entrada
pOut_Employee_Id    OUT EMPLOYEES.EMPLOYEE_ID%TYPE,--Salida
pOut_Manager_Id     OUT EMPLOYEES.MANAGER_ID%TYPE,--Salida
pOut_Department_Id  OUT EMPLOYEES.DEPARTMENT_ID%TYPE,--Salida
pOut_Department_Name
                    OUT DEPARTMENTS.DEPARTMENT_NAME%TYPE,--Salida
)
AS
BEGIN
    --Recuperar datos de la tabla employees
    BEGIN
        SELECT employee_id, manager_id, department_id,
```

```
        INTO pOut_Employee_Id,  
        pOut_Manager_Id,  
        pOut_Department_Id,  
        FROM employees WHERE employee_id = pIn_Employee_Id;  
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        pOut_Employee_Id:=NULL;  
        pOut_Manager_Id:=NULL;  
        pOut_Department_Id:=NULL;  
END;  
  
-- Recuperar el DEPARTMENT_NAME  
BEGIN  
    IF (pOut_Department_Id = 50) THEN  
        BEGIN  
            SELECT department_name  
            INTO pOut_Department_Name  
            FROM departments  
            WHERE department_id = pOut_Department_Id;  
        EXCEPTION  
            WHEN NO_DATA_FOUND  
            THEN pOut_Department_Name:=NULL;  
            --Null, no hay datos a pesar de ser el dpto 50  
        END;  
    ELSE  
        BEGIN  
            pOut_Department_Name:=0;  
            --0, por ser diferente a dpto 50  
        END;  
    END IF;  
END;  
END Ver_Empleados;
```