



ACTIVIDADES TEMA 6 PARTE IV PL/SQL PAQUETES E INTRODUCCIÓN A LAS EXPECCIONES

1. *Crear un paquete llamado MIPACK que tenga los siguientes procedimientos y funciones:*
 - a. *Procedimiento para que a partir de un emp_no, busque y visualice todos los datos del empleado (BUSQ_EMP) . Utilizar las excepciones para detectar cuando no existe un empleado*
 - b. *Una función que a partir del dept_no devuelva el nombre del departamento (BUSQ_NOM_DEP). Utilizar las excepciones para detectar cuando no existe un departamento*
 - c. *Una función que a partir de un emp_no calcule su sueldo (CALC_SUELDO). Salario + nomina, recordar que si la nómina es null será necesario convertirla a número NVL*
 - d. *Un procedimiento que visualice todos los empleados de un departamento que se pasa por parámetro (VER_EMPLEADOS_DEP)*

Recordar que primero hay que definir la cabecera y luego el cuerpo del paquete.

NOTA: Realizar el proceso en un principio solo para el punto A y cuando funcione ir ampliándolo sucesivamente con el punto B, punto C,...

Recordar:

- *el objetivo de crear un paquete es tener reunido en un solo sitio todos los procedimientos y funciones*
- *para usar un procedimiento o una función de un paquete NOMPAQUETE.nom_procd o NOMPAQUETE.nom_func*

2. *Crear un bloque que a partir de un número de empleado introducido por teclado muestre*
 - *El departamento al que pertenece*
 - *Los datos del empleado*
 - *Y el sueldo que cobra*

Utilizar las funciones y procedimientos del paquete.

3. *Añadir MIPACK los siguientes procedimiento:*
 - *Función llamada ULT_EMPNO que permita calcular el último emp_no que existe en la tabla EMPLE*
 - *Procedimiento para insertar un departamento INSERT_DEP. Se pasarán como parámetros el número de departamento, el nombre del departamento y la localidad. Habrá que validar que no exista ya dicho departamento utilizando para ello la función ya existente en MIPACK*
 - *Procedimiento para borrar un departamento. Se pasarán los mismos parámetros que en el apartado anterior. Deberá de verificarse que dicho departamento existe en DEPART*



4. Con el objetivo de tener organizados todos los procedimientos y funciones diseñar dos paquetes `PACK_EMPLE` y `PACK_DEPART` y realizar los siguientes ejercicios

- **Copiar los subprogramas de `MIPACK` y pegarlos en el correspondiente paquete `PACK_EMPLE` o `PACK_DEPART` según corresponda**
- **Diseñar los siguientes subprogramas para cada uno de los paquetes**

`PACK_EMPLE`:

- **`Busq_jefe`:** que pasado como parámetros un `dept_no` y un oficio devuelva el número de empleado que es el jefe.
- **`Insertar_nuevo_empleado`:** permite insertar un empleado nuevo. Los datos nombre, oficio, departamento se introducirán por teclado; el numero de empleado será el siguiente al mayor `emp_no` que existe en la tabla, como fecha de alta la fecha del sistema, la comisión será nula, como jefe (`DIR`) el dir que tiene asignado los usuarios dicho oficio para ese departamento y como salario el menor de los salarios para dicho oficio.
- **`Empno_Emple`:** función que devuelve el `emp_no` de un empleado a partir de su apellido.
- **`Modif_comision_empno`:** procedimiento que permita modificar la comisión de un empleado que se pasa como parámetro.
- **`Incrementar_salario`:** procedimiento que pasado dos parámetros `dept_no` y valor, permita modificar el salario incrementándolo con dicho valor para todos los empleados del departamento

`GEST_DEPART`:

- **`Insertar_nuevo_depart`:** permite insertar un departamento nuevo. El procedimiento recibe el nombre y la localidad del nuevo departamento. Creará el nuevo departamento comprobando que el nombre no se duplica y le asignará como número de departamento la decena siguiente al último número de departamento utilizado.
- **`Borrar_depart`:** permite borrar un departamento. El procedimiento recibirá dos números de departamento, de los cuales el primero corresponde al departamento que queremos borrar y el segundo al departamento al que pasarán los empleados del departamento que se va a eliminar. El procedimiento se encargará de realizar los cambios oportunos en los números de departamento de los empleados correspondientes.
- **`Modif_loc_depart`:** modifica la localidad del departamento. El procedimiento recibirá el número del departamento que se modifica y la nueva localidad y realizará el cambio solicitado.
- **`Visualizar_datos_depart`:** visualizará los datos de un departamento cuyo número se pasará en la llamada. Deberá de comprobar que exista el departamento utilizando una función específica para ello. En caso de existir además de los datos relativos al departamento, se visualizará el número de empleados que pertenecen actualmente al departamento.