**ACTIVIDADES TEMA 11**

Sobre la base de datos *Pedidos* lleve a cabo las siguientes operaciones:

Pedido (RefPed, FecPed)

LíneaPedido (RefPed, CodArt, CantArt)

Artículo (CodArt, DesArt, PVPArt)

1. Cree un procedimiento llamado *EliminarPedido* que reciba la referencia de un pedido y que muestre en pantalla dicha referencia y la fecha del pedido y se encargue de eliminarlo de la base de datos. Debe mostrase un mensaje indicando tal hecho.
2. Cree una función llamada *NumPedidos* que reciba la descripción de un artículo y que devuelva un número entero que indique en cuántos pedidos diferentes ha sido solicitado dicho artículo.

Sobre la base de datos *Empresa* lleve a cabo las siguientes operaciones:

Depart (dept\_no, dnombre, loc)

Emple (emp\_no, apellido, oficio, dir, fecha\_alt, salario, comision, dept\_no)

1. Escriba una función llamada *NumSubordinados* que reciba el número de un empleado de la tabla *Emple* y que devuelva el número de subordinados que tiene ese empleado, o lo que es lo mismo, el número de empleados de los cuales es director o jefe directo. Recuerde que el atributo d*ir* de la tabla *Emple* indica el número del empleado director o jefe directo. Escriba un ejemplo de llamada a la función.
2. Cree una función llamada *SalarioJefe* que reciba el apellido de un empleado y que devuelva un número real que tome el valor del salario del empleado dividido entre el número de empleados subordinados que tenga. En caso de que el empleado no tenga subordinados, la función deberá devolver el valor -1.
3. Cree un procedimiento llamado *AñadirDepar* que reciba el nombre de un departamento y su localidad. Se debe insertar ese departamento en la tabla *Depart* asignándole como número el que resulte de sumar 10 al número más alto de los departamentos de la empresa. Al final muestre un mensaje con el texto: “Se ha creado un nuevo departamento con el nº xxx”.
4. Cree un procedimiento llamado *ModificarSalario* que reciba el número de un empleado y un número entero. El procedimiento debe modificar el salario del empleado con el nº recibido como primer parámetro según el porcentaje pasado como segundo parámetro y mostrar el salario del empleado antes y después de la modificación.
5. Cree un procedimiento llamado *AsignarComision* que reciba un número de empleado y compruebe su salario en la tabla *Emple* de la base de datos *Empresa*. Si el salario del empleado es menor que 1000 €, se le debe asignar una comisión que será el 5% del salario; en caso de que su salario sea superior o igual a 1000 €, pero inferior a 2000 €, se le deberá asignar una comisión igual al 2,5% del salario; en caso de que cobre 2000 € o más, se le pondrá a 0 € la comisión. Muestre un mensaje como el siguiente: “Al empleado nº XXXX se le ha asignado una comisión de YY.YY €”.
6. Cree una función llamada *PorcenSalarial* que reciba el número de un empleado y nos devuelva un número real que indique el porcentaje que supone el salario del empleado en relación con la suma salarial de todos los empleados de su departamento.
7. Cree un procedimiento llamado *ModificarSalariosDepar* que reciba el número de un departamento y un número real. El procedimiento, en primer lugar, obtendrá el número de empleados que trabajan en el departamento cuyo número se ha recibido como primer parámetro. En caso de que no trabaje ningún empleado en dicho departamento, se mostrará el mensaje “En el dpto. nº xxx no trabaja ningún empleado”. En caso contrario, se deberá modificar el salario de los empleados del departamento cuyo número se ha pasado como primer parámetro en base al porcentaje recibido como segundo parámetro, después de lo cual se mostrará el mensaje “Se ha modificado el salario de todos los empleados del dpto. nº xx en un yyy,yy%”.

Sobre la base de datos *geografia* realice las siguientes operaciones:

Localidades (id\_localidad, nombre, poblacion, n\_provincia)

Provincias (n\_provincia, nombre, superficie, id\_capital, id\_comunidad)

Comunidades (id\_comunidad, nombre, id\_capital)

1. Cree un procedimiento llamado *MostrarComunidad* que reciba el nombre de una comunidad autónoma y que muestre el número de provincias y el número de localidades de que consta dicha comunidad autónoma. Si dicha comunidad autónoma no está en la base de datos, se deberá mostrar el mensaje: “La comunidad autónoma llamada XXXX no está en la base de datos”.
2. Cree una función llamada *ProvinciaMayor* que reciba el nombre de una comunidad autónoma y devuelva el nombre de la provincia con mayor superficie de la misma.