**ACTIVIDADES TEMA 13**

Lleve a cabo las siguientes operaciones sobre las bases de datos *Empresa* y *Pedidos*, cuyos esquemas relacionales son los siguientes:

Depart (dept\_no, dnombre, loc)

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

Pedido (RefPed, FecPed)

LíneaPedido (RefPed, CodArt, CantArt)

Artículo (CodArt, DesArt, PVPArt)

1. Cree una nueva tabla dentro de la base de datos *Pedidos* llamada *ArticulosAntiguos*. Esta tabla tendrá los mismos atributos que la tabla *Articulo* y servirá para contener los datos de los artículos eliminados de la tabla *Articulo*. Cree un disparador llamado *BorradoArticulos* que se ejecute cada vez que se realice un borrado sobre la tabla *Articulo*, de manera que el artículo que se borre se añada a la tabla *ArticulosAntiguos*.
2. Cree una tabla llamada *AuditoriaPrecios* con un solo atributo llamado *Linea* de tipo varchar(100). Cree después un disparador llamado *AuditarPrecios* que dé fé de las modificaciones de precios sobre los artículos de la base de datos. Cada vez que se incremente o decremente el precio de un artículo más de un 10% se debe añadir una fila a la tabla *AuditoriaPrecios* con el texto: “El articulo xxxx ha cambiado su precio de yyyy a zzzz”, donde xxxx es la descripción del artículo, yyyy el precio antes de la modificación y zzzz el precio después de la modificación.
3. Cree una nueva tabla llamada *EmpleadosAntiguos* copia de la tabla *Emple* y sin datos, pero con dos atributos adicionales: *fecha\_baja*, de tipo fecha, y *finiquito* de tipo float. Cree un trigger llamado *BajaEmpleado* sobre la tabla *Emple*, de manera que cuando se borre a un empleado de la tabla *Emple*, se añada dicho empleado a la nueva tabla *EmpleadosAntiguos*. Los datos del empleado en la tabla *EmpleadosAntiguos* serán los mismos que aparecen en la tabla *Emple* y además al atributo *fecha\_baja* se le asignará la fecha del día en el que se elimina al empleado y al atributo *finiquito* se le asignará un valor que se calculará en función de su oficio:
   * Si el empleado tiene el oficio EMPLEADO o VENDEDOR, el finiquito se calculará multiplicando 30 por el salario diario del empleado y por el número de meses enteros de antigüedad divididos entre 12 (número de meses de un año). El salario diario del empleado se calculará dividiendo entre 30 su salario mensual.
   * Si el empleado tiene cualquier otro oficio, el finiquito se calculará multiplicando 20 por el salario diario del empleado y por el número de meses enteros de antigüedad divididos entre 12.

NOTA: La antigüedad se debe calcular como el número de meses transcurridos entre la fecha del día de hoy (fecha que nos devuelve la función current\_date()) y la fecha de alta del empleado en la empresa (atributo *fecha\_alt*). Para llevar a cabo este cálculo MySQL nos proporciona la función *timestampdiff (unidad\_tiempo, fecha1, fecha2)* que recibe como parámetro una unidad de tiempo y dos fechas tal que fecha2 es superior o igual a fecha1. Esta función nos devuelve la diferencia entre esas dos fechas en la unidad de tiempo indicada como primer parámetro

1. Se desea almacenar dentro de la base de datos *Empresa* información sobre las modificaciones salariales de los empleados. Para ello se debe crear la tabla *CambiosSalariales* con la siguiente orden SQL:

create table CambiosSalariales

(emp\_no int,

apellido varchar(40) not null,

SalAnt float,

SalNue float,

Porcen float,

Fecha date);

Cree un disparador llamado *RegistrarCambiosSalariales* que cada vez que se efectúe una modificación en el salario de un empleado, añada un registro a la tabla *CambiosSalariales*, almacenando el número del empleado al que se le ha cambiado el salario (valor nuevo de este campo), su apellido (valor nuevo), el salario antes del cambio, el salario después del cambio, el porcentaje de modificación del salario y la fecha en la que se efectúa el cambio. NOTA: El porcentaje de modificación del salario se calcula mediante la siguiente fórmula:



Escriba una orden de modificación del salario de uno o varios empleados y compruebe que se almacenado correctamente información en la tabla *CambiosSalariales*.

1. Cree una nueva tabla en la base de datos *Empresa* llamada *ComisionesAntig* que tenga los siguientes atributos:
   * *Emp\_no*: número de empleado, del mismo tipo que el campo homónimo de la tabla *Emple*. Este campo será clave primaria y clave ajena al atributo homónimo de la tabla *Emple*.
   * *Trienios*: campo entero.
   * *Plus*: campo float.

Cree un evento llamado *CalcularPlusesTrienios* que se ejecute en el momento de crearlo y cada trimestre y que se encargue de calcular los pluses por trienios de los empleados del siguiente modo: En primer lugar, borrará todo el contenido de la tabla *ComisionesAntig*. A continuación, recorrerá toda la tabla *Emple* y para cada empleado que lleve más de tres años trabajando en la empresa, se añadirá un registro a la tabla *ComisionesAntig*, almacenando el número del empleado, el número de trienios enteros que lleva trabajando en la empresa y su plus, que será de 45 € por cada trienio.

1. Cree dentro de la base de datos *Pedidos* una nueva tabla vacía llamada *EstadisticasArticulos* para almacenar mensualmente información estadística sobre los artículos de la tabla *Articulo*. Cree esta tabla con la siguiente orden SQL:

create table EstadisticasArticulos

(CodArt char(5) not null,

DesArt varchar(30) not null,

PVPArt float,

UnidadesVendidas int,

Fecha date);

Cree un evento llamado *GenerarEstadisticaArt* que se ejecute en el momento de crearlo y cada mes y que almacene sobre cada artículo su código, descripción, precio, el número de unidades totales solicitadas de ese artículo en todos los pedidos de la base de datos y la fecha del día en que se ejecuta el evento.

1. Cree una nueva tabla vacía llamada *ArticulosObsoletos* con la misma estructura que la tabla *Articulo* de la base de datos *Pedidos* y un campo adicional llamado *FecBaja* de tipo fecha.

Cree un evento llamado *BajaArticulosObsoletos* que se encargue de borrar de la tabla *Articulo* de la base de datos *Pedidos* los artículos obsoletos. Se consideran artículos obsoletos aquellos que se pidieron por última vez hace cinco años o más. Pues bien, estos artículos, además de ser eliminados de las tablas *LineaPedido* y *Articulo*, deben ser añadidos a la nueva tabla *ArticulosObsoletos*, asignando al atributo *FecBaja* la fecha del día en que el artículo es eliminado de la tabla *Articulo*. Este evento se ha de ejecutar en el momento de crearlo y cada mes.