

```

/*          SOLUCIONES EJERCICIOS DML. ACTUALIZACIÓN DE DATOS
*/

USE Gestion8;

--1. Crear una copia de empleados con el nombre Nuevaempleados, lo mismo para la
tabla de oficinas (Nuevaoficinas), la de clientes (Nuevaclientes), la de productos
(Nuevaproduktos) y la de pedidos (Nuevapedidos).
SELECT * INTO Nuevaempleados FROM empleados;
SELECT * INTO Nuevaoficinas FROM oficinas;
SELECT * INTO Nuevaclientes FROM clientes;
SELECT * INTO Nuevaproduktos FROM produktos;
SELECT * INTO Nuevapedidos FROM pedidos;

--2. Añadir una nueva oficina para la ciudad de Elx, con el número de oficina 40,
con director 108 y con un objetivo de 100.000€.

INSERT INTO oficinas (oficina, ciudad, dir, objetivo) VALUES (40, 'Elx', 108,
100000);

--3 . Añadir un nuevo empleado numemp:115, nombre:Luis Garcia, oficina:40, sin
objetivo ni ventas ni director
INSERT INTO empleados (numemp, nombre, oficina) VALUES (115, 'Luis Garcia', 40);

--4. Crear una tabla (Oeste) con todas las oficinas del Oeste, la tabla tendrá los
mismos datos que oficinas.

SELECT * INTO Oeste
FROM oficinas
WHERE region = 'Oeste';

--5. Añadir a la tabla Oeste las oficinas del Este.

INSERT INTO Oeste SELECT *
                        FROM oficinas
                        WHERE region = 'Este';

--6. Añadir a la oficina 40 un empleado Antonio García López, con número de empleado
435, contratado hoy sin ventas con cuota 1200,45 € con título Vendedor, de momento
no le asignaremos jefe.

INSERT INTO empleados (numemp, nombre, titulo, contrato, ventas, cuota, oficina)
VALUES (435, 'Antonio García López', 'Vendedor', GETDATE(), 0, 1200.45, 40);

--7. Añadir a la oficina 40 otro empleado, Luis Valverde, con número de empleado
436, con los mismos datos que el anterior pero su jefe será el director de la
oficina 40 (no sabemos qué número tiene).

INSERT INTO empleados (numemp, nombre, titulo, contrato, ventas, cuota, oficina,
jefe)
SELECT 436, 'Luis Valverde', 'Vendedor', GETDATE(), 0, 1200.45, 40, dir
FROM oficinas WHERE oficina = 40;

--8. Subir un 5% el precio de todos los productos del fabricante QSA.

UPDATE produktos SET precio = ROUND(precio * 1.05,2)
WHERE idfab = 'qsa';

```

--9. Poner a cero las ventas y cuota del empleado Luis Garcia, si hay varios con el mismo nombre actualizarlos todos.

```
UPDATE empleados set ventas=0, cuota=0
WHERE nombre = 'Luis Garcia';
```

--10. Cambiar los empleados de la oficina 40 a la oficina 30.

```
UPDATE empleados SET oficina = 30
WHERE oficina = 40;
```

--11. Actualizar los pedidos del fabricante rei dejando como representante el empleado asignado al cliente.

```
UPDATE pedidos SET rep=repclie
FROM pedidos inner join clientes ON clie=numclie
WHERE rep <> repclie and fab ='rei'
```

--12. Actualizar el campo objetivo de la oficina 30 con las cuotas de los empleados asignados a ella.

```
UPDATE oficinas set oficinas.objetivo =(SELECT SUM (cuota)
from empleados where oficinas.oficina= empleados.oficina)
where oficina=30;
```

--13. Actualizar el precio de los productos de BIC obteniendo el nuevo valor del precio medio del artículo vendido en los pedidos (si hay pedidos).

```
UPDATE productos SET precio = round((SELECT avg(importe/cant) FROM pedidos WHERE
fab=idfab and producto=idproducto),2)
WHERE idfab='bic' AND EXISTS (SELECT * FROM pedidos WHERE fab=idfab and
producto=idproducto);
```

```
SELECT *
FROM productos
WHERE idfab='bic' -- La SELECT es para ver el resultado
-- O bien
UPDATE productos SET precio = round(ISNULL((SELECT avg(importe/cant) FROM pedidos
WHERE fab=idfab and producto=idproducto),precio),2)
WHERE idfab='bic' ;
```

```
SELECT *
FROM productos
WHERE idfab='bic' -- La SELECT es para ver el resultado
```

--14. Pasar los pedidos de octubre 1989 a diciembre 2008.

```
UPDATE pedidos SET fechapedido=DATEADD(month,230,fechapedido)
WHERE year(fechapedido)=1989 and month(fechapedido)=10;
```

--15. Pasar ahora los pedidos anteriores al mes y año actual.

```
UPDATE pedidos SET
fechapedido=DATEADD(month,DATEDIFF(month,fechapedido,getdate()),fechapedido)
WHERE year(fechapedido)=2008 and month(fechapedido)=12;
```

--16. Queremos actualizar el importe de los pedidos del mes actual con el precio almacenado en la tabla productos.

```
SELECT codigo, numpedido, fechapedido, cant, importe, importe/cant AS [precio
pedido], precio
FROM pedidos inner join productos ON fab=idfab and producto = idproducto
WHERE YEAR(fechapedido)=YEAR(GETDATE()) and MONTH(fechapedido)=MONTH(GETDATE())
/*
Actualizar ahora la tabla de pedidos cambiando los importes para que el precio
unitario corresponda con el precio del producto.*/
UPDATE pedidos SET importe=cant*precio
FROM pedidos inner join productos ON fab=idfab and producto = idproducto
WHERE YEAR(fechapedido)=YEAR(GETDATE()) and MONTH(fechapedido)=MONTH(GETDATE())AND
precio IS NOT NULL;
```

--17. Hemos visto que hay productos que no tienen precio, ahora vamos a actualizar el precio de estos productos con el precio medio utilizado en los pedidos donde aparecen.

-- Primero sacamos los productos que queremos actualizar con los pedidos correspondientes:

```
SELECT idfab, idproducto, precio, codigo, numpedido, fechapedido, importe,
importe/cant
FROM productos left join pedidos on idfab=fab AND idproducto=producto
WHERE precio IS NULL;

UPDATE productos SET precio = (SELECT ROUND(AVG(importe/cant),2) FROM pedidos WHERE
fab=idfab AND producto=idproducto)
WHERE precio IS NULL
```

--18. Eliminar el empleado 435

```
DELETE FROM empleados
WHERE numemp = 435;
```

--19. Eliminar los pedidos del representante 105.

```
DELETE FROM pedidos
WHERE rep = 105;
```

--20. Eliminar los pedidos del representante Luis Antonio.

```
DELETE FROM pedidos
WHERE rep IN (SELECT numemp FROM empleados
              WHERE nombre = 'Luis Antonio');
```

--21. Eliminar las oficinas que no tengan empleados.

```
DELETE FROM oficinas
WHERE NOT EXISTS (SELECT *
                  FROM empleados WHERE empleados.oficina = oficinas.oficina);
```

--22.1. Intenta eliminar el empleado 102.

```
DELETE FROM empleados WHERE numemp=102;
```

-- Te saldrá un error Instrucción DELETE en conflicto con la restricción...

--22.2. Reflexiona sobre el error y elabora una consulta que liste los empleados que pueden dar problemas.

-- No podemos eliminar empleados que sean padres en alguna relación, es decir empleados que sean jefes,

-- que dirijan alguna oficina, que tengan algún pedido, que sirvan algún cliente.

```
SELECT jefe FROM empleados
UNION
SELECT dir FROM oficinas
UNION
SELECT rep FROM pedidos
UNION
SELECT repclie FROM clientes
ORDER BY jefe
```

-- o bien

```
SELECT numemp from empleados
WHERE numemp IN (select jefe from empleados)
OR numemp IN (SELECT dir FROM oficinas)
OR numemp IN (SELECT rep FROM pedidos)
OR numemp IN (SELECT repclie FROM clientes)
```

-- En esta segunda opción no aparece el valor NULL, aunque para lo que la vamos a utilizar da igual

--22.3. Elabora ahora una consulta que liste los que se pueden borrar sin problemas

```
SELECT numemp FROM empleados
EXCEPT
SELECT jefe from empleados
EXCEPT
SELECT dir FROM oficinas
EXCEPT
SELECT rep FROM pedidos
EXCEPT
SELECT repclie FROM clientes
```

-- O bien

```
SELECT numemp from empleados
WHERE NOT EXISTS (SELECT * from empleados e WHERE e.jefe = empleados.numemp)
AND NOT EXISTS (SELECT * FROM oficinas WHERE oficinas.dir = empleados.numemp)
AND NOT EXISTS (SELECT * FROM pedidos WHERE pedidos.rep = empleados.numemp)
AND NOT EXISTS (SELECT * FROM clientes WHERE clientes.repclie = empleados.numemp)
```

-- Utilizo NOT EXISTS en vez de NOT IN para no tener que preocuparme por los valores nulos

--22.4 Borra el primero de la lista, éste sí lo puedes borrar.

```
DELETE empleados WHERE numemp=112
```

--23. Eliminar los pedidos de productos de ACI cuyo precio de venta en el pedido no corresponda con el precio unitario del producto de la tabla de productos.

```
DELETE FROM pedidos
WHERE fab = 'ACI' AND Importe/cant <>
(SELECT precio FROM productos WHERE fab = idfab AND idproducto = producto);
```

```

--24.1 Empieza por eliminar los datos de las tablas.
TRUNCATE TABLE pedidos
-- utilizamos TRUNCATE porque la tabla no está referenciada y para que el contador
empiece desde 1 otra vez.
DELETE FROM productos
DELETE FROM clientes
UPDATE oficinas SET dir = NULL; -- Para poder borrar los empleados
UPDATE empleados SET oficina=NULL, jefe=NULL; -- Para poder borrar las oficinas y
los empleados respectivamente.
DELETE FROM oficinas;
DELETE FROM empleados;

--24.2. Inserta los datos de las tablas Nueva.. a las tablas normales.

INSERT INTO oficinas (oficina, ciudad, region,objetivo,ventas)
    SELECT oficina, ciudad, region,objetivo,ventas
    FROM Nuevaoficinas;
--No podemos rellenar todavía la columna Dir ya que no tenemos los empleados dado de
alta

INSERT INTO empleados (numemp, nombre, edad, oficina,titulo,contrato,cuota,ventas)
    SELECT numemp, nombre, edad, oficina,titulo,contrato,cuota,ventas
    FROM NuevaEmpleados;
--Lo mismo para las la columna jefe

UPDATE empleados SET jefe=nueva.jefe
FROM empleados INNER JOIN NuevaEmpleados nueva ON empleados.numemp=nueva.numemp

--Estas dos últimas sentencias se podían haber resumido en una:
--INSERT INTO empleados
--    SELECT *
--    FROM NuevaEmpleados;
--Las hemos dejado en dos porque con otros SQLs no se podría hacer en un sólo paso.

UPDATE oficinas SET dir = (SELECT dir FROM Nuevaoficinas nueva
                           WHERE oficinas.oficina=nueva.oficina)

INSERT INTO clientes (numclie,nombre,repclie,limitecredito)
    SELECT numclie,nombre,repclie,limitecredito
    FROM Nuevaclientes;

INSERT INTO productos
    SELECT *
    FROM Nuevaproductos;

INSERT INTO pedidos (numpedido,fechapedido,clie,rep,fab,producto,cant,importe)
    SELECT numpedido,fechapedido,clie,rep,fab,producto,cant,importe
    FROM Nuevapedidos;

```