

/* PARCIAL 1: 2do.SEM 2023 (Grupo: N3Crem) (25 puntos)

Ejercicios:

1) 5 puntos

2) 15 Puntos:

2.a) 3 puntos

2.b) 3 puntos

2.c) 3 puntos

2.d) 3 puntos

2.e) 3 puntos

3) 5 puntos

*/

use Northwind

set dateformat dmy

Go

/*

1) Se pide obtener los Clientes más valiosos por categoría de producto:
Encontrar los 10 clientes que han gastado más dinero en productos de las 3
categorías que mas han vendido.
Incluir en la consulta el nombre del cliente, el ID y nombre de la categoría del
producto y el importe total gastado por cada cliente en esa categoría.
(5 puntos)

*/

```
SELECT TOP 10 p.CategoryID,D.CategoryName, c.CompanyName, SUM(od.Quantity *  
od.UnitPrice) AS TotalGastado
```

```
FROM Customers AS c
```

```
INNER JOIN Orders AS o ON c.CustomerID = o.CustomerID
```

```
INNER JOIN OrderDetails AS od ON o.OrderID = od.OrderID
```

```
INNER JOIN Products AS p ON od.ProductID = p.ProductID
```

```
inner join Categories D ON D.CategoryID =p.CategoryID
```

```
INNER JOIN (select top 3 p.CategoryID, SUM(od.Quantity * od.UnitPrice)  
AS CategTotalRecaudado
```

```
from OrderDetails AS od JOIN Products AS p  
ON od.ProductID = p.ProductID
```

```
Group by p.CategoryID
```

```
Order by CategTotalRecaudado Desc
```

```
) as Subquery
```

```
ON p.CategoryID = Subquery.CategoryID
```

```
GROUP BY p.CategoryID, D.CategoryName,c.CompanyName
```

```
ORDER BY TotalGastado DESC,D.CategoryName,c.CompanyName
```

*/

/*

2.a.) Insertar un nuevo producto:

Inserta un nuevo producto en la tabla "Products" con Categoría 'SeaFood' y un
valor de precio unitario 10% superior al precio promedio de todos los productos
en la categoría 'SeaFood'.

(3 puntos)

*/

```
INSERT INTO Products (ProductName, CategoryID, UnitPrice)
```

```
SELECT 'Nuevo Producto', p.CategoryID,
```

```
(select AVG(p1.UnitPrice) * 1.10
```

```
From Products p1 JOIN Categories c1
```

```
ON p1.CategoryID = c1.CategoryID
```

```
WHERE c1.CategoryName = 'SeaFood' )
```

```
FROM Products p JOIN Categories c
```

```
ON p.CategoryID = c.CategoryID
```

```
WHERE c.CategoryName = 'SeaFood'
```

/*

2.b) Actualizar el precio de los productos, para aquellos productos que han pedidos por los clientes específicos: 'ALFKI', 'HANAR', 'VICTE', 'CHOPS', 'WELLI'. El Precio Unitario se debe incrementar en un 10% el precio de todos los productos vendidos a los clientes específicos. (3 puntos)

*/

```
UPDATE Products SET UnitPrice = UnitPrice * 1.10
WHERE ProductID IN ( SELECT distinct od.ProductID
                     FROM Orders o JOIN OrderDetails od
                               ON o.orderID = od.OrderID
                     WHERE o.CustomerID IN (
                           'ALFKI', 'HANAR', 'VICTE', 'CHOPS', 'WELLI' )
                     )
```

/*

2.c) Eliminar empleados sin órdenes: Elimina todos los empleados que no tienen ninguna orden asociada en la tabla "Orders". (3 puntos) */

```
DELETE FROM Employees WHERE EmployeeID NOT IN ( SELECT DISTINCT o.EmployeeID
                                                FROM Orders o)
```

/*

2.d) Actualizar el precio (Decrementar en un 15%), de los productos que tienen un precio unitario más alto que todos los Mayores Precio de cualquier producto en las otras categorías: (3 puntos) */

```
UPDATE Products SET UnitPrice = UnitPrice * (1-0.15)
WHERE UnitPrice > ALL ( SELECT MAX(p3.UnitPrice) FROM Products p3
                       group by p3.CategoryID )
```

/*

2.e) Eliminar las categorías sin productos: Elimina todas las categorías de productos que no tienen ningún producto asociado en la tabla "Products". (3 puntos) */

```
DELETE FROM Categories
WHERE CategoryID NOT IN ( SELECT DISTINCT CategoryID FROM Products );
```

/*

3.) (5 puntos)

CLASIFICAR a los CLIENTES en franjas en función del Volumen anual de Pedidos. Considerar para determinar la franja, si el importe Total de los Pedidos generados por el Cliente es Menor a \$25000 devuelva Franja 'BAJA', si esta entre \$25001 y \$75000 devuelva Franja 'MEDIA', si es mayor a \$75000 devuelva Franja 'ALTA'.

Solo se deben considerar los Clientes donde el "Total de los Pedidos Realizados en el año" sea superior al Promedio de Pedidos Realizados en el año anterior 1997.

La Consulta debe devolver: Identificador del Customer, Nombre del Customer, año, Total de los Pedidos Realizados en el año 1998 y la Franja a la que pertenece, los datos se deben mostrar ordenados por Franja y por CustomerID.

*/

```
SELECT c.CustomerID, c.CompanyName, T.TotalPedidosRealizadosAño,
       CASE
         WHEN T.TotalPedidosRealizadosAño < 5000 then 'BAJA'
         WHEN T.TotalPedidosRealizadosAño >= 5000 and
T.TotalPedidosRealizadosAño <= 25000 then 'MEDIA'
         WHEN T.TotalPedidosRealizadosAño > 20000 then 'ALTA'
       END as Franja
FROM Customers c JOIN (
  SELECT o.CustomerID, SUM(od.quantity * od.Unitprice)
    as TotalPedidosRealizadosAño
  FROM Orders o JOIN Orderdetails od ON od.orderID = o.orderID
  Where Year(o.orderdate) = 1998
  GROUP BY o.CustomerID
) T
ON c.CustomerID = T.CustomerID
WHERE T.TotalPedidosRealizadosAño > (
  SELECT AVG(od1.quantity * od1.Unitprice)
  FROM Orders o1 JOIN Orderdetails od1
    ON od1.orderID = o1.orderID
  Where Year(o1.orderdate) = 1997
    AND o1.CustomerID = c.CustomerID
  GROUP BY o1.CustomerID
)
ORDER BY Franja, CustomerID
```

-----FIN PARCIAL-----