Ejercicios SQL

1) Todas las ventas realizadas por MargaretPeacock al cliente Frankenversand.La salida debe contener los siguientes campos: Nombre y Apellido del Vendedor, Nombre del Cliente, ID de la Orden de Compra (tabla Orders), Nombre del Producto (tabla Products), Presentación (Unit, de tabla Products), Cantidad (tabla OrderDetails), Precio (tabla Products) y Total (calculado como Precio\*Cantidad).

SELECT e.FirstName as NombreVendedor, e.LastName as ApellidoVendedor, c.CustomerName as NombreCliente, o.OrderID, p.ProductName as NombreProducto, p.Unit as Presentacion, od.Quantity as Cantidad, p.Price as Precio, round(p.Price \* (od.Quantity) , 2) as Total

FROM Categories c

INNER JOIN Products p ON p.CategoryID = c.CategoryID

INNER JOIN Customers c ON c.CustomerID = o.CustomerID

INNER JOIN Employees e ON e.EmployeeID = o.EmployeeID

INNER JOIN OrderDetails od ON p.ProductID = od.ProductID

INNER JOIN Orders o ON o.OrderID = od.OrderID

WHERE c.CustomerName = "Frankenversand" and e.LastName = "Peacock"

2) Un descuento de 12% para aquellos productos con precio mayor a 75, un descuento del8% para productos con precio entre 50 y 75 (inclusive), y 4% para productos con precio mayor o igual a 30 y hasta 49.99.La salida debe contener los siguientes campos: Nombre del Producto, Presentación (Unidades), Precio Original, Descuento (%), Descuento ($) y Precio con Descuento.

CASE WHEN Price > 75 THEN "12%"

WHEN Price BETWEEN 50 and 75 THEN "8%"

WHEN Price BETWEEN 30 and 49.99 THEN "4&"

ELSE "0%" END as Descuento,

round(Price \* (CASE WHEN Price > 75 THEN 0.12

WHEN Price BETWEEN 50 and 75 THEN 0.08

WHEN Price BETWEEN 30 and 49.99 THEN 0.04 ELSE 0 END),2) as Descuento\_$,

round(Price - Price \* (CASE WHEN Price > 75 THEN 0.12

WHEN Price BETWEEN 50 and 75 THEN 0.08

WHEN Price BETWEEN 30 and 49.99 THEN 0.04 ELSE 0 END),2) as PrecioConDto

FROM Products

ORDER BY Descuento

3) El precio máximo y mínimo por cada categoría de los productos. La salida debe contener los siguientes campos: Nombre de la categoría, Precio Máximo, Precio Mínimoy debe ser ordenado por el precio máximo de forma descendente.

SELECT c.CategoryName as Categoria, MAX(p.Price) as PrecioMaximo, MIN(p.Price) as PrecioMinimo

FROM Categories c

INNER JOIN Products p ON c.CategoryID = p.CategoryID

GROUP BY c.CategoryName

ORDER BY PrecioMaximo DESC

4) Todas las ventas realizadas a los clientes de España.La salida debe contener los siguientes campos: Nombre del Cliente, ciudad y país del cliente, ID de la Orden de Compra (tabla Orders), Nombre del Producto (tabla Products), Presentación (Unit, de tabla Products), Cantidad (tabla OrderDetails), Precio (tabla Products) y Total (calculado como Precio\*Cantidad).

SELECT c.CustomerName as NombreCliente, c.City as Ciudad, c.Country as Pais, o.OrderID, p.ProductName as NombreProducto, p.Unit as Presentacion, od.Quantity as Cantidad, p.Price as Precio, round(p.Price \* (od.Quantity),2) as Total

FROM Categories c

INNER JOIN Products p ON p.CategoryID = c.CategoryID

INNER JOIN Customers c ON c.CustomerID = o.CustomerID

INNER JOIN Employees e ON e.EmployeeID = o.EmployeeID

INNER JOIN OrderDetails od ON p.ProductID = od.ProductID

INNER JOIN Orders o ON o.OrderID = od.OrderID

WHERE c.Country = "Spain"

5) Un recargode 7% para losproductos que se venden en cajasy un 5% para productos que se venden en botellas o paquetes, para el resto de los productos el recargo es del 2%.La salida debe contener los siguientes campos: Nombre del Producto, Presentación (Unidades), Precio Original, Recargo(%), Recargo($) y Precio con Recargo.

SELECT ProductName as NombreProducto, Unit as Presentacion, Price as Precio,

CASE WHEN Unit LIKE '%boxes%' THEN '7%'

WHEN Unit LIKE '%bottle%' THEN '5%'

ELSE'2%'END AS RecargoPorcentaje,

round(Price\*(CASE WHEN Unit LIKE '%boxes%' THEN 0.07

WHEN Unit LIKE '%bottle%' THEN 0.05

ELSE 0.02 END),2) AS RecargoDinero,

round(Price + Price\*(CASE WHEN Unit LIKE '%boxes%' THEN 0.07

WHEN Unit LIKE '%bottle%' THEN 0.05

ELSE 0.02 END),2)+ Price AS PrecioConRegargo

FROM Products

6) La cantidad de órdenes de compra y el total gastadopor cliente, para los clientes de Estados Unidos. La salida debe contener los siguientes campos: Nombre del Cliente, Cantidad de órdenes, Total (calculado comola suma total delPrecio\*Cantidad).

SELECT c.CustomerName as NombreCliente, count(distinct o.OrderID) as CantidadDeOrdenes, sum(p.Price \* od.Quantity) as Total

FROM Categories c

INNER JOIN Products p ON p.CategoryID = c.CategoryID

INNER JOIN Customers c ON c.CustomerID = o.CustomerID

INNER JOIN Employees e ON e.EmployeeID = o.EmployeeID

INNER JOIN OrderDetails od ON p.ProductID = od.ProductID

INNER JOIN Orders o ON o.OrderID = od.OrderID

WHERE c.Country = "USA"

GROUP BY NombreCliente

7) Un comisión del 20% para los emplearos que realizaron ventas por más de$100.000, del 10% para los que tuvieron ventas entre $40.000 y $99.999 y del 5% para los que vendieron entre $30.000 y $39000. La salida debe contener los siguientes campos: Nombre y Apellido del Vendedor, Cantidad de ventas(cantidad de órdenes, no de productos), Totalvendido($),Comisión(%).

SELECT e.FirstName as Nombre, e.LastName as Apellido, count(o.OrderID) as CantidadDeVentas, sum(p.Price \* od.Quantity) as TotalVendido,

CASE WHEN sum(p.Price \* od.Quantity) > 100000 THEN "20%"

WHEN sum(p.Price \* od.Quantity) BETWEEN 40000 and 99999 THEN "10%"

WHEN sum(p.Price \* od.Quantity) BETWEEN 30000 AND 39999 THEN "5%"

ELSE "0%" END as Comision\_$

FROM Employees e

LEFT JOIN Orders o ON e.EmployeeID = o.EmployeeID

LEFT JOIN OrderDetails od ON o.OrderID = od.OrderID

LEFT JOIN Products p ON od.ProductID = p.ProductID

GROUP BY e.EmployeeID

order by CantidadDeVentas DESC

8) Total de productos vendidoscuyos proveedores son de Japón.La salida debe contener los siguientes campos: Nombre del Proveedor, ciudad y país del proveedor, Contacto del proveedor, Nombre del Producto (tabla Products), Nombre de la categoría del producto (tabla Categories), Cantidad (tabla OrderDetails), Precio (tabla Products) y Total (calculado como Precio\*Cantidad).

SELECT s.SupplierName as NombreProveedor, s.City as CiudadProveedor, s.Country as PaisProveedor, s.ContactName as ContactoProveedor, p.ProductName as NombreProducto, od.Quantity as Cantidad, p.Price as Precio, round(p.Price \* (od.Quantity), 2) as Total

FROM Categories c

INNER JOIN Products p ON p.CategoryID = c.CategoryID

INNER JOIN Customers c ON c.CustomerID = o.CustomerID

INNER JOIN Employees e ON e.EmployeeID = o.EmployeeID

INNER JOIN OrderDetails od ON p.ProductID = od.ProductID

INNER JOIN Orders o ON o.OrderID = od.OrderID

INNER JOIN Suppliers s ON s.SupplierID = p.SupplierID

WHERE s.Country = "Japan"

9) La cantidad de proveedores y de productos, agrupando por cada pais y ciudad de los proveedores. La salida debe contener los siguientes campos:Paísy ciudad de los proveedores (Tabla Suppliers), Cantidad de proveedores, Cantidad de productos.

SELECT s.City as CiudadProveedor, s.Country as PaisProveedor, count(DISTINCT s.SupplierID) as CantSupp, count(DISTINCT p.ProductID) as CantidadProductos

FROM Suppliers s

INNER JOIN Products p ON s.SupplierID = p.SupplierID

GROUP BY PaisProveedor, CiudadProveedor

ORDER BY CantidadProductos DESC

10) Los clientes que gastaronmásde $14.000 en el total de todas sus compras.La salida debe contener los siguientes campos:Nombre del Cliente,ciudad y país del cliente, Cantidad de compras, Total (calculado como la suma total del Precio\*Cantidad).

SELECT U.CustomerName as NombreCliente, U.City as Ciudad, U.Country as Pais, count(DISTINCT O.OrderID) AS Compras,

round(SUM(P.Price\*O.Quantity),2) as Total

FROM Categories C

INNER JOIN Products P ON C.CategoryID = P.CategoryID

INNER JOIN OrderDetails O ON P.ProductID = O.ProductID

INNER JOIN Suppliers S ON P.SupplierID = S.SupplierID

INNER JOIN Orders R ON O.OrderID = R.OrderID

INNER JOIN Employees E ON R.EmployeeID = E.EmployeeID

INNER JOIN Customers U ON R.CustomerID = U.CustomerID

GROUP BY U.CustomerName

HAVING Total > 14000