Ejercicios

Uriel Paluch

19/9/2021

Ejercicios de SQL

https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

Ejercicio 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 “Nombre del vendedor”,
e.LastName as “Apellido del vendedor”,
c.CustomerName as “Nombre del cliente”,
o.OrderID as “ID de la orden”,
p.ProductName as “Nombre del producto”,
p.Unit as “Presentacion”,
od.Quantity as “Cantidad”,
p.Price as “Precio”,
Price * Quantity as “Total”
FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
INNER JOIN Employees e ON o.EmployeeID = e.EmployeeID
INNER JOIN OrderDetails od ON o.OrderID = od.OrderID
INNER JOIN Products p ON p.ProductID = od.ProductID
WHERE “Nombre del vendedor” = “Margaret” AND “Apellido del vendedor” = “Peacock” AND “Nombre del cliente” = “Frankenversand”;

Ejercicio 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.

Algo que menciono Melisa en clase es que la instrucción CASE, una vez que encuentra una coincidencia elimina ese dato del data set, por eso es posible hacerlo así. Sin poner, por ejemplo entre 50 y 75 75 > price > 50, y solo poner solo price > 50.

SELECT ProductName as “Nombre del producto”,
Unit as “Presentacion”,
Price as “Precio Original”,
CASE WHEN Price > 75 THEN “12%” WHEN Price > 50 THEN “8%” WHEN Price >= 30 THEN “4%”

ELSE “0%” END as “Descuento %”,
round(Price * (CASE WHEN Price > 75 THEN 0.12 WHEN Price > 50 THEN 0.8 WHEN Price >= 30 THEN 0.4 ELSE 0 END), 2) as “Descuento $”,
round(Price - Price * (CASE WHEN Price > 75 THEN 0.12 WHEN Price > 50 THEN 0.8 WHEN Price >= 30 THEN 0.4 ELSE 0 END), 2) as “Precio con descuento”
FROM Products

Ejercicio 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 “Nombre de la categoria”,
MIN(p.Price) as “Precio minimo”,
MAX(p.Price) as “Precio maximo”
FROM Products p
INNER JOIN Categories c ON p.CategoryID = c.CategoryID
GROUP BY p.CategoryID
ORDER BY MAX(p.Price) DESC

Ejercicio 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 “Nombre del cliente”,
c.City as “Ciudad”,
c.Country as “Pais”,
o.OrderID as “ID de la orden de compra”,
p.ProductName as “Nombre del producto”,
p.Unit as “Presentacion”,
od.Quantity as “Cantidad”,
p.Price as “Precio”,
p.Price * od.Quantity as “Total”
FROM Customers c
INNER JOIN Orders o ON c.CustomerID = o.CustomerID
INNER JOIN OrderDetails od ON od.OrderID = o.OrderID
INNER JOIN Products p ON p.ProductID = od.ProductID
WHERE c.Country = “Spain”

Ejercicio 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 “Nombre del producto”,
Unit as “Presentacion”,
Price as “Precio”,

```
CASE WHEN Unit LIKE “%box%” THEN “7%”
WHEN Unit LIKE “%bag%” OR Unit LIKE“%bottles%” THEN “5%”
ELSE “2%”
END as “Recargo %”,
round(Price * (CASE WHEN Unit LIKE “%box%” THEN 0.07
WHEN Unit LIKE “%bag%” OR Unit LIKE“%bottles%” THEN 0.05
ELSE 0.02
END),2) as “Recargo $”,
round(Price + Price * (CASE WHEN Unit LIKE “%box%” THEN 0.07
WHEN Unit LIKE “%bag%” OR Unit LIKE“%bottles%” THEN 0.05
ELSE 0.02
END),2) as “Precio con recargo”
FROM Products
```

Ejercicio 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 “Nombre del cliente”,
COUNT(DISTINCT(o.OrderID)) as “Cantidad de pedidos”,
ROUND(SUM(p.Price * od.Quantity), 2) as “Total”
FROM Customers c
INNER JOIN Orders o ON c.CustomerID = o.CustomerID
INNER JOIN OrderDetails od ON o.OrderID = od.OrderID
INNER JOIN Products p on od.ProductID = p.ProductID
WHERE Country = “USA”
GROUP BY o.CustomerID
```

Ejercicio 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(%).

En algunas bases de datos se concatena con “||” como aca, en otras se puede usar la función CONCAT()

```
SELECT
e.FirstName || " " || e.LastName AS “Nombre del vendedor”,
COUNT(e.EmployeeID) AS “Cantidad de ventas”,
ROUND(SUM(p.Price * od.Quantity),2) AS “Total vendido”,
CASE WHEN ROUND(SUM(p.Price * od.Quantity),2) > 100000 THEN “20%”
WHEN ROUND(SUM(p.Price * od.Quantity),2) > 40000 THEN “10%”
WHEN ROUND(SUM(p.Price * od.Quantity),2) > 30000 THEN “5%”END AS “Comisión %”
FROM Orders o
INNER JOIN Employees e ON e.EmployeeID = o.EmployeeID
INNER JOIN OrderDetails od ON od.OrderID = o.OrderID
INNER JOIN Products p ON p.ProductID = od.ProductID
GROUP BY e.EmployeeID
ORDER BY “Total vendido” DESC
```

Ejercicio 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 “Nombre del proveedor”,
s.City AS “Ciudad”,
s.Country AS “Pais”,
s.Phone AS “Contacto”,
p.ProductName AS “Nombre del producto”,
c.CategoryName AS “Categoria del producto”,
SUM(od.Quantity) AS “Cantidad”,
p.Price AS “Precio”,
ROUND(SUM(p.Price * od.Quantity),2) AS “Total”
FROM Suppliers s
INNER JOIN Products p ON p.SupplierID = s.SupplierID
INNER JOIN Categories c ON c.CategoryID = p.CategoryID
INNER JOIN OrderDetails od ON od.ProductID = p.ProductID
WHERE Country = “Japan”
GROUP BY p.ProductID
```

Ejercicio 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.Country AS “Pais”,
s.City AS “Ciudad”,
COUNT(s.City) AS “Cantidad de proveedores”,
COUNT(DISTINCT(p.ProductID)) AS “Cantidad de productos”
FROM Suppliers s
INNER JOIN Products p ON p.SupplierID = s.SupplierID
GROUP BY s.City
ORDER BY s.Country
```

Ejercicio 10:

Los clientes que gastaron más de 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).

Cuando se quiere filtrar en base a una clausula de la consulta se utiliza HAVING

```
SELECT
c.CustomerName AS “Nombre del cliente”,
c.City AS “Ciudad”,
c.Country AS “Pais”,
COUNT(o.CustomerID) AS “Cantidad de compras”,
ROUND(SUM(p.Price * od.Quantity), 2) AS “Total”
FROM Customers c
INNER JOIN Orders o ON o.CustomerID = c.CustomerID
INNER JOIN OrderDetails od ON od.OrderID = o.OrderID
```

```
INNER JOIN Products p ON p.ProductID = od.ProductID
GROUP BY o.CustomerID
HAVING “Total” > 14000
```