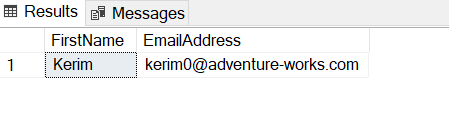
**MSSQL Adventure Works Database**

1. **Show the first name and the email address of customer with CompanyName 'Bike World’**

SELECT DISTINCT FirstName, EmailAddress

FROM SalesLT.Customer

WHERE CompanyName = 'Bike World';



1. **Show the CompanyName for all customers with an address in City 'Dallas'.**

SELECT DISTINCT c.CompanyName

FROM SalesLT.Customer AS c

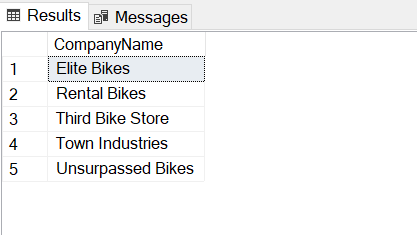
JOIN SalesLT.CustomerAddress AS ca

ON c.CustomerID = ca.CustomerID

JOIN SalesLT.Address AS a

ON ca.AddressID = a.AddressID

WHERE a.City = 'Dallas';

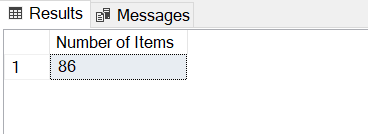


1. **How many items with ListPrice more than $1000 have been sold?**

SELECT COUNT(\*) AS 'Number of Items'

FROM SalesLT.Product

WHERE ListPrice >= 1000;



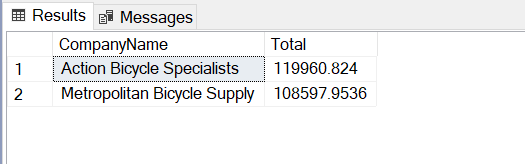
1. **Give the CompanyName of those customers with orders over $100000. Include the subtotal plus tax plus freight.**

SELECT c.CompanyName, oh.SubTotal + oh.TaxAmt + oh.Freight AS 'Total'

FROM SalesLT.Customer AS c

JOIN SalesLT.SalesOrderHeader AS oh ON c.CustomerID = oh.CustomerID

WHERE oh.SubTotal + oh.TaxAmt + oh.Freight > 100000;



1. **Find the number of left racing socks ('Racing Socks, L') ordered by CompanyName 'Riding Cycles'**

SELECT SalesOrderDetail.OrderQty

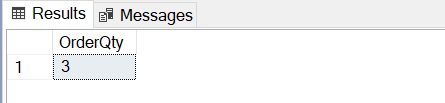
FROM SalesLT.Product

JOIN SalesLT.SalesOrderDetail ON Product.ProductID = SalesOrderDetail.ProductID

JOIN SalesLT.SalesOrderHeader ON SalesOrderDetail.SalesOrderID = SalesOrderHeader.SalesOrderID

JOIN SalesLT.Customer ON SalesOrderHeader.CustomerID = Customer.CustomerID

WHERE Product.Name = 'Racing Socks, L' AND Customer.CompanyName = 'Riding Cycles';

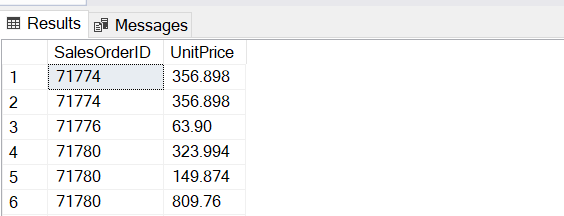


1. **A "Single Item Order" is a customer order where only one item is ordered. Show the SalesOrderID and the UnitPrice for every Single Item Order.**

SELECT sod.SalesOrderID, sod.UnitPrice

FROM SalesLT.SalesOrderDetail AS sod

WHERE sod.OrderQty = 1;



1. **Where did the racing socks go? List the product name and the CompanyName for all Customers who ordered ProductModel 'Racing Socks'.**

SELECT pm.Name AS 'Product Name', c.CompanyName AS 'Company Name'

FROM SalesLT.ProductModel AS pm

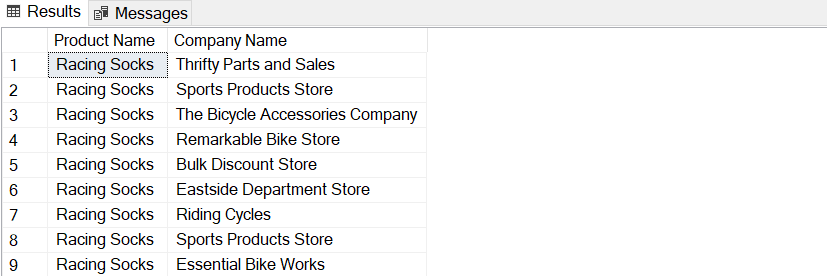
JOIN SalesLT.Product AS p ON pm.ProductModelID = p.ProductModelID

JOIN SalesLT.SalesOrderDetail AS od ON p.ProductID = od.ProductID

JOIN SalesLT.SalesOrderHeader AS oh ON od.SalesOrderID = oh.SalesOrderID

JOIN SalesLT.Customer AS c ON oh.CustomerID = c.CustomerID

WHERE pm.Name = 'Racing Socks';



1. **Show the product description for culture 'fr' for product with ProductID 736.**

SELECT pd.Description

FROM SalesLT.ProductDescription AS pd

JOIN SalesLT.ProductModelProductDescription AS pdpd ON pd.ProductDescriptionID = pdpd.ProductDescriptionID

JOIN SalesLT.ProductModel AS pm ON pdpd.ProductModelID = pm.ProductModelID

JOIN SalesLT.Product AS p ON p.ProductModelID = pm.ProductModelID

WHERE pdpd.Culture = 'fr' AND p.ProductID = 735;



1. **Use the SubTotal value in SaleOrderHeader to list orders from the largest to the smallest. For each order show the CompanyName and the SubTotal and the total weight of the order.**

SELECT c.CompanyName, SUM(oh.subtotal) AS 'Sub Total', SUM(p.weight \* od.OrderQty) AS 'Total Weight'

FROM SalesLT.Customer AS c

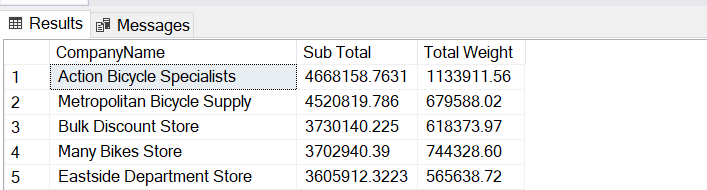
JOIN SalesLT.SalesOrderHeader AS oh ON c.CustomerID = oh.CustomerID

JOIN SalesLT.SalesOrderDetail AS od ON oh.SalesOrderID = od.SalesOrderID

JOIN SalesLT.Product AS p ON od.ProductID = p.ProductID

GROUP BY c.CompanyName

ORDER BY SUM(oh.subtotal) DESC;



1. **How many products in ProductCategory 'Cranksets' have been sold to an address in 'London'?**

SELECT SUM(od.OrderQty) AS 'Order Quantity'

FROM SalesLT.ProductCategory AS pc

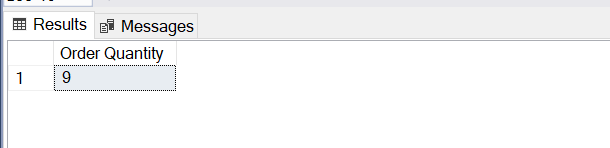
JOIN SalesLT.Product AS p ON pc.ProductCategoryID = p.ProductCategoryID

JOIN SalesLT.SalesOrderDetail AS od ON p.ProductID = od.ProductID

JOIN SalesLT.SalesOrderHeader AS oh ON od.SalesOrderID = oh.SalesOrderID

JOIN SalesLT.Address AS a ON oh.ShipToAddressID = a.AddressID

WHERE pc.Name = 'Cranksets' AND a.City = 'London';



1. **For every customer with a 'Main Office' in Dallas show AddressLine1 of the 'Main Office' and AddressLine1 of the 'Shipping' address - if there is no shipping address leave it blank. Use one row per customer.**

SELECT main.CustomerID, main.MainOffice, shipping.Shipping

FROM

( SELECT ca.CustomerID, a.AddressLine1 AS 'MainOffice'

FROM SalesLT.CustomerAddress AS ca

JOIN SalesLT.Address AS a ON ca.AddressID = a.AddressID

WHERE ca.AddressType = 'Main Office' AND a.City = 'Dallas') AS main

LEFT JOIN

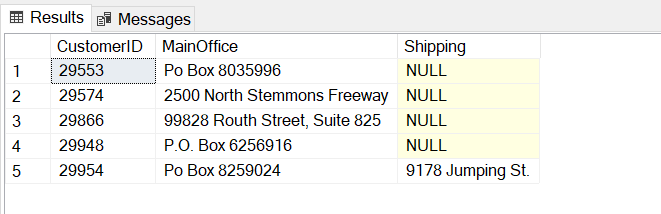
( SELECT ca.CustomerID, a.AddressLine1 AS 'Shipping'

FROM SalesLT.CustomerAddress AS ca

JOIN SalesLT.Address AS a ON ca.AddressID = a.AddressID

WHERE ca.AddressType = 'Shipping') AS shipping

ON main.CustomerID = shipping.CustomerID;



1. **For each order show the SalesOrderID and SubTotal calculated three ways:  
   A) From the SalesOrderHeader  
   B) Sum of OrderQty\*UnitPrice  
   C) Sum of OrderQty\*ListPrice**

SELECT oh.SalesOrderID, oh.SubTotal, SUM(od.OrderQty \* od.UnitPrice) AS 'SubTotalA',

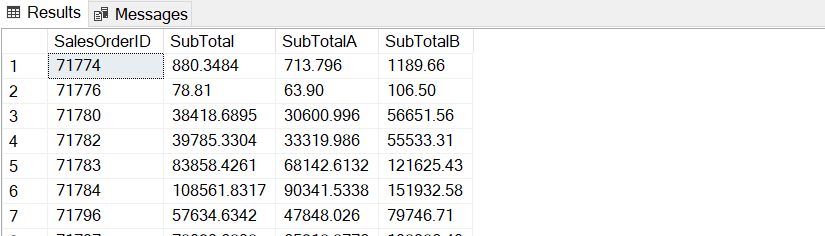
SUM(od.OrderQty \* p.ListPrice) AS 'SubTotalB'

FROM SalesLT.SalesOrderHeader AS oh

JOIN SalesLT.SalesOrderDetail AS od ON oh.SalesOrderID = od.SalesOrderID

JOIN SalesLT.Product AS p ON od.ProductID = p.ProductID

GROUP BY oh.SalesOrderID, oh.SubTotal;



1. **Show the best selling item by value.**
2. **By QtySold**

SELECT p.ProductID, p.Name, od.Qty

FROM SalesLT.Product AS p

JOIN (

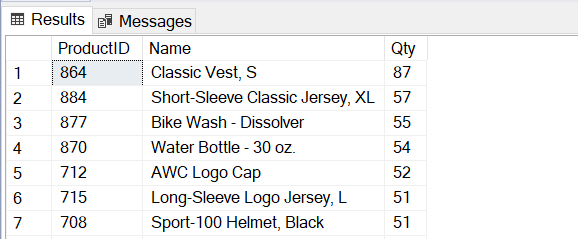
SELECT ProductID, SUM(OrderQty) AS 'Qty'

FROM SalesLT.SalesOrderDetail

GROUP BY ProductID ) AS od

ON p.ProductID = od.ProductID

ORDER BY od.Qty DESC;



1. **By total price**

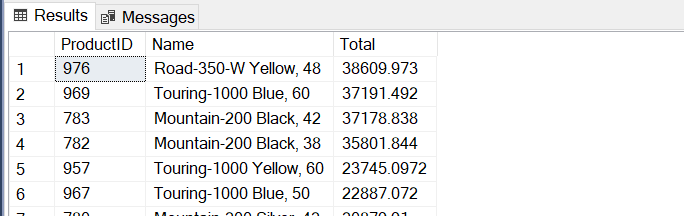
SELECT p.ProductID, p.Name, SUM(od.OrderQty \* (od.UnitPrice - od.UnitPriceDiscount)) AS 'Total'

FROM SalesLT.SalesOrderDetail AS od

JOIN SalesLT.Product AS p ON od.ProductID = p.ProductID

GROUP BY p.ProductID, p.Name

ORDER BY 'Total' DESC;



1. **Show how many orders are in the following ranges (in $):**

RANGE Num Orders Total Value

0- 99

100- 999

1000-9999

10000-

SELECT oh.Range, SUM(oh.TotalDue) AS 'Total', COUNT(oh.SalesOrderId) AS 'Count'

FROM (

SELECT SalesOrderId, TotalDue,

CASE WHEN 0 <= TotalDue AND TotalDue <= 99 THEN '0-99'

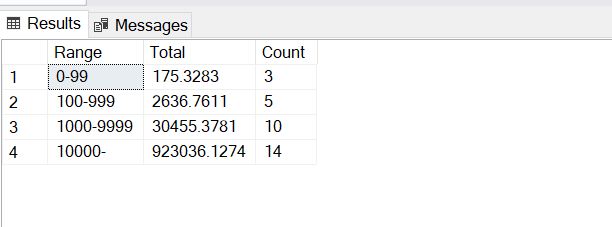
WHEN 100 <= TotalDue AND TotalDue <= 999 THEN '100-999'

WHEN 1000 <= TotalDue AND TotalDue <= 9999 THEN '1000-9999'

ELSE '10000-' END AS 'Range'

FROM SalesLT.SalesOrderHeader ) AS oh

GROUP BY oh.Range;



1. **Identify the three most important cities. Show the break down of top level product category against city.**

Top of Form