Customer(**CustomerID**, FirstName, MiddleName, LastName, CompanyName, EmailAddress)

CustomerAddress(***CustomerID***, *AddressID*, **AddressType**)

Address(**AddressID**, AddressLine1, AddressLine2, City, StateProvince, CountyRegion, PostalCode)

SalesOrderHeader(**SalesOrderID**, RevisionNumber, OrderDate, *CustomerID*, *BillToAddressID*, *ShipToAddressID*, ShipMethod, SubTotal, TaxAmt, Freight)

SalesOrderDetail(*SalesOrderID*, **SalesOrderDetailID**, OrderQty, *ProductID*, UnitPrice, UnitPriceDiscount)

Product(**ProductID**, Name, Color, ListPrice, Size, Weight, *ProductModelID*, *ProductCategoryID*)

ProductModel(**ProductModelID**, Name)

ProductCategory(**ProductCategoryID**, *ParentProductCategoryID*, Name)

ProductModelProductDescription(***ProductModelID***, *ProductDescriptionID*, **Culture**)

ProductDescription(**ProductDescriptionID**, Description)



***The diagram indicates the relationship between different Tables, and the diagram consists by 10 different tables. For each table, it has 1 or 2 columns as combination of keys, and each of them has it’s own PK ( Primary Key ), and some of the has FK ( Foreign Key ). PK is a column to identify unique row in a table, and FK is a column to identify unique row in other table. When we join tables, we usually join two tables based on table 1’s PK and table 2’s FK. In this way, we are able to join two or more tables.***

**1.**

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

select FirstName, EmailAddress, CompanyName

from Customer

WHERE CompanyName = 'Bike World';

*Or use like 'Bike World%'*

**2.**

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

SELECT CompanyName

From Address a

Join CustomerAddress ca on a.AddressID = ca.AddressID

join Customer c on c.CustomerID = ca.CustomerID

WHERE City = 'Dallas';

*1. Do all customers have both Main Office address and/or Shipping address?*

*2. Is there any customer has only Shipping address?*

*After data investigation, it is sure that all customers have Main Office address.*

**3.**

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

SELECT COUNT(distinct p.ProductID) ***As '# of Orders'***, ***Sum(OD.OrderQty) As '# of Items'***

FROM Product p

JOIN SalesOrderDetail od ON p.ProductID = od.ProductID

WHERE p.ListPrice > 1000;2.

*1. DISTINCT product ID and Sum(od.OrderQty) need to show both unique productid and total Qty sold*

*2.*

***SELECT COUNT(1) as '# of items'***

***FROM (SELECT COUNT(DISTINCT sod.productID) as num***

***FROM SalesOrderDetail sod***

***JOIN (Select Product.ProductID***

***From Product***

***Where Product.ListPrice > 1000) p***

***ON sod.ProductID = p.ProductID***

***GROUP BY p.productID) a***

***Compare with mine, instead of join SalesOrderDetail or Procudt table, join only the necessary part of the table will be much more efficient, because the Joined table maybe very large.***

***Remember if need to use join, instead of join the table completely, try to join only the required columns to improve efficiency.***

**4.**

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

SELECT c.CompanyName

From Customer c

JOIN SalesOrderHeader oh on c.CustomerID = oh.CustomerID

WHERE (oh.SubTotal+ oh.TaxAmt+ Freight) > 100000

GROUP BY c.CustomerID, c.CompanyName;

*SELECT SalesOrderID,*

*CONCAT(c.FirstName, ' ', c.LastName) AS CustName,*

*c.CompanyName, soh.SubTotal, soh.TaxAmt, soh.Freight,*

*soh.SubTotal+soh.TaxAmt+soh.Freight AS 'Sale Amount'*

*FROM SalesOrderHeader AS soh*

*LEFT JOIN Customer AS c*

*ON soh.CustomerID = c.CustomerID*

*WHERE soh.SubTotal+soh.TaxAmt+soh.Freight > 100000*

*Usually from bigger table, join smaller table. Better to include more relevant info.*

*Never use select \* from a large table in the work, use where to filter data like time*

**5.**

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

SELECT count(\*)

FROM Customer c

JOIN SalesOrderHeader oh on c.CustomerID = OH.CustomerID

JOIN SalesOrderDetail od on oh.SalesOrderID = od.SalesOrderID

JOIN Product p on od.ProductID = p.ProductID

WHERE P.Name = 'Racing Socks, L' AND c.CompanyName = 'Riding Cycles';

*Select CN.CompanyName, PN.OrderQty*

*From (Select SalesOrderID, OrderQty*

*From SalesOrderDetail As SOD*

*Join Product p*

*On SOD.ProductID = p.ProductID*

*Where Product.Name = 'Racing Socks, L') As PN*

*Join (Select SalesOrderID, SOH.CustomerID, CompanyName*

*From SalesOrderHeader SOH*

*Join Customer c*

*On SOH.CustomerID = c.CustomerID*

*Where CompanyName = 'Riding Cycles') As CN*

*On PN.SalesOrderID = CN.SalesOrderID*

**6.**

**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.**

WITH CTE AS (

SELECT SalesOrderID, SUM(OrderQty) AS TotalOrder

FROM SalesOrderDetail

GROUP BY SalesOrderID

HAVING SUM(OrderQty) = 1)

SELECT od.SalesOrderID, UnitPrice

FROM SalesOrderDetail od

JOIN CTE c on c.SalesOrderID = od.SalesOrderID;

*SELECT sod.SalesOrderID, ProductID, UnitPrice*

*FROM SalesOrderDetail sod*

*JOIN (SELECT SalesOrderID*

*FROM SalesOrderDEtail*

*GROUP BY SalesOrderID*

*HAVING COUNT(ProductID)=1) oo*

*ON sod.SalesOrderID=oo.SalesOrderID*

*only one item is ordered means one productID, not one Qty*

**7.**

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

SELECT DISTINCT p.Name, c.CompanyName

FROM ProductModel pm

JOIN Product p on p.ProductModelID = pm.ProductModelID

JOIN SalesOrderDetail od on od.ProductID = P.ProductID

JOIN SalesOrderHeader oh on od.SalesOrderID = oh.SalesOrderID

JOIN Customer c on oh.CustomerID = c.CustomerID

WHERE pm.name= 'Racing Socks'

# or use group by p.Name, c.CompanyName instead of using distinct

**8.**

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

SELECT pd.Description, pmpd.culture, P.ProductID

FROM ProductDescription pd

JOIN ProductModelProductDescription pmpd on pd.ProductDescriptionID = pmpd.ProductDescriptionID

JOIN ProductModel pm ON pm.ProductModelID = pmpd.ProductModelID

JOIN Product p on p.ProductModelID = pm.ProductModelID

WHERE pmpd.culture = 'fr' AND P.ProductID = 736

*ProductModel Table does not need to be used, since there is a already relationship between ProductModelProductDescription Table and Product Table.*

*Also can use '%fr%', in case of there are spaces before and after fr*

**9.**

**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, oh.SubTotal, SUM(p.Weight) as TotalWeight # wrong SUM(od.OrderQty \*p.Weight)

FROM SalesOrderHeader oh

JOIN Customer c on c.CustomerID = oh.CustomerID

JOIN SalesOrderDetail od on oh.SalesOrderID = od.SalesOrderID

JOIN Product p on p.ProductID = od.ProductID

GROUP BY od.SalesOrderID, c.CompanyName, oh.SubTotal

ORDER BY SubTotal DESC

*Because Use the SubTotal value in SaleOrderHeader, so its better to use from SalesOrderHeader left JOIN others*

*SELECT soh.SalesOrderID, c.CompanyName, soh.SubTotal, sdp.TotalWeight*

*FROM SalesOrderHeader soh*

*JOIN (SELECT SalesOrderID, SUM(sod.OrderQty\*p.Weight) as TotalWeight*

*FROM SalesOrderDetail sod*

*JOIN Product p ON sod.ProductID = p.ProductID*

*GROUP BY SalesOrderID) sdp*

*ON soh.SalesOrderID = sdp.SalesOrderID*

*JOIN Customer c*

*ON soh.CustomerID = c.CustomerID*

*ORDER BY SubTotal DESC*

*Always try to find filter conditions to optimize SQL performance through Subquery. If not exist, then use JOIN.*

*Use* SUM (oh. OrderQty \*p.Weight)

**10.**

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

SELECT COUNT(\*) # wrong sum(OrderQty)

FROM ProductCategory pc

JOIN Product p on p.ProductCategoryID = PC.ProductCategoryID

JOIN SalesOrderDetail od on od.ProductID = P.ProductID

JOIN SalesOrderHeader oh on od.SalesOrderID = oh.SalesOrderID

JOIN Address a on oh.ShipToAddressID = a.AddressID

WHERE pc.name = 'Cranksets' AND a.City = 'London'

*SELECT SUM(DP.salesQTY) totalProducts*

*FROM*

*(SELECT SalesOrderID, SUM(OrderQty) salesQTY*

*FROM SalesOrderDetail sod*

*JOIN Product p ON sod.ProductID = p.ProductID*

*JOIN ProductCategory pc ON p.ProductCategoryID = pc.ProductCategoryID*

*WHERE pc.Name = 'Cranksets'*

*GROUP BY SalesOrderID) DP*

*JOIN (SELECT SalesOrderID*

*FROM SalesOrderHeader soh*

*JOIN Address a ON soh.ShipToAddressID = a.AddressID*

*WHERE a.City = 'London') HA*

*ON DP.SalesOrderID = HA.SalesOrderID*

*GROUP BY DP.SalesOrderID*

*How many products means total Qty not total ProductID. Should use* sum(OrderQty)

**11.**

**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.**

Wrong, should group by company name instead of showing all duplicate results. Every customer more likely means company not customer name

SELECT C.CompanyName, C.EmailAddress,

A1.AddressLine1 AS MainOfficeAddressLine1,

COALESCE(SA.AddressLine1, '') AS ShippingAddressLine1

FROM Customer AS C

LEFT JOIN CustomerAddress AS CA1 ON C.CustomerID = CA1.CustomerID AND CA1.AddressType = 'Main Office'

LEFT JOIN Address AS A1 ON CA1.AddressID = A1.AddressID

LEFT JOIN CustomerAddress AS CA2 ON C.CustomerID = CA2.CustomerID AND CA2.AddressType = 'Shipping'

LEFT JOIN Address AS SA ON CA2.AddressID = SA.AddressID;

***Method 1:***

*Select CompanyName,*

*Max(Main\_Office\_Address) As MAdde,*

*Max(Shipping\_Address) As AhAdd*

*From (*

*Select CompanyName,*

*(Case When CA.AddressType = 'Main Office'*

*Then AD.AddressLine1*

*Else '' End) As 'Main\_Office\_Address',*

*(Case When CA.AddressType = 'Shipping'*

*Then AD.AddressLine1*

*Else'' End) As 'Shipping\_Address'*

*From CustomerAddress As CA*

*Join Address As AD On CA.AddressID = AD.AddressID*

*Join Customer As CT On CA.CustomerID = CT.CustomerID*

*Where City = 'Dallas') BAS*

*Group by CompanyName*

***Method 2:***

*Use the below method, when solve this kind of question, should organise the left part table first which contain City = 'Dallas' And AddressType = 'Main Office’. Then left join the second requirement of AddressType ='Shipping'. Better to structure how the data table looks like and then join each part.*

*Select Mo.CompanyName, Mo.AddressLine1 M\_Ad, Sh.AddressLine1 S\_Ad*

*From (Select CompanyName, AddressLine1*

*From CustomerAddress As CA*

*Join Address As AD On CA.AddressID = AD.AddressID*

*Join Customer As CT On CA.CustomerID = CT.CustomerID*

*Where City = 'Dallas' And AddressType = 'Main Office') Mo*

*Left Outer Join*

*(Select CompanyName, AddressLine1*

*From CustomerAddress As CA*

*Join Address As AD On CA.AddressID = Ad.AddressID*

*Join Customer As CT On CA.CustomerID = CT.CustomerID*

*Where AddressType ='Shipping') Sh*

*On Mo.CompanyName = Sh.CompanyName*

*group by Mo.CompanyName, Mo.AddressLine1, Sh.AddressLine1 better to use group by in case there duplicate records in the data set*

**12.**

**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 (promotion amount)**

SELECT oh.SalesOrderID, oh.SubTotal,

SUM (od.OrderQty \* od.UnitPrice) as SubTotal\_by\_UnitPrice,

SUM (od.OrderQty \* p.ListPrice) as SubTotal\_by\_ListPrice

FROM SalesOrderHeader oh

JOIN SalesOrderDetail od on oh.SalesOrderID = od.SalesOrderID

LEFT JOIN Product p on p.ProductID = od.ProductID

GROUP BY oh.SalesOrderID, oh.SubTotal

**13.**

**Show the best selling item by value.**

1. Order by Profit Margin

SELECT TOP 1

PM.ProductModelID,

PM.Name AS ProductModelName,

SUM(SD.OrderQty \* SD.UnitPrice) AS TotalSalesValue,

(SUM(SD.UnitPrice \* SD.OrderQty) - SUM(P.StandardCost \* SD.OrderQty)) / SUM(SD.UnitPrice \* SD.OrderQty) \* 100 AS ProfitMargin

FROM

SAW. ProductModel PM

JOIN

SAW.Product P ON PM.ProductModelID = P.ProductModelID

LEFT JOIN

SAW.SalesOrderDetail SD ON P.ProductID = SD.ProductID

GROUP BY

PM.ProductModelID, PM.Name

ORDER BY

ProfitMargin DESC;

**2. Order by Total Sales**

SELECT TOP 1

P.ProductID,

P.Name AS 'Best Selling Item',

SUM(SOD.OrderQty \* SOD.UnitPrice) AS ‘Total Sales’

FROM

SAW.Product AS P

INNER JOIN

SAW.SalesOrderDetail AS SOD ON P.ProductID = SOD.ProductID

GROUP BY

P.ProductID, P.Name

ORDER BY

SUM(SOD.OrderQty \* SOD.UnitPrice) DESC;

**Method 1:**

Select Top 1 Name, Sum(OrderQty) As 'Total Qty', Sum(UnitPrice) As 'Total Price', Sum(OrderQty\*UnitPrice) As 'Value'

From SalesOrderDetail SOD

Join Product PD On SOD.ProductID = PD.ProductID

Group By Name

Order By 'Value' Desc

**Method 2:**

Select \*

From (Select ProductID, Sum(OrderQty\*UnitPrice) As 'Value',

RANK() OVER (PARTITION BY 1 ORDER BY Sum(OrderQty\*UnitPrice) DESC) AS Rank

From SalesOrderDetail SOD Group by ProductID) base

Where Rank=1

Window function can only appear in the select or order by clauses. Can’t use in having.

3. Order by Total Volume

SELECT TOP 1

P.ProductID,

P.Name AS BestSellingItem,

SUM(SOD.OrderQty) AS TotalVolumn

FROM Product AS P

JOIN

SalesOrderDetail AS SOD ON P.ProductID = SOD.ProductID

GROUP BY

P.ProductID, P.Name

ORDER BY

TotalVolumn DESC;

14.

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

RANGE Num Orders Total Value

0- 99

100- 999

1000-9999

10000-

SELECT

CASE WHEN SubTotal BETWEEN 0 AND 99 THEN '0- 99'

WHEN SubTotal BETWEEN 100 AND 999 THEN '100- 999'

WHEN SubTotal BETWEEN 1000 AND 9999 THEN '1000- 9999'

WHEN SubTotal >= 10000 then'10000-'

END AS RANGE,

COUNT(\*) AS [Num Orders],

SUM(SubTotal) as 'Total Value'

From SalesOrderHeader

GROUP BY CASE WHEN SubTotal BETWEEN 0 AND 99 THEN '0- 99'

WHEN SubTotal BETWEEN 100 AND 999 THEN '100- 999'

WHEN SubTotal BETWEEN 1000 AND 9999 THEN '1000- 9999'

WHEN SubTotal >= 10000 then'10000-' END

*Select X.Range, sum(1) As '# Of Orders', Sum(X.SubTotal) As 'Total Value'*

*From (Select Case When SubTotal < 100 Then '0-99'*

*When SubTotal Between 100 And 999 Then '100-999'*

*When SubTotal Between 1000 And 9999 Then '1000-9999'*

*When SubTotal > 10000 Then '10000-' Else '' End As 'Range',*

*1 As 'NumOrder', SubTotal*

*From SalesOrderHeader As SOD) As X*

*Group By X.Range*

*1 As 'NumOrder' means count how many orders*

**15.**

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

**TOP 3 by TotalSales**

WITH CTE as ( SELECT a.City, PC.Name AS ProductCategory, PC.ParentProductCategoryID, SUM(SOD.OrderQty \* P.ListPrice) AS TotalSales

FROM Customer AS C

JOIN SalesOrderHeader AS SOH ON C.CustomerID = SOH.CustomerID

JOIN Address a on a.AddressID = soh.ShipToAddressID

JOIN SalesOrderDetail AS SOD ON SOH.SalesOrderID = SOD.SalesOrderID

JOIN Product AS P ON SOD.ProductID = P.ProductID

JOIN ProductCategory AS PC ON P.ProductCategoryID = PC.ProductCategoryID

GROUP BY a.City, PC.Name, PC.ParentProductCategoryID)

SELECT TOP 3

c.City, c.ProductCategory as 'Product Category ', pc.name AS 'top level product category', C.TotalSales as 'Total Sales'

FROM CTE c

JOIN ProductCategory AS PC ON C.ParentProductCategoryID = PC.ProductCategoryID

GROUP BY c.City, c.ProductCategory, pc.name,C.TotalSales

ORDER BY C.TotalSales DESC ;

*Select \**

*From (Select ee.City, dd.ProductCateName, sum(Tot) As Total,*

*rank() over (Partition by ee.city order by sum(tot) desc) as Ranknum*

*From (Select SalesOrderId, cc.Name ProductCateName, Sum(OrderQty\*UnitPrice) Tot*

*From SalesOrderDetail aa*

*Join Product bb*

*On aa.ProductID=bb.ProductID*

*Join ProductCategory cc*

*On bb.ProductCategoryID=cc.ProductCategoryID*

*Group by SalesOrderId, cc.Name ) dd*

*Join*

*(Select bb.City, SalesOrderID, SubTotal*

*From SalesOrderHeader aa*

*Join Address bb On aa.ShipToAddressID=bb.AddressID*

*Join (Select top 3 City From SalesOrderHeader aa*

*Join Address bb On aa.ShipToAddressID = bb.AddressID*

*Group by City*

*Order by sum (SubTotal) desc) CC On bb.City=CC.City) ee*

*On dd.SalesOrderID=ee.SalesOrderID*

*Group by ee.City, dd.ProductCateName) ddd*

Where Ranknum=1

First part subquery is to select product name and total sales. Second part after join is to select top 3 city by its subtotal

**16.**

**List the SalesOrderNumber for the customer 'Good Toys' 'Bike World'**

SELECT c.CompanyName, oh.SalesOrderID

FROM Customer c

LEFT JOIN SalesOrderHeader oh on c.CustomerID = oh.CustomerID

WHERE c.CompanyName in ('Good Toys','Bike World')

GROUP BY c.CompanyName, oh.SalesOrderID

*SELECT SalesOrderID AS SalesOrderNumber*

*FROM SalesOrderHeader*

*WHERE CustomerID IN (SELECT CustomerID FROM Customer*

*WHERE CompanyName IN ('Good Toys', 'Bike World'))*

**17.**

**List the ProductName and the quantity of what was ordered by 'Futuristic Bikes'**

**Method 1:**

SELECT p.Name AS ProductName, sod.OrderQty AS QuantityOrdered

FROM Product p

JOIN SalesOrderDetail sod ON p.ProductID = sod.ProductID

JOIN SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID

JOIN Customer c ON soh.CustomerID = c.CustomerID

WHERE c.CompanyName = 'Futuristic Bikes';

***Method 2:***

*Select Name, Sum(OrderQty) as OrderQty*

*from SalesOrderDetail sd*

*join product pd on sd.Productid=pd.Productid*

*where SalesOrderID in (Select SalesOrderID*

*from SalesOrderHeader SH*

*where CustomerID in (SELECT CustomerID*

*FROM Customer*

*WHERE CompanyName ='Futuristic Bikes'))*

*Group by Name*

**18.**

**List the name and addresses of companies containing the word 'Bike' (upper or lower case) and companies containing 'cycle' (upper or lower case). Ensure that the 'bike's are listed before the 'cycles's.**

SELECT CompanyName, AddressLine1

FROM CustomerAddress ca

JOIN Customer c ON ca.CustomerID = c.CustomerID

JOIN Address a ON ca.AddressID = a.AddressID *AND ca.AddressType = 'Main Office'*

WHERE (UPPER(CompanyName) LIKE '%BIKE%' OR UPPER(CompanyName) LIKE '%CYCLE%')

ORDER BY CASE WHEN UPPER(CompanyName) LIKE '%BIKE%' THEN 1

WHEN UPPER(CompanyName) LIKE '%CYCLE%' THEN 2 END, CompanyName;

*ELECT (CASE WHEN UPPER(CompanyName) LIKE '%BIKE%'*

*THEN 'Bike'*

*ELSE 'Cycle' END) AS Keyword,*

*CompanyName, AddressLine1*

*FROM Customer c*

*JOIN CustomerAddress ca*

*ON c.CustomerID = ca.CustomerID AND ca.AddressType = 'Main Office'*

*JOIN Address a*

*ON ca.AddressID = a.AddressID*

*WHERE UPPER(CompanyName) LIKE '%BIKE%' or UPPER(CompanyName) LIKE '%CYCLE%'*

*ORDER BY Keyword*

*It must have the condition of ‘Main Office’ in either JOIN or WHERE clause; otherwise, there are duplicated values in the result.*

**19.**

**Show the total order value for each CountryRegion. List by value with the highest first.**

1. wrong

WITH CTE AS ( SELECT c.CustomerID, c.CompanyName, a.CountryRegion, s.TotalSales

FROM SAW.Customer c

JOIN (SELECT CustomerID, SUM(SubTotal) AS TotalSales

FROM saw.SalesOrderHeader

GROUP BY CustomerID) s ON c.CustomerID = s.CustomerID

JOIN saw.SalesOrderHeader oh on oh.CustomerID = c.CustomerID

JOIN SAW.Address a on a.AddressID = oh.ShipToAddressID

GROUP BY a.CountryRegion, c.CustomerID, c.CompanyName, s.TotalSales)

SELECT TOP 1

CustomerID, CompanyName, CountryRegion, TotalSales

FROM CTE;

2. correct

with cte as (select a.CountyRegion,

sum(SubTotal) as TotalSales,

DENSE\_RANK() OVER (PARTITION BY a.CountyRegion ORDER BY sum(oh.SubTotal) DESC) AS Ranking

from SalesOrderHeader oh

JOIN Address a on a.AddressID = oh.ShipToAddressID

Group by CountyRegion)

select \*

from cte

where ranking = 1

*Select CountyRegion, Sum(SubTotal) As 'SubTotal'*

*From SalesOrderHeader As SOH*

*Join Address As AD*

*On SOH.ShipToAddressID = AD.AddressID*

*Group By CountyRegion*

*Order By 'SubTotal' Desc*

**20.**

**Find the best customer in each region.**

WITH CTE AS (

SELECT c.CustomerID, c.CompanyName, a.CountryRegion, SUM(oh.SubTotal) AS TotalSales,

RANK() OVER (PARTITION BY a.CountryRegion ORDER BY SUM(oh.SubTotal) DESC) AS CustomerRank

FROM SAW.Customer c

JOIN SAW.SalesOrderHeader oh ON oh.CustomerId = c.CustomerID

JOIN SAW.Address a ON oh.ShipToAddressID = a.AddressID

GROUP BY c.CustomerID, c.CompanyName, a.CountryRegion)

SELECT CustomerID, CompanyName, CountryRegion, TotalSales

FROM CTE

WHERE CustomerRank = 1;

*Select CountyRegion, CompanyName, SubTotal*

*From (Select CountyRegion, CustomerID, Sum(SubTotal) As 'SubTotal',*

*Rank() over (Partition by CountyRegion*

*Order by Sum(SubTotal) Desc) as Rnum*

*From SalesOrderHEader As SOH*

*Join Address As AD*

*On SOH.ShipToAddressID = AD.AddressID*

*Group by CountyRegion, CustomerID) As SBQ*

*Join Customer CU*

*On SBQ.CustomerID = CU.CustomerID*

*Where Rnum = 1*

**Some extra insights**

**Daily Actively Customer Percentage**

SELECT

(COUNT(DISTINCT c.CustomerID) \* 100.0) / (SELECT COUNT(DISTINCT CustomerID) FROM SAW.Customer) AS 'Daily Actively Customer Percentage'

FROM

SAW.Customer c

JOIN

SAW.SalesOrderHeader oh ON c.CustomerID = oh.CustomerID;

**Product profit margin**

SELECT

PM.ProductModelID,

PM.Name AS ProductModelName,

(SUM(SD.UnitPrice \* SD.OrderQty) - SUM(P.StandardCost \* SD.OrderQty)) / SUM(SD.UnitPrice \* SD.OrderQty) \* 100 AS ProfitMargin

FROM SAW. ProductModel PM

JOIN SAW.Product P ON PM.ProductModelID = P.ProductModelID

LEFT JOIN SAW.SalesOrderDetail SD ON P.ProductID = SD.ProductID

GROUP BY PM.ProductModelID, PM.Name

ORDER BY ProfitMargin DESC;

**Order customers by total sales (Customer Lifetime Value (CLV) Analysis OF ONE DAY)**

SELECT

C.CustomerID,

C.FirstName,

C.LastName,

SUM(SH.SubTotal) AS TotalSales

FROM SAW.Customer C

JOIN SAW.SalesOrderHeader SH ON C.CustomerID = SH.CustomerID

GROUP BY C.CustomerID, C.FirstName, C.LastName

ORDER BY TotalSales DESC

**Total Sales Percentage by gender**

SELECT Title,

sum(case when Title = 'Mr.' then 1 else 0 end) / Count(\*) as Male\_Percentage,

sum(case when Title = 'Ms.' then 1 else 0 end) / Count(\*) as Female\_Percentage

FROM SAW.Customer

group by Title;

**TOTAL ORDER COUNT per Customer**

SELECT

C.CustomerID,

C.FirstName,

C.LastName,

COUNT(SH.SalesOrderID) AS OrderCount

FROM SAW.Customer C

JOIN SAW.SalesOrderHeader SH ON C.CustomerID = SH.CustomerID

JOIN SAW.SalesOrderDetail SD ON SH.SalesOrderID = SD.SalesOrderID

JOIN SAW.Product P ON SD.ProductID = P.ProductID

JOIN SAW.ProductCategory PC ON P.ProductCategoryID = PC.ProductCategoryID

GROUP BY c.CustomerID, C.FirstName, C.LastName

ORDER BY OrderCount DESC

OFFSET 0 ROWS

FETCH NEXT 10 ROWS ONLY;

**Geographic Distribution**

SELECT City, StateProvince, CountyRegion, PostalCode,

COUNT(\*) AS CustomerCount

FROM SAW.Address

JOIN SAW.CustomerAddress CA ON Address.AddressID = CA.AddressID

GROUP BY City, StateProvince, CountyRegion, PostalCode

ORDER BY CustomerCount DESC;

**Customer Segmentation by Product Category**

SELECT C.CustomerID, C.FirstName, C.LastName, PC.Name AS ProductCategory,

COUNT(SH.SalesOrderID) AS OrderCount

FROM SAW.Customer C

JOIN SAW.SalesOrderHeader SH ON C.CustomerID = SH.CustomerID

JOIN SAW.SalesOrderDetail SD ON SH.SalesOrderID = SD.SalesOrderID

JOIN SAW.Product P ON SD.ProductID = P.ProductID

JOIN SAW.ProductCategory PC ON P.ProductCategoryID = PC.ProductCategoryID

GROUP BY

C.CustomerID, C.FirstName, C.LastName, PC.Name

ORDER BY

C.CustomerID, OrderCount DESC;

**Salesperson**

SELECT SalesPerson, COUNT(\*) AS CustomerCount

FROM SAW.Customer

GROUP BY SalesPerson

ORDER BY CustomerCount DESC;

**Top Salespersons by Revenue:**

SELECT C.SalesPerson, SUM(SH.SubTotal) AS TotalSales

FROM SAW.Customer C

JOIN SAW.SalesOrderHeader SH ON C.CustomerID = SH.CustomerID

GROUP BY C.SalesPerson

ORDER BY TotalSales DESC;

**Sales Order Insights --- Total revenue of the day**

SELECT SUM(SubTotal) AS TotalRevenue

FROM SAW.SalesOrderHeader

**Total product sold and revenue**

SELECT P.ProductID, P.Name AS ProductName, SUM(SD.OrderQty) AS TotalQuantitySold,

SUM(SubTotal) AS TotalRevenue

FROM SAW.Product P

JOIN SAW.SalesOrderDetail SD ON P.ProductID = SD.ProductID

JOIN SAW.SalesOrderHeader SH ON SD.SalesOrderID = SH.SalesOrderID

GROUP BY P.ProductID, P.Name

ORDER BY TotalQuantitySold DESC;

**View the details of individual sales orders**

SELECT

SH.SalesOrderID,

C.CustomerID,

C.FirstName,

C.LastName,

SD.ProductID,

P.Name AS ProductName,

SD.OrderQty,

SD.UnitPrice,

SD.UnitPriceDiscount,

SD.OrderQty \* SD.UnitPrice AS TotalPrice

FROM

SAW.SalesOrderHeader SH

JOIN

SAW.Customer C ON SH.CustomerID = C.CustomerID

JOIN

SAW.SalesOrderDetail SD ON SH.SalesOrderID = SD.SalesOrderID

JOIN

SAW.Product P ON SD.ProductID = P.ProductID

Order by SH.SalesOrderID

**Product Insights:**

Top-Selling Products: by Qty

SELECT

P.ProductID,

P.Name AS ProductName,

P.Color,

SUM(SD.OrderQty) AS TotalQuantitySold

FROM

SAW.Product P

JOIN

SAW.SalesOrderDetail SD ON P.ProductID = SD.ProductID

GROUP BY

P.ProductID, P.Name, P.Color

ORDER BY

TotalQuantitySold DESC;

**Product Sales Revenue:**

SELECT

P.ProductID,

P.Name AS ProductName,

P.Color,

SUM(SD.OrderQty \* SD.UnitPrice) AS TotalRevenue

FROM

SAW.Product P

JOIN

SAW.SalesOrderDetail SD ON P.ProductID = SD.ProductID

GROUP BY

P.ProductID, P.Name, P.Color

ORDER BY

TotalRevenue DESC;

**Average Order Quantity by Product Category:**

SELECT

PC.ProductCategoryID,

PC.Name AS CategoryName,

AVG(SD.OrderQty) AS AverageOrderQuantity

FROM SAW.Product P

JOIN SAW.SalesOrderDetail SD ON P.ProductID = SD.ProductID

JOIN SAW.ProductCategory PC ON P.ProductCategoryID = PC.ProductCategoryID

GROUP BY PC.ProductCategoryID, PC.Name

ORDER BY AverageOrderQuantity DESC;

**Product Categories and Customer Demand: Total Quantity Sold**

SELECT

PC.ProductCategoryID,

PC.Name AS CategoryName,

SUM(SD.OrderQty) AS TotalQuantitySold

FROM

SAW.Product P

JOIN

SAW.SalesOrderDetail SD ON P.ProductID = SD.ProductID

JOIN

SAW.ProductCategory PC ON P.ProductCategoryID = PC.ProductCategoryID

GROUP BY

PC.ProductCategoryID, PC.Name

ORDER BY

TotalQuantitySold DESC;

**Culture: number of products**

SELECT

pdd.Culture,

COUNT(\*) AS NumberOfProducts,

FROM

SAW.ProductDescription pd

JOIN SAW.ProductModelProductDescription pdd on pd.ProductDescriptionID = pdd.ProductDescriptionID

GROUP BY

pdd.Culture

ORDER BY

NumberOfProducts DESC;

**Order Detail Analysis**

SELECT ProductID, AVG(OrderQty) AS AverageQuantity, AVG(UnitPrice) AS AverageUnitPrice

FROM SAW.SalesOrderDetail

GROUP BY ProductID;

**Product Unit Price Distribution**

SELECT

ProductID,

MIN(UnitPrice) AS MinUnitPrice,

MAX(UnitPrice) AS MaxUnitPrice,

AVG(UnitPrice) AS AvgUnitPrice,

COUNT(\*) AS NumberOfOrders

FROM

SAW.SalesOrderDetail

GROUP BY

ProductID;

**Inventory turnover: ??????**

SELECT

P.ProductID,

P.Name AS ProductName,

SUM(SD.OrderQty \* P.StandardCost) AS CostOfGoodsSold,

SUM(SD.OrderQty \* SD.UnitPrice) AS TotalRevenue,

SUM(SD.OrderQty \* P.StandardCost) / SUM(SD.OrderQty \* SD.UnitPrice) AS InventoryTurnover

FROM

SAW.Product P

JOIN

SAW.SalesOrderDetail SD ON P.ProductID = SD.ProductID

GROUP BY

P.ProductID, P.Name;