Taha Ahmed

--Question 1:

select

top 5

c.CustomerID,c.Name,s.TotalAmount

from

Customer c

inner join SalesOrder s

on s.CustomerID = c.CustomerID

order by s.TotalAmount desc

--Q2:

SELECT

s.SupplierID,s.Name AS SupplierName,

COUNT(DISTINCT p.ProductID) AS ProductCount

FROM

Supplier AS s

JOIN

PurchaseOrder AS po

ON s.SupplierID = po.SupplierID

JOIN

PurchaseOrderDetail AS pod

ON po.OrderID = pod.OrderID

JOIN

Product AS p

ON pod.ProductID = p.ProductID

GROUP BY

s.SupplierID, s.Name

HAVING

COUNT(DISTINCT p.ProductID) > 10

ORDER BY

ProductCount DESC;

--Q3:

SELECT

p.ProductID,

p.Name AS ProductName,

SUM(sod.Quantity) AS TotalOrderedQuantity

FROM product p

JOIN salesorderdetail sod ON p.ProductID = sod.ProductID

WHERE p.ProductID NOT IN (SELECT DISTINCT rd.ProductID

FROM returndetail rd)

GROUP BY

p.ProductID, p.Name;

--Q4

SELECT

c.CategoryID,

c.Name AS CategoryName,

p.Name AS ProductName,

p.Price

FROM

Product AS p

JOIN

Category AS c

ON p.CategoryID = c.CategoryID

WHERE

p.Price = (

SELECT MAX(p2.Price)

FROM Product AS p2

WHERE p2.CategoryID = p.CategoryID

)

ORDER BY

c.CategoryID;

--Q5:

select

so.OrderID, c.Name as Customer\_name, p.Name as Product\_name, ct.Name as Category\_name, s.name as Supplier\_name ,pod.Quantity

from

SalesOrder so

inner join Customer c

on c.CustomerID = so.CustomerID

inner join SalesOrderDetail sd

on sd.OrderID = so.OrderID

inner join Product p

on p.ProductID = sd.ProductID

inner join Category ct

on ct.CategoryID = p.CategoryID

inner join PurchaseOrderDetail pod

on pod.ProductID = p.ProductID

inner join PurchaseOrder po

on po.OrderID = POD.OrderID

INNER JOIN Supplier s

on s.SupplierID = po.SupplierID

--Q6:

SELECT

s.ShipmentID,

s.OrderID,

w.WarehouseID,

w.ContactInfo AS WarehouseContact,

e.Name AS ManagerName,

p.Name AS ProductName,

sd.Quantity,

sd.UnitPrice,

sd.TotalAmount,

s.ShipmentDate,

s.EstimatedArrivalDate,

s.ActualArrivalDate,

s.Status

FROM dbo.Shipment AS s

JOIN dbo.ShipmentDetail AS sd ON s.ShipmentID = sd.ShipmentID

JOIN dbo.Warehouse AS w ON s.WarehouseID = w.WarehouseID

JOIN dbo.Employee AS e ON w.ManagerID = e.EmployeeID

JOIN dbo.Product AS p ON sd.ProductID = p.ProductID;

--Q7:

select \* from(

select

c.CustomerID,

Name ,

OrderID,

TotalAmount,

rank() over (

partition by c.CustomerID

order by so.totalAmount desc

) Highest\_value\_order

from

Customer c

inner join SalesOrder so

on so.CustomerID = c.CustomerID

) t

where Highest\_value\_order <= 3

--Q8:

SELECT

p.ProductID,

p.Name AS ProductName,

sod.OrderID,

so.OrderDate,

sod.Quantity,

LAG(sod.Quantity) OVER (

PARTITION BY p.ProductID

ORDER BY so.OrderDate

) AS PrevQuantity,

LEAD(sod.Quantity) OVER (

PARTITION BY p.ProductID

ORDER BY so.OrderDate

) AS NextQuantity

FROM SalesOrderDetail AS sod

JOIN SalesOrder AS so ON sod.OrderID = so.OrderID

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

ORDER BY p.ProductID, so.OrderDate;

--Q9:

CREATE VIEW vw\_CustomerOrderSummary AS

SELECT

c.CustomerID,

c.Name AS CustomerName,

COUNT(o.OrderID) AS TotalOrders,

SUM(o.TotalAmount) AS TotalAmountSpent,

MAX(o.OrderDate) AS LastOrderDate

FROM Customer AS c

JOIN SalesOrder AS o

ON c.CustomerID = o.CustomerID

GROUP BY

c.CustomerID,

c.Name;

SELECT \* FROM vw\_CustomerOrderSummary;

--Q10:

create procedure sp\_GetSupplierSales(@id as decimal)

as

begin

select

SupplierID,sum(TotalAmount) Total\_Sales

from

PurchaseOrder

where

SupplierID = @id

group by SupplierID

end;

exec sp\_GetSupplierSales 1