**Guidelines**

* Create all your resources in SharePoint folder shared with you over mail.  
  Follow below nomenclature for your file name:  
  *{GDTC\_EMAIL\_WITHOUT\_DOMAIN}\_SQL\_Hands\_On\_{DD\_MM\_YY}.docx*  
  Example:   
  *Your Email*: [abc.def@godigitaltc.com](mailto:abc.def@godigitaltc.com)*File Name*: abc.def\_SQL\_Hands\_On\_10\_06\_25.docx
* Once document is created, copy view link of the file and enter the document link in below mentioned Sheet across your name:  
  {Insert Sheet Link}
* You will be using below Online SQL Editor to perform all your hands-on training  
  [Online SQL Editor](https://www.programiz.com/sql/online-compiler)
* Once your editor is opened, you need to run queries from two scripts in expected order
  + [DDL Script](https://godigitaltcllp.sharepoint.com/:u:/s/GDTC-InternsProcessDocumentation-PanelUpdates/EY1zAjKEuf9LlNut3G6BofsB9gxqSJb-tYiKSwYKYaAQLw?e=x3Q9ZQ): This is a DDL script which creates table needed for your hands-on to be executed first
  + [DML Script](https://godigitaltcllp.sharepoint.com/:u:/s/GDTC-InternsProcessDocumentation-PanelUpdates/EZjxLvRyHGNJnw_ipL3yNocBuxEn3CRh4Mlli4drwOB5yA?e=PVeENh): This is a DML script which creates table needed for your hands-on to be executed second

Tip: Copy all from the script, paste it in editor and click on “Run All”

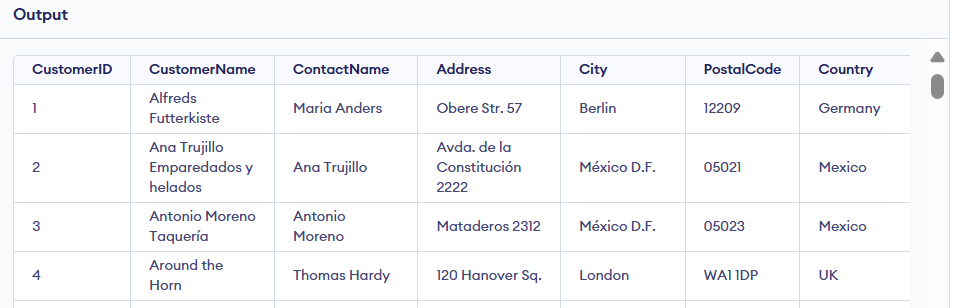
Note: You may need to do this action every time you reload the editor page

* Once data is loaded, you can proceed with solving each question.
* For each question, you are expected to provide the query and the output.

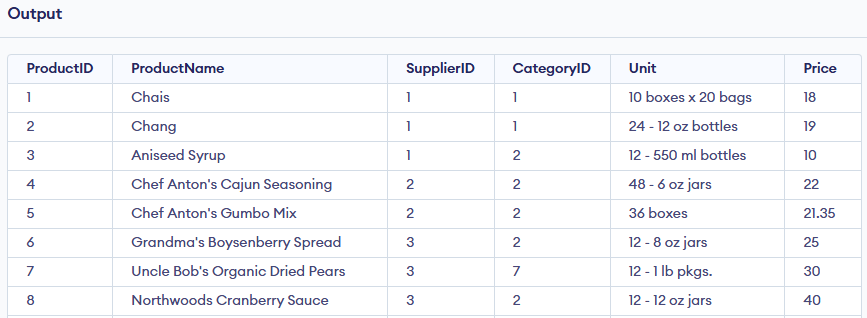
Note: In some questions you may be asked to query for data based on certain data value which may or not match with actual data, you are expected to improvise and use data value as present in our data in that scenario.

**Hands-On Questions**

1. List all customers:

Query: Select \* from customers  
Output:

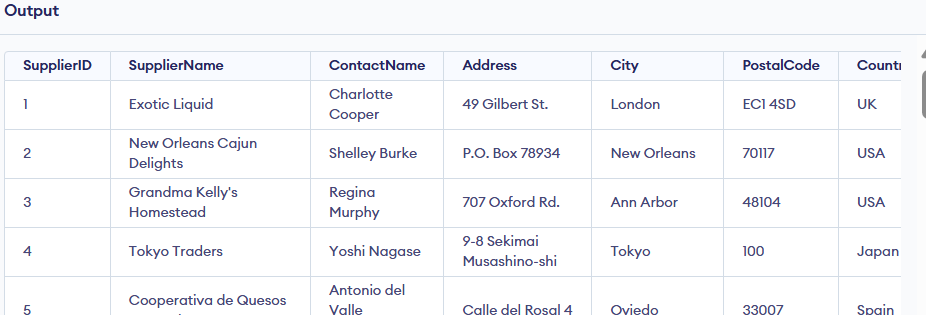
2. List all products:

Query: Select \* from products  
Output:

3. List all orders:

Query: Select \* from orders  
Output:

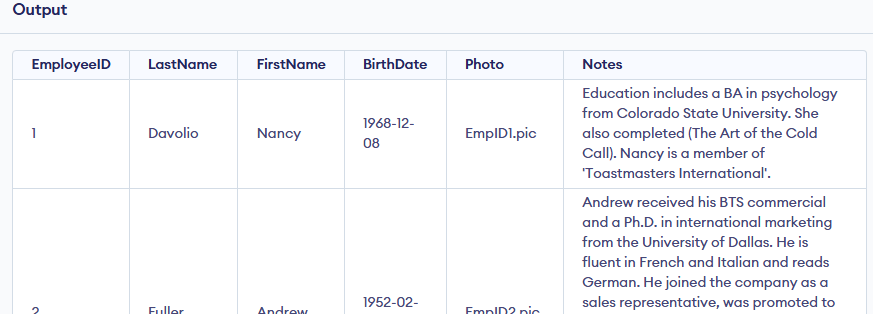
4. List all suppliers:

Query: Select \* from suppliers  
Output:

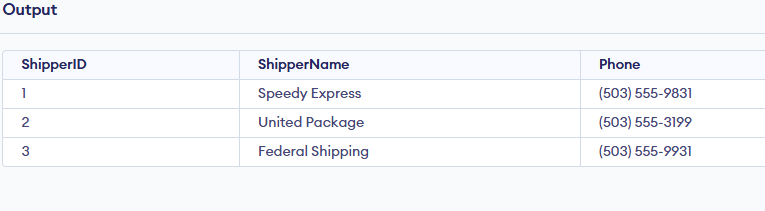
5. List all categories:

Query: Select \* from categories  
Output:

6. List all employees:

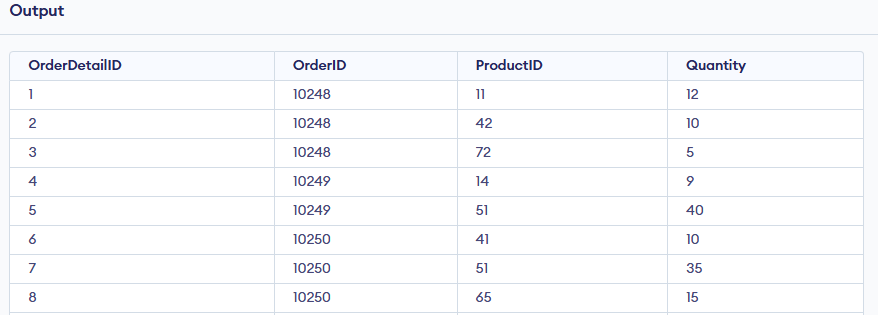
Query: Select \* from employees  
Output:

7. List all shippers:

Query: Select \* from shippers  


8. List all order details:

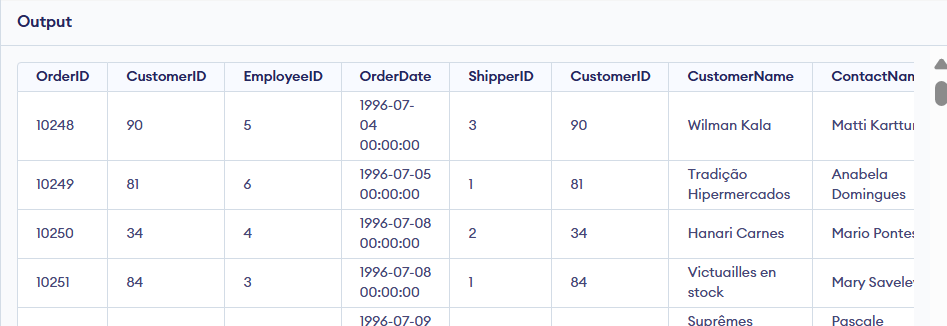
Query: Select \* from orderDetails

Output:

9. List all orders with customer details:

Query: Select \* from orders as o

join customers as c on

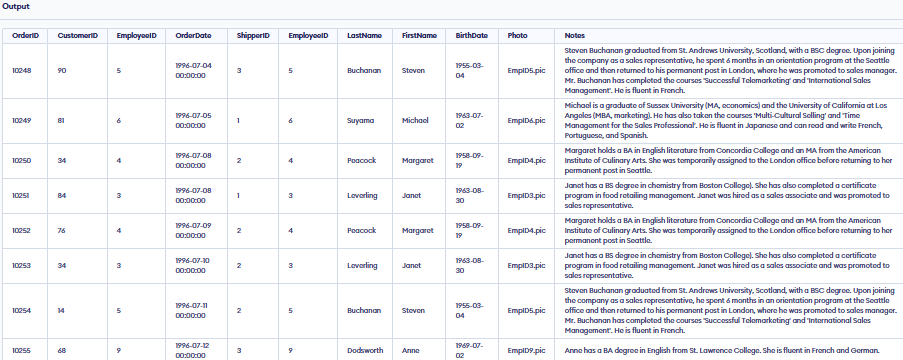
o.CustomerID = c.CustomerID  
Output:

10. List all orders with employee details:

Query:

Select \* from orders as o

join employees as e on

o.EmployeeID = e.EmployeeID  
Output:

11. List all orders with shipper details:

Query:Select \* from orders as o

join Shippers as e on

o.ShipperID = e.ShipperID

Output:



12. List all products along with their supplier and category:

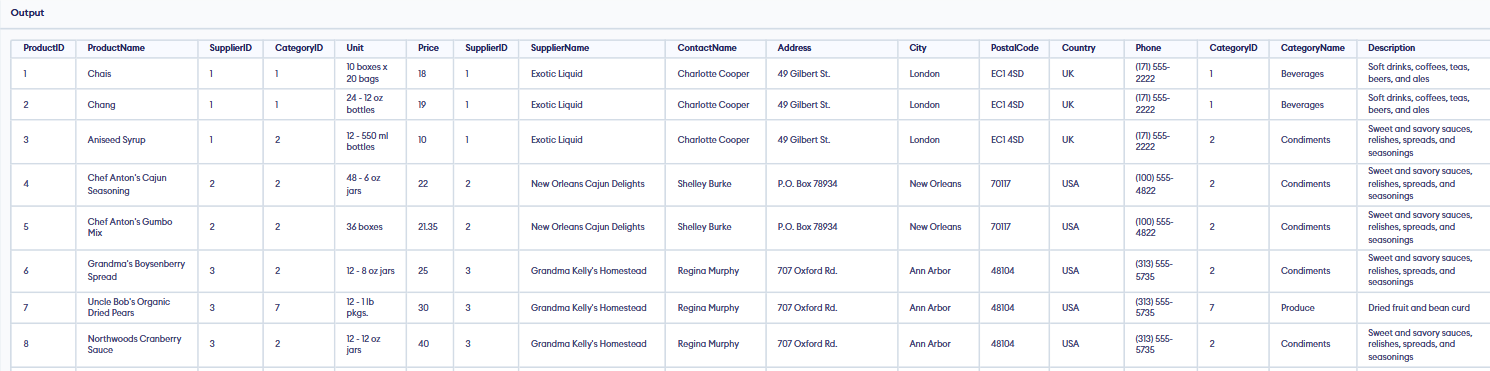
Query:

Select \* from products as p

join Suppliers as s

on p.SupplierID = s.SupplierID

join Categories as c

on p.CategoryID = c.CategoryID  
Output:  


13. List all order details with product and category information:

Query:

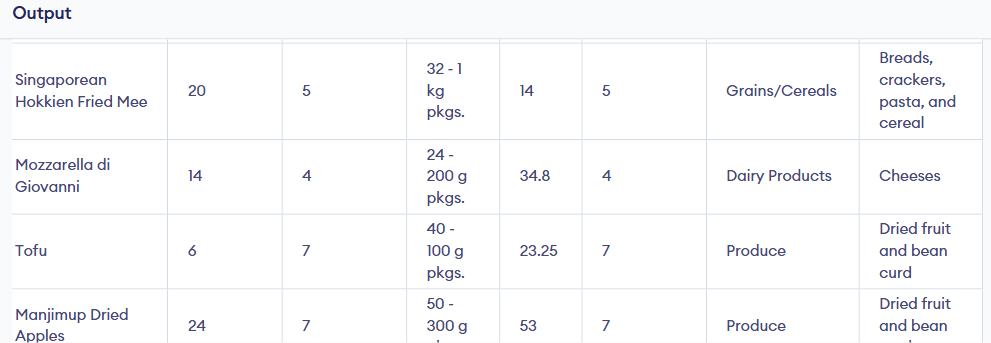
Select \* from OrderDetails as o

join Products as p

on p.ProductID = o.ProductID

join Categories as c

on p.CategoryID = c.CategoryID  
Output:



14. List all customers with their orders and the employees who handled them:

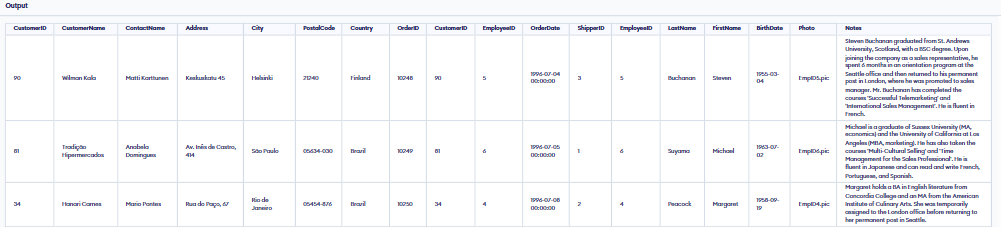
Query:

Select \* from Customers as c

join Orders as o

on c.CustomerID = o.CustomerID

join Employees as e

on o.EmployeeID = e.EmployeeID  
Output:

15. List all orders with product and supplier details:

Query:

Select \* from Orders as o

join OrderDetails as d

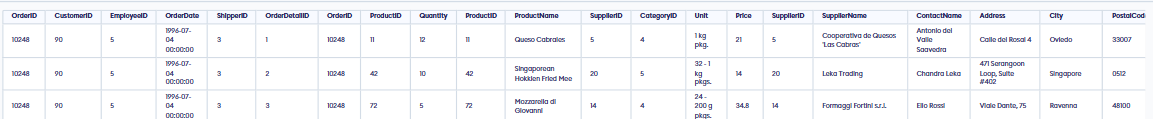
on o.OrderID = d.OrderID

join Products as p

on d.ProductID = p.ProductID

join Suppliers as s

on p.SupplierID = s.SupplierID  
Output:



16. List all orders with customer, employee, and shipper details:

Query:

Select \* from Orders as o

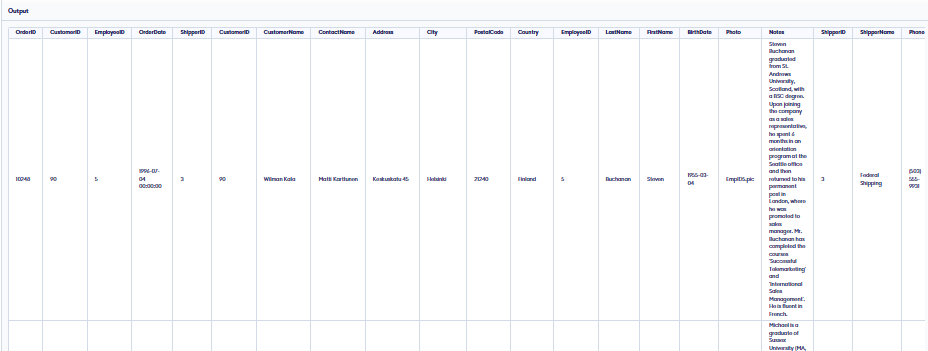
join Customers as d

on o.CustomerID = d.CustomerID

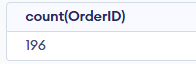
join Employees as e

on o.EmployeeID = e.EmployeeID

join Shippers as s

on o.ShipperID = s.ShipperID  
Output:  


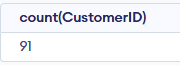
17. Count total number of orders:

Query: Select count(OrderID) from Orders   
Output:

18. Count total number of products:

Query: Select count(ProductID) from Products   
Output:

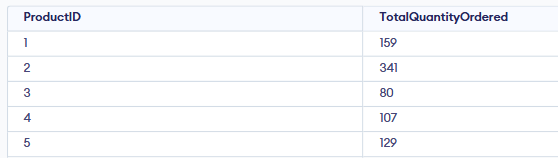
19. Count total number of customers:

Query: Select count(CustomerID) from Customers   
Output:

20. Find the total quantity of each product ordered:

Query: SELECT ProductID, SUM(Quantity) AS TotalQuantityOrdered

FROM OrderDetails

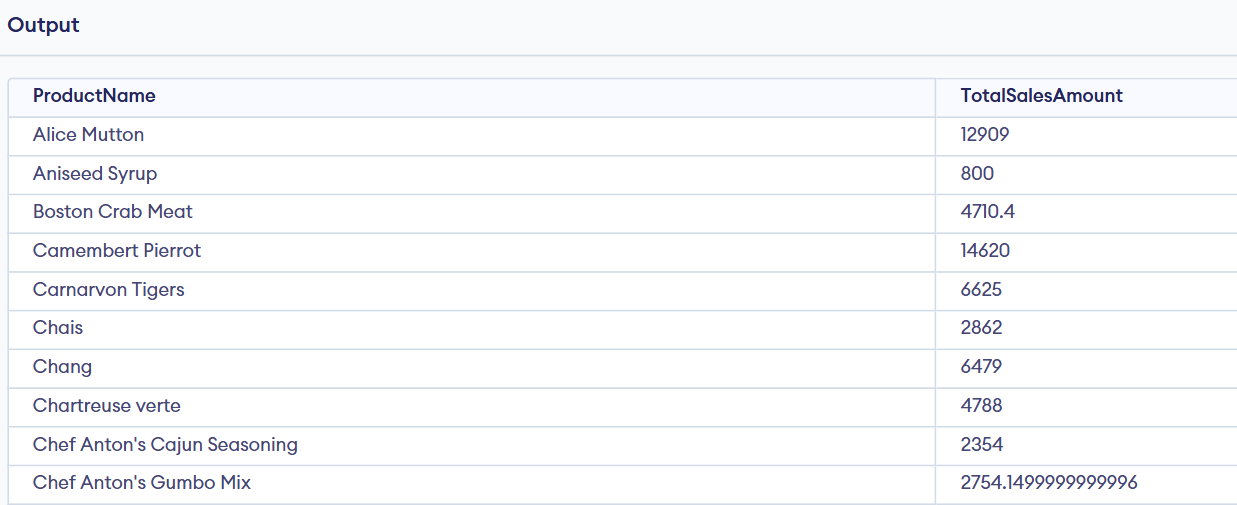
GROUP BY ProductID;  
Output:

21. Find the total sales amount for each product:

Query: SELECT p.ProductName,SUM(od.Quantity \* p.Price) AS TotalSalesAmount

FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY p.ProductName;  
Output:

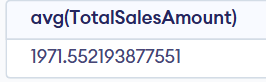
22. Find the average order total:

Query: SELECT avg(TotalSalesAmount)

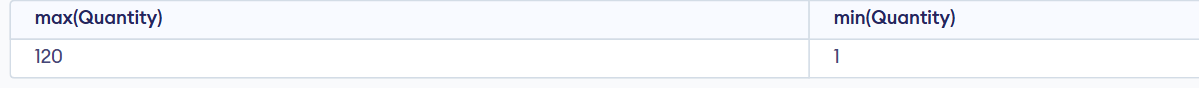
from (select od.orderid, SUM(od.Quantity \* p.Price) AS TotalSalesAmount

FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY od.orderid) as orderTotal ;  
Output:

23. Find the maximum and minimum order quantities:

Query: select max(Quantity), min(Quantity) from OrderDetails  
Output:

24. Find the total revenue generated by each employee:

Query:

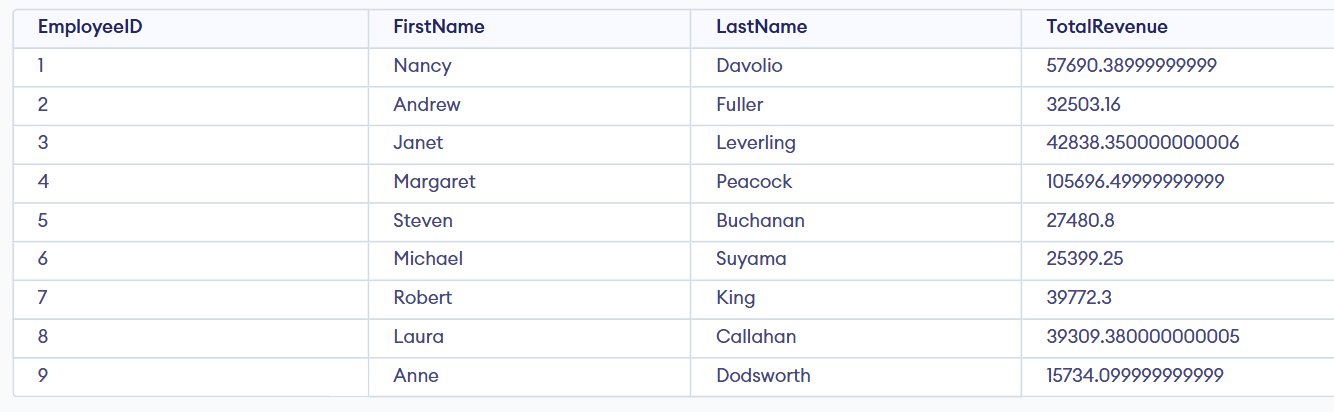
select e.EmployeeID,e.FirstName, e.LastName, sum(od.Quantity\*p.Price) as TotalRevenue

from employees as e

join orders as o on e.EmployeeID = o.EmployeeID

join OrderDetails as od on o.OrderID = od.OrderID

join Products as p on od.ProductID = p.ProductID

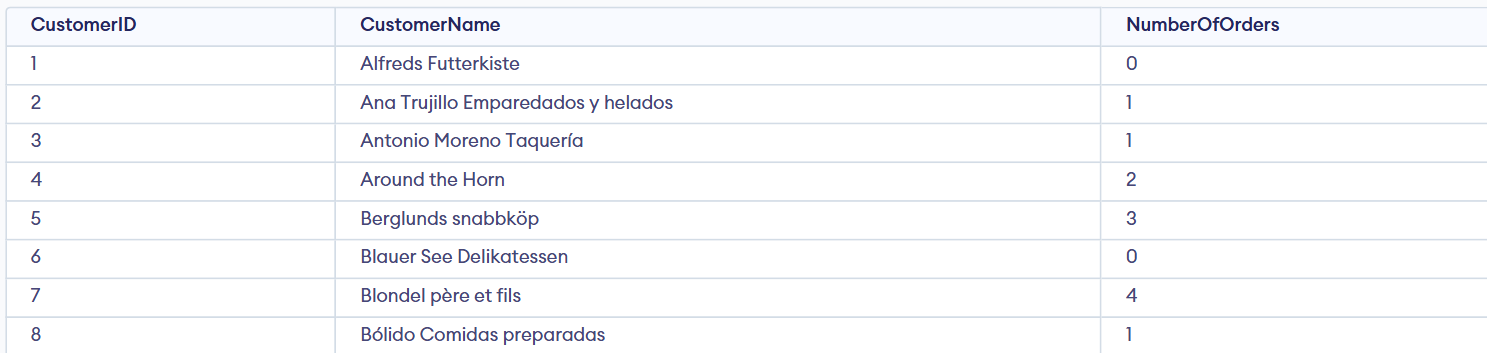
GROUP BY e.EmployeeID, e.FirstName, e.LastName;  
Output:

25. Find the number of orders placed by each customer:

Query: SELECT c.CustomerID, c.CustomerName,COUNT(o.OrderID) AS NumberOfOrders

FROM Customers c

LEFT JOIN Orders o ON c.CustomerID = o.CustomerID

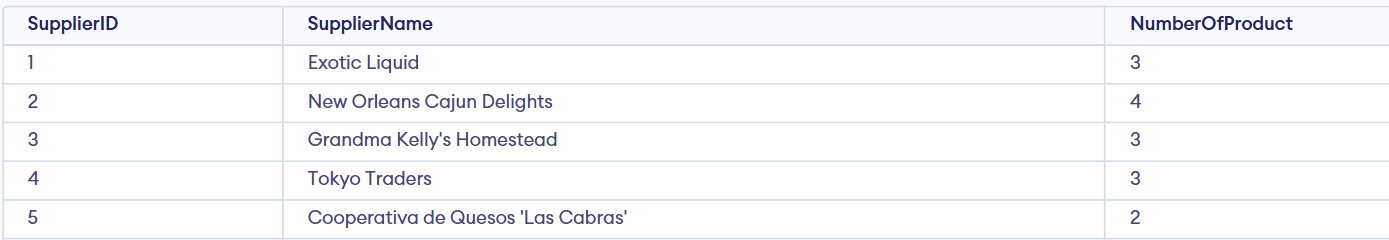
GROUP BY c.CustomerID, c.CustomerName;  
Output:

26. Find the number of products supplied by each supplier:

Query: SELECT s.SupplierID, s.SupplierName,COUNT(p.ProductID) AS NumberOfProduct

FROM Suppliers s

LEFT JOIN Products as p ON s.SupplierID = p.SupplierID

GROUP BY s.SupplierID, s.SupplierName;  
Output:

27. Find the total number of products in each category:

Query: SELECT c.CategoryID, c.CategoryName,COUNT(p.ProductID) AS NumberOfProduct

FROM Categories c

LEFT JOIN Products as p ON c.CategoryID = p.CategoryID

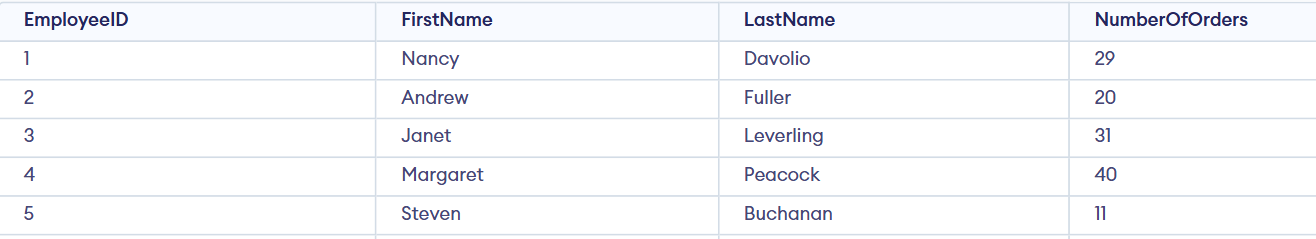
GROUP BY c.CategoryID, c.CategoryName;  
Output:

28. Find the total number of orders handled by each employee:

Query: SELECT e.EmployeeID,e.FirstName, e.LastName,COUNT(o.OrderID) AS NumberOfOrders

FROM Employees e

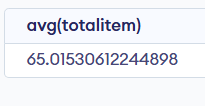
LEFT JOIN Orders as o ON e.EmployeeID = o.EmployeeID

GROUP BY e.EmployeeID,e.FirstName, e.LastName;  
Output:

29. Find the average number of items per order:

Query: select avg(totalitem) from

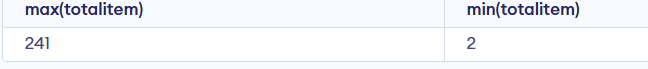
(select OrderID, Sum(Quantity) as totalitem from OrderDetails

group by OrderID)  
Output:

30. Find the highest and lowest order totals:

Query: select max(totalitem),min(totalitem) from

(select OrderID, Sum(Quantity) as totalitem from OrderDetails

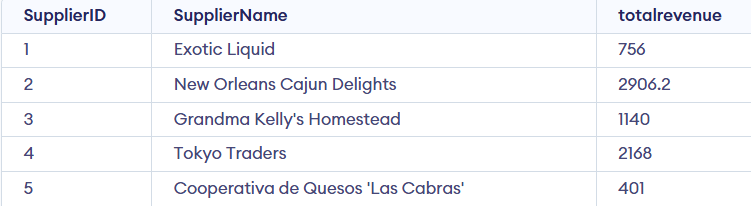
group by OrderID)  
Output:

31. Find the total revenue generated by each supplier:

Query: select p.SupplierID,s.SupplierName , sum(p.Unit\*p.Price) as totalrevenue

from Products as p

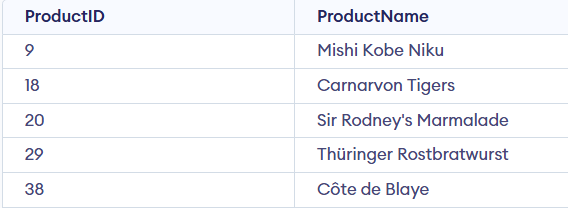
join Suppliers as s on p.SupplierID = s.SupplierID

group by s.SupplierID,s.SupplierName  
Output:

32. List all products with a unit price greater than $50:

Query: select ProductID,ProductName from Products

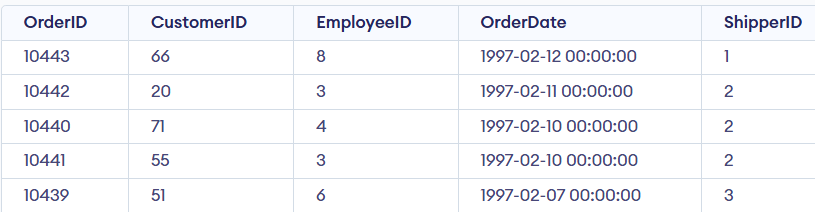
where Price> 50

Output:

33. List all orders placed in the last 30 days:

Query: SELECT \* FROM Orders

order by OrderDate DESC

limit 30;  
Output:  


34. List all customers who have placed more than 5 orders:

Query: SELECT c.CustomerID,c.CustomerName, od.Quantity

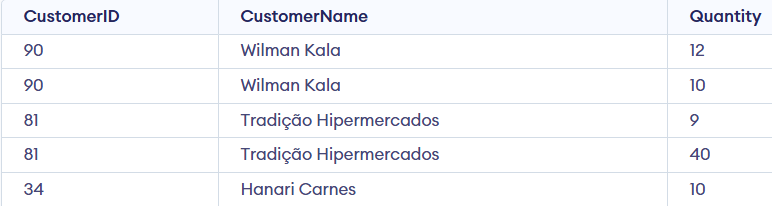
FROM Customers as c

join Orders as o on c.CustomerID = o.CustomerID

join OrderDetails as od on o.OrderID = od.OrderID

where od.Quantity>5

Output:



35. List all employees who have handled orders worth more than $10,000:

Query:

select e.EmployeeID,e.FirstName|| ' ' || e.LastName as name, sum(od.Quantity \* p.Price) as totalamount

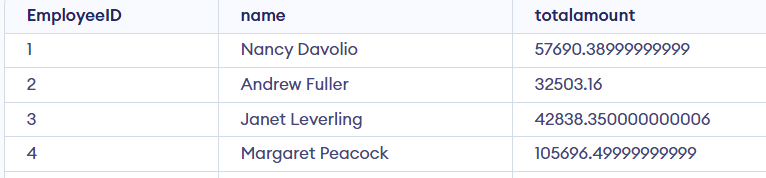
from Employees as e

join orders as o on e.EmployeeID = o.EmployeeID

join OrderDetails as od on o.OrderID = od.OrderID

join Products as p on od.ProductID = p.ProductID

group by e.EmployeeID, name  
having totalamount>10000;  
Output:



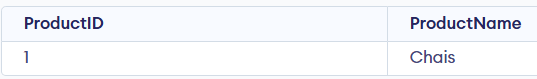
36. List all products supplied by 'Supplier A':

Query: select ProductID,ProductName from Products as p

join Suppliers as s on p.SupplierID =s.SupplierID

group by p.SupplierID

having s.SupplierID = 1  
Output:



37. List all orders shipped by 'Shipper B':

Query:  
select \* from Orders as o

join Shippers as s on o.ShipperID = s.ShipperID

group by s.ShipperID

having s.ShipperID = 2  
Output:



38. List all orders placed by 'Customer C':

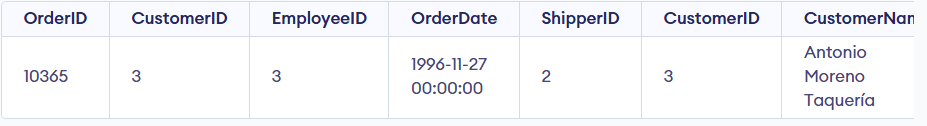
Query:

select \* from Orders as o

join Customers as c on o.CustomerID = c.CustomerID

group by c.CustomerID

having c.CustomerID = 3  
Output:



39. List all products in the 'Electronics' category:

Query: select \* from Products as p

join Categories as c on p.CategoryID = c.CategoryID

group by c.CategoryName

having c.CategoryName = "Electronics"  
Output:

40. List all employees who have not handled any orders:

Query: select \* from Employees as e

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

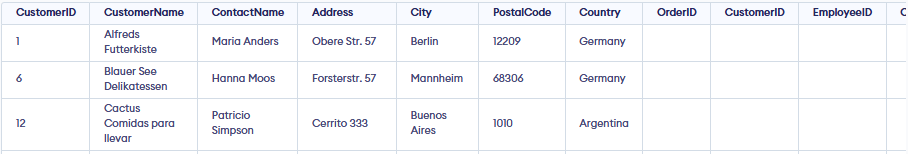
WHERE o.OrderID IS NULL;  
Output:



41. List all customers who have not placed any orders:

Query: select \* from Customers as c

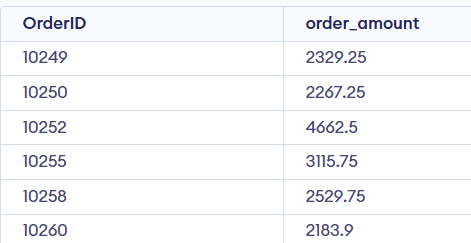
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID

WHERE o.OrderID IS NULL;  
Output:

42. List all orders where the total order amount is greater than the average order amount:

Query:

with total\_order\_amount as ( select od.OrderID, sum(od.Quantity\*p.Price) as order\_amount from OrderDetails as od join Products as p on od.ProductID = p.ProductID group by od.orderID)

select \* from total\_order\_amount where order\_amount>(select avg(order\_amount) from total\_order\_amount)  
Output:

43. Find the product with the highest sales amount:

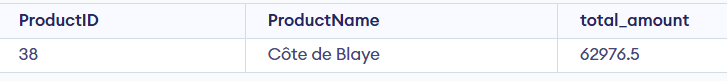
Query: select p.ProductID,p.ProductName, sum(od.Quantity \* p.Price) as total\_amount

from Products as p

join OrderDetails as od on p.ProductID = od.ProductID

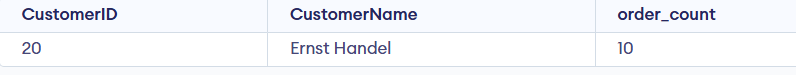
group by p.ProductID,p.ProductName

order by total\_amount desc

limit 1  
Output:

44. Find the customers who have placed the highest number of orders:

Query: with customer\_order as (select c.CustomerID, c.CustomerName, count(o.OrderID) as order\_count from Customers as c join Orders as o on c.CustomerID = o.CustomerID group by c.CustomerID, c.CustomerName )

select \* from customer\_order where order\_count = (select Max(order\_count) from customer\_order)  
Output:

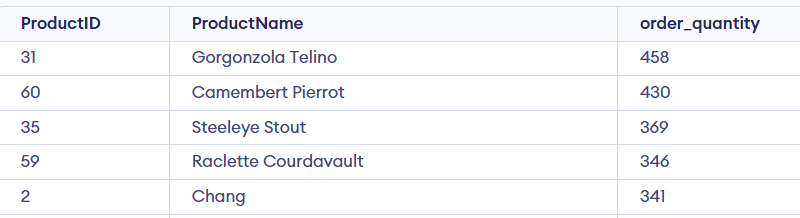
45. List the most popular products based on order quantity:

Query: select p.ProductID,p.ProductName, sum(od.quantity) as order\_quantity from Products as p

join OrderDetails as od on p.ProductID = od.ProductID

group by p.ProductID,p.ProductName

order by order\_quantity desc

Output:

46. List all customers who have placed orders worth more than $5000:

Query:

select c.CustomerID, c.CustomerName, sum(od.quantity\*p.price) as amount from Customers as c join Orders as o on c.CustomerID = o.CustomerID join OrderDetails as od on o.OrderID=od.OrderID join Products as p on od.ProductID = p.ProductID group by c.CustomerID, c.CustomerName having amount > 5000  
Output:

47. Find the top 5 most expensive products:

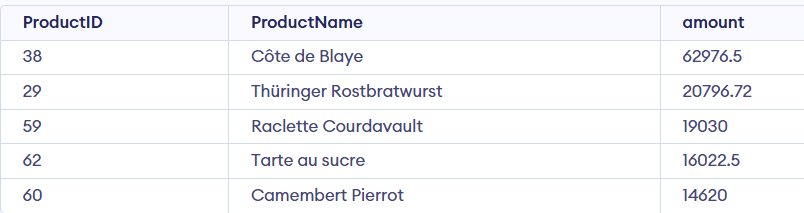
Query: select p.ProductID, p.ProductName, sum(od.quantity\*p.price) as amount

from products as p

join OrderDetails as od on p.ProductID = od.ProductID

group by p.ProductID, p.ProductName

order by amount desc

limit 5  
Output:

48. List all products that have never been ordered:

Query: select p.ProductID,p.ProductName from Products as p

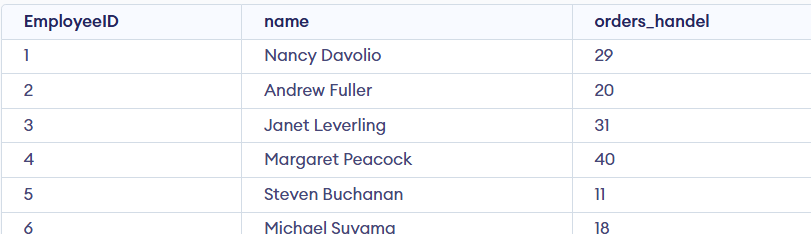
left join OrderDetails as od on p.ProductID = od.ProductID

group by p.ProductID,p.ProductName

having od.OrderID is null  
Output:49. List all employees along with the number of orders they have handled:

Query: select e.EmployeeID, e.FirstName || ' ' || e.LastName as name , count(o.OrderID) as orders\_handel from Employees as e

left join Orders as o on e.EmployeeID=o.EmployeeID

group by e.EmployeeID, name  
Output:

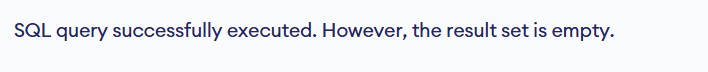
50. List all suppliers who supply products in the 'Electronics' category:

Query: select s.SupplierID, s.SupplierName from Suppliers as s

left join Products as p on s.SupplierID = p.SupplierID

left join Categories as c on p.CategoryID = c.CategoryID

group by s.SupplierID, s.SupplierName

having CategoryName="Electronics"  
Output:

51. Find the employees who have handled the fewest orders:

Query:

with orderCount as

(select e.EmployeeID, e.FirstName || ' ' || e.LastName as name,count(o.OrderID) as totalOrder from employees as e

left join Orders as o on e.EmployeeID = o.EmployeeID

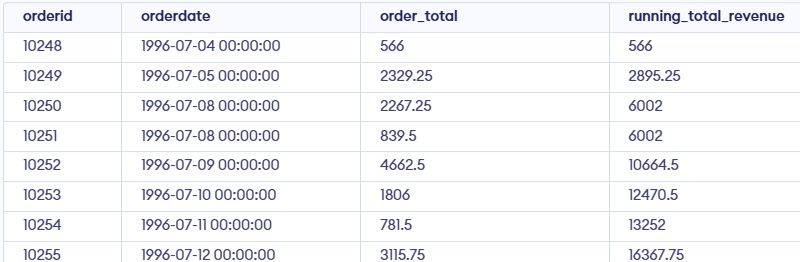
group by e.EmployeeID, name)

Select \* from orderCount

where totalOrder = (select min(totalOrder) from orderCount)  
Output:

52. Calculate the running total of orders over time:

Query: with order\_totals as ( select o.orderid,o.orderdate, sum(od.quantity \* p.price) as order\_total from orders o join orderdetails od on o.orderid = od.orderid join products p on od.productid = p.productid group by o.orderid )

select orderid,orderdate,order\_total, sum(order\_total) over (order by orderdate) as running\_total\_revenue from order\_totals order by orderdate;  
Output:

53. Rank customers by the total amount spent:

Query: select c.CustomerID, c.CustomerName, sum(od.quantity*p.price) as amountSpent, rank () over(order by sum(od.quantity*p.price) desc) as rank

from Customers as c join Orders as o on c.CustomerID = o.CustomerID

join OrderDetails as od on o.OrderID = od.OrderID

join Products as p on od.ProductID = p.ProductID

group by c.CustomerID  
Output:



54. Calculate the difference in order totals between consecutive orders:

Query:

with orderTotal as (

select o.OrderID, sum(od.Quantity\*p.price) as TotalAmount from Orders o

join OrderDetails od on o.OrderID = od.OrderID

join Products p on od.ProductID = p.ProductID

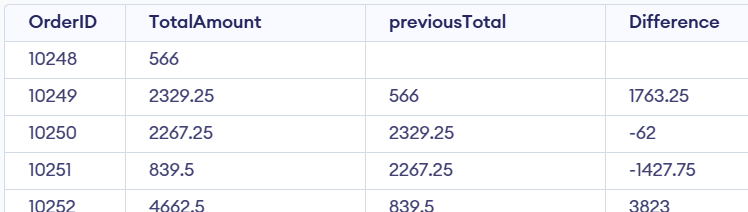
group by o.OrderID)

select OrderID, TotalAmount,

LAG(TotalAmount) over (order by ot.OrderID) as previousTotal,

TotalAmount - LAG(TotalAmount) over (order by ot.OrderID) as Difference

from orderTotal as ot  
Output:



55. Calculate the cumulative total for each customer:

Query:

with cte as (select c.CustomerID, c.CustomerName,o.OrderID,o.OrderDate,

SUM(od.Quantity \* p.Price) AS OrderTotal

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

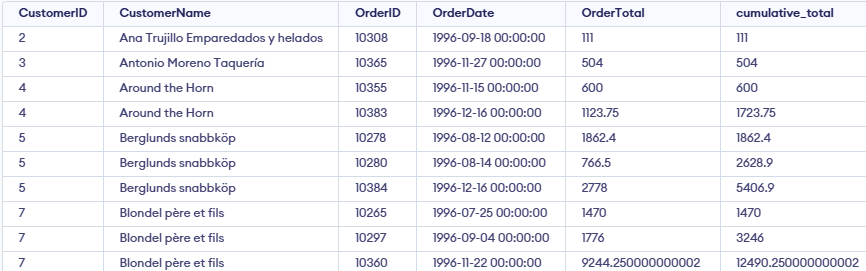
JOIN OrderDetails od ON o.OrderID = od.OrderID

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY c.CustomerID, c.CustomerName, o.OrderID, o.OrderDate)

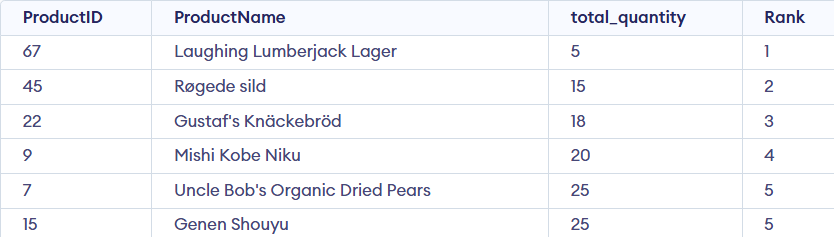
select CustomerID, CustomerName,OrderID,OrderDate, OrderTotal,

sum(OrderTotal) over (partition by CustomerID order by OrderDate) as cumulative\_total

from cte  
Output:

56. Rank products by total quantity ordered:

Query: select p.ProductID, p.ProductName , sum(od.quantity) as total\_quantity, rank() over(order by sum(od.quantity) ) as Rank from Products as p join OrderDetails as od on p.ProductID = od.ProductID group by p.ProductID, p.ProductName

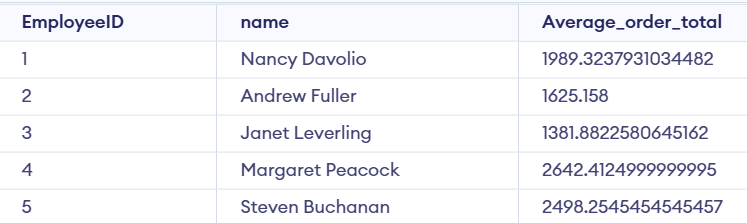
Output:

57. Calculate the average order total by employee:

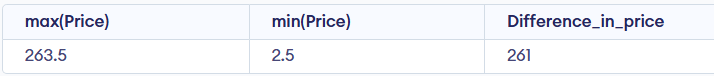
Query:

with Total as (select e.EmployeeID,o.orderid, sum(od.Quantity\*p.price) as TotalAmount from Employees e join Orders o on e.EmployeeID=o.EmployeeID join OrderDetails od on o.OrderID=od.OrderID join Products p on od.ProductID = p.ProductID group by e.EmployeeID,o.orderid)

select e.EmployeeID, e.FirstName || ' ' || e.LastName as name, avg(TotalAmount) as Average\_order\_total from Employees e join Total t on e.EmployeeID = t.EmployeeID group by e.EmployeeID, e.FirstName,e.LastName

Output:

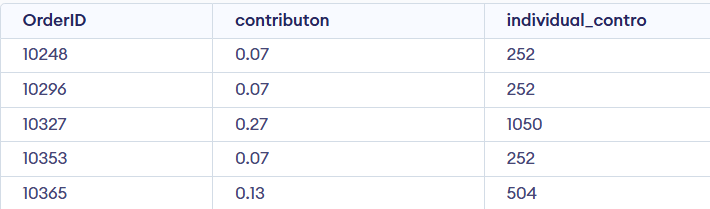
58. Find the difference between the highest and lowest product prices:

Query: select max(Price),min(Price), max(Price)-min(Price) as Difference\_in\_price from Products  
Output:

59. Calculate the percentage contribution of each order to the total revenue:

Query: with totalrevenue as ( select od.ProductID, sum(od.Quantity\*p.Price) as totalamount from orders o join OrderDetails od on o.OrderID=od.OrderID join Products p on od.ProductID = p.ProductID )

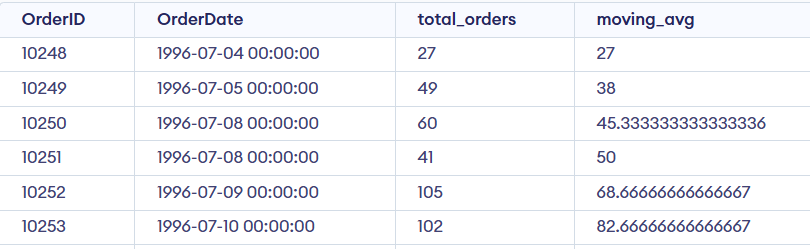
select od.OrderID,round((sum(od.Quantity\*p.Price)\**100)/t.totalamount,2) as contributon,sum(od.Quantity*p.Price) from totalrevenue as t join OrderDetails od on t.ProductID= od.ProductID join Products p on od.ProductID = p.ProductID group by od.OrderID

Output:

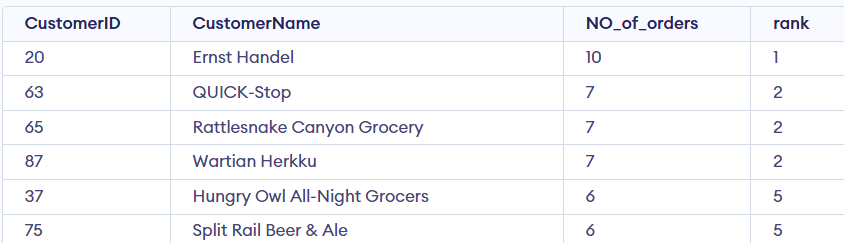
60. Calculate the moving average of order totals over a 3-order window:

Query: with orderTotal as (select od.orderid,sum(od.Quantity) as total\_orders from orderdetails od

join Products p on od.ProductID = p.ProductID group by od.orderid)

select o.orderid,o.OrderDate,ot.total\_orders,avg(ot.total\_orders) over (order by o.orderdate rows between 2 preceding and current row) as moving\_avg  
 from orderTotal ot join orders o on ot.orderid = o.orderid order by ot.orderid  
  
Output:

61. Rank customers by the number of orders they have placed:

Query: select c.CustomerID, c.CustomerName, count(o.OrderID) as NO\_of\_orders, rank() over (order by count(o.OrderID) desc) as rank from Customers c join Orders o on c.CustomerID = o.CustomerID group by c.CustomerID, c.CustomerName order by rank;  
Output:

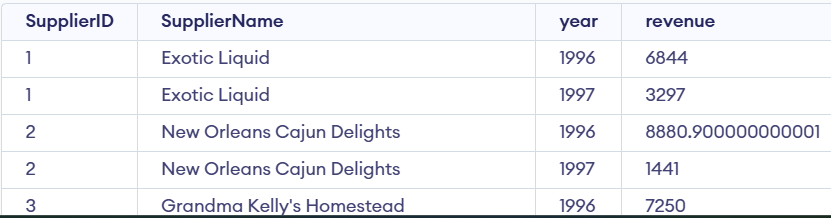
62. Calculate the total revenue for each supplier by year:

Query: select s.SupplierID,s.SupplierName,strftime("%Y",o.OrderDate) as year, sum(od.Quantity\*p.price) as revenue from Suppliers s

join Products p ON s.SupplierID = p.SupplierID

join OrderDetails od ON p.ProductID = od.ProductID

join Orders o ON od.OrderID = o.OrderID

group by s.SupplierID, s.SupplierName, strftime('%Y', o.OrderDate)  
Output:

63. List the average order amount by employee:

Query:

with orderamount as (select e.EmployeeID, e.FirstName || ' ' || e.LastName as name,sum(od.Quantity\*p.price) as Order\_amount

from Employees e

join Orders o on e.EmployeeID = o.EmployeeID

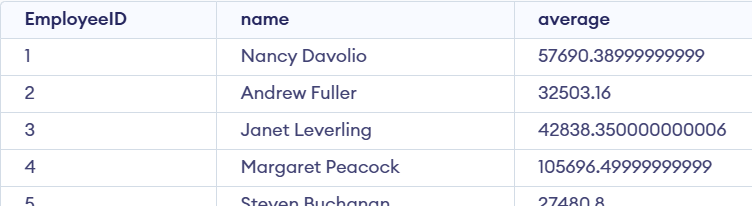
join OrderDetails od on o.OrderID = od.OrderID

join Products p on od.ProductID = p.ProductID

group by e.EmployeeID, e.FirstName,e.LastName)

select EmployeeID, name, avg(Order\_amount) as average

from orderamount

group by EmployeeID, name  
Output:

64. List the top 3 categories by sales amount:

Query: select c.categoryid,c.categoryname,

sum(od.quantity \* p.price) as totalsales

from categories c

join products p on c.categoryid = p.categoryid

join orderdetails od on p.productid = od.productid

group by c.categoryid, c.categoryname

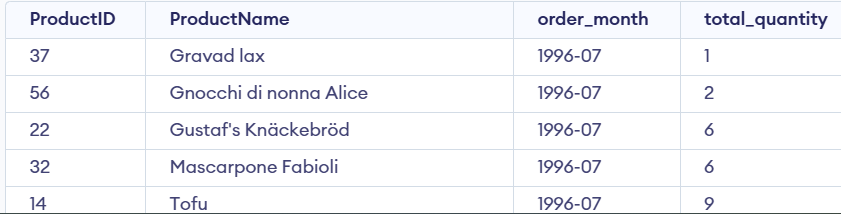
order by totalsales desc

limit 3;

Output:

65. Calculate the total quantity ordered for each product by month:

Query:

select p.productid,p.productname, strftime('%Y-%m', o.orderdate) as order\_month, sum(od.quantity) as total\_quantity from products p join orderdetails od on p.productid = od.productid join orders o on od.orderid = o.orderid group by p.productid, p.productname, order\_month order by order\_month, total\_quantity   
Output:

66. List the total revenue generated by each product in the last 6 months:

Query:

select p.productid,p.productname,

sum(od.quantity \* p.price) as total\_revenue from products p

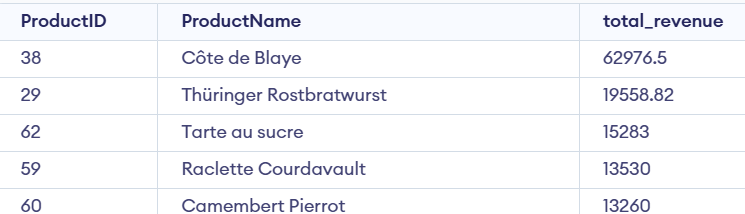
join orderdetails od on p.productid = od.productid

join orders o on od.orderid = o.orderid

where date(o.orderdate) >= "1996-08"

group by p.productid, p.productname

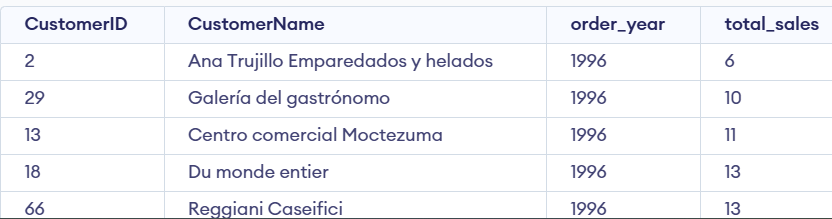
order by total\_revenue desc;

Output:

67. Calculate the total sales for each customer by year:

Query:

select c.customerid,c.customername, strftime('%Y', o.orderdate) as order\_year, sum(od.quantity) as total\_sales from customers c join orders o on c.customerid = o.customerid join orderdetails od on o.orderid = od.orderid join products p on od.productid = p.productid group by c.customerid, c.customername, order\_year order by order\_year, total\_sales ;  
Output:



68. List the average quantity ordered per product:

Query:

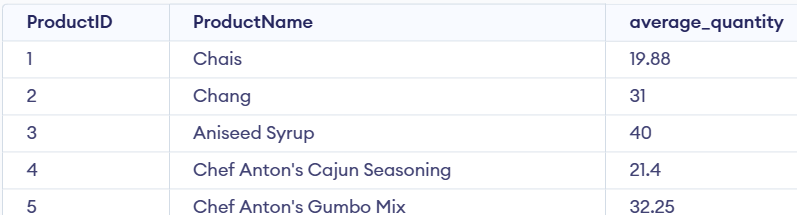
select p.productid,p.productname,

round(avg(od.quantity),2) as average\_quantity

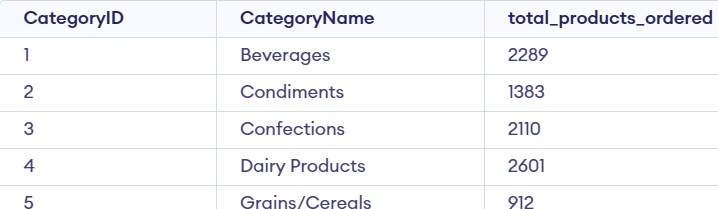
from products p

join orderdetails od on p.productid = od.productid

group by p.productid, p.productname

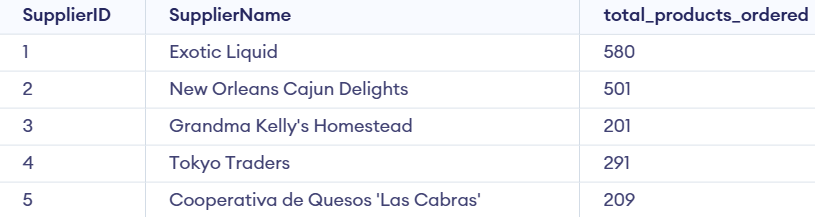
order by p.productid;  
Output:

69. Calculate the total number of products ordered per category:

Query: select c.categoryid,c.categoryname, sum(od.quantity) as total\_products\_ordered from categories c join products p on c.categoryid = p.categoryid join orderdetails od on p.productid = od.productid group by c.categoryid, c.categoryname order by c.categoryid;  
Output:

70. List the number of products ordered per supplier:

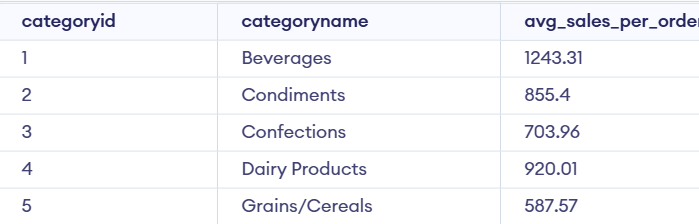
Query: select s.supplierid,s.suppliername, sum(od.quantity) as total\_products\_ordered from suppliers s join products p on s.supplierid = p.supplierid join orderdetails od on p.productid = od.productid group by s.supplierid, s.suppliername order by s.supplierid,s.suppliername;

Output:

71. Find the average sales amount per order by category:

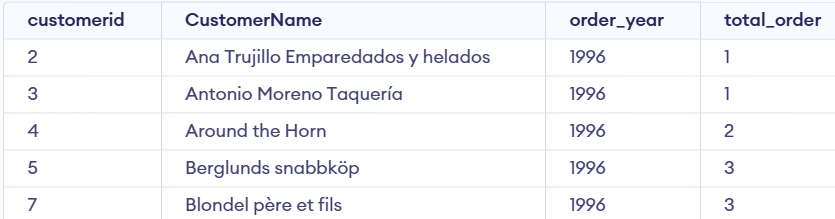
Query:

with category\_order\_totals as ( select c.categoryid,c.categoryname,o.orderid, sum(od.quantity \* p.price) as order\_total from categories c join products p on c.categoryid = p.categoryid join orderdetails od on p.productid = od.productid join orders o on od.orderid = o.orderid group by c.categoryid, c.categoryname, o.orderid )

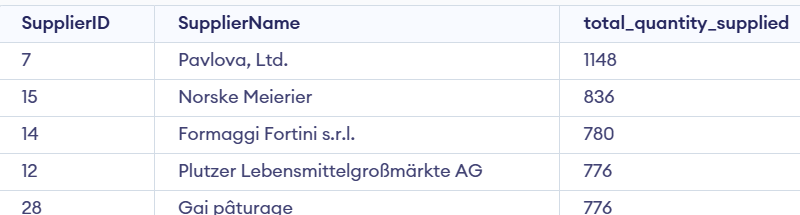
select categoryid, categoryname, round(avg(order\_total), 2) as avg\_sales\_per\_order from category\_order\_totals group by categoryid, categoryname  
Output:

72. Calculate the total number of orders per customer by year:

Query: --Calculate the total number of orders per customer by year with total\_no\_orders as (select c.customerid, c.CustomerName,strftime("%Y",OrderDate) as order\_year,count(o.OrderID) as total\_order from Customers as c join Orders as o on c.customerid=o.customerid group by c.customerid, c.CustomerName,order\_year )

select customerid,CustomerName,order\_year,total\_order from total\_no\_orders  
Output:

73. List the top 5 suppliers by total quantity supplied:

Query: select s.supplierid,s.suppliername, sum(od.quantity) as total\_quantity\_supplied from suppliers s join products p on s.supplierid = p.supplierid join orderdetails od on p.productid = od.productid group by s.supplierid, s.suppliername order by total\_quantity\_supplied desc limit 5;  
Output:

74. Calculate the total revenue for each category by month:

Query: select c.categoryid,c.categoryname,

strftime('%Y-%m', o.orderdate) as order\_month,

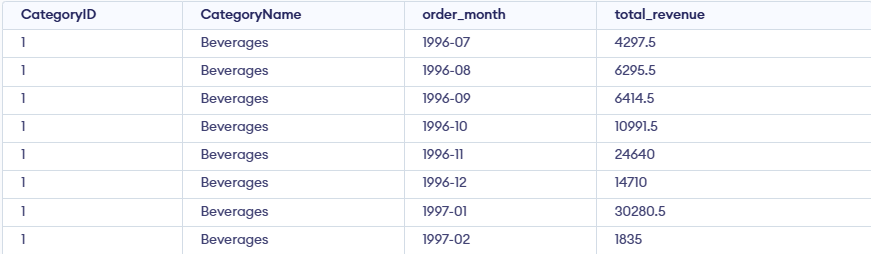
sum(od.quantity \* p.price) as total\_revenue

from categories c

join products p on c.categoryid = p.categoryid

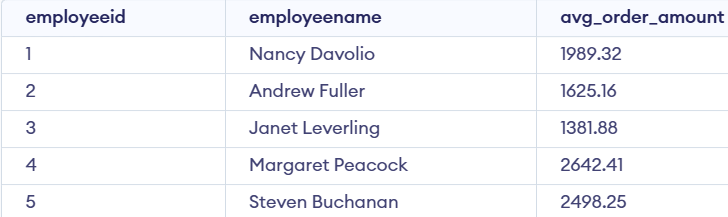
join orderdetails od on p.productid = od.productid

join orders o on od.orderid = o.orderid

group by c.categoryid, c.categoryname, order\_month  
Output:

75. Find the average order amount for each employee:

Query:with order\_totals as ( select e.employeeid, e.firstname || ' ' || e.lastname as employeename, o.orderid, sum(od.quantity \* p.price) as order\_total from employees e join orders o on e.employeeid = o.employeeid join orderdetails od on o.orderid = od.orderid join products p on od.productid = p.productid group by e.employeeid, o.orderid )

select employeeid, employeename, round(avg(order\_total), 2) as avg\_order\_amount from order\_totals group by employeeid, employeename  
Output:

76. List all products that are supplied by more than one supplier:

Query: select productid, count(distinct supplierid) as supplier\_count

from products

group by productid

having supplier\_count > 1;  
Output:

77. List the most frequently ordered product in each category:

Query: with product\_quantity as (

select c.categoryid,c.categoryname,p.productid,p.productname,

sum(od.quantity) as total\_quantity,

row\_number() over (partition by c.categoryid order by sum(od.quantity) desc) as rnk

from categories c

join products p on c.categoryid = p.categoryid

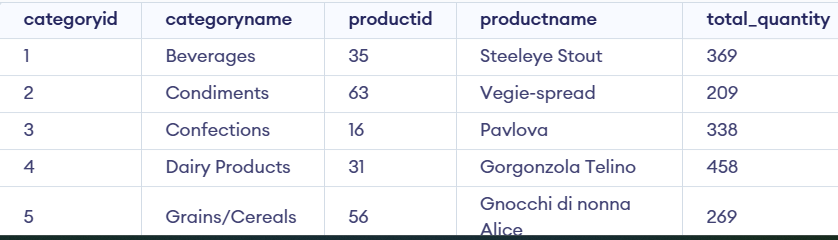
join orderdetails od on p.productid = od.productid

group by c.categoryid, c.categoryname, p.productid, p.productname)

select categoryid,categoryname,productid, productname, total\_quantity

from product\_quantity

where rnk = 1;

Output:

78. Calculate the total revenue for each product by quarter:

Query: with product\_sales as (

select p.productid, p.productname ,strftime('%Y', o.orderdate) as order\_year,

case

when cast(strftime('%m', o.orderdate) as integer) between 1 and 3 then 'Q1'

when cast(strftime('%m', o.orderdate) as integer) between 4 and 6 then 'Q2'

when cast(strftime('%m', o.orderdate) as integer) between 7 and 9 then 'Q3'

when cast(strftime('%m', o.orderdate) as integer) between 10 and 12 then'Q4'

end as quarter,

sum(od.quantity \* p.price) as total\_revenue

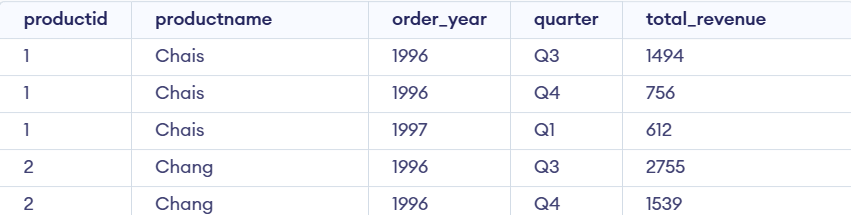
from orderdetails od

join orders o on od.orderid = o.orderid

join products p on od.productid = p.productid

group by p.productid, order\_year, quarter)

select \* from product\_sales

Output:

79. List all customers who have placed orders in every quarter of the year:

Query: with customer\_quarters as (

select c.customerid,c.customername,

strftime('%Y', o.orderdate) as year,

case

when cast(strftime('%m', o.orderdate) as integer) between 1 and 3 then 'Q1'

when cast(strftime('%m', o.orderdate) as integer) between 4 and 6 then 'Q2'

when cast(strftime('%m', o.orderdate) as integer) between 7 and 9 then 'Q3'

when cast(strftime('%m', o.orderdate) as integer) between 10 and 12 then 'Q4'

end as quarter

from customers c

join orders o on c.customerid = o.customerid

group by c.customerid, year, quarter),

quarter\_counts as (

select customerid, customername, year,

count(distinct quarter) as quarters\_ordered

from customer\_quarters

group by customerid, year)

select customerid, customername, year

from quarter\_counts

where quarters\_ordered = 4

80. Find the total revenue generated by each employee:

Query: select e.employeeid, e.firstname || ' ' || e.lastname as employee\_name,

sum(od.quantity \* p.price) as total\_revenue

from employees e

join orders o on e.employeeid = o.employeeid

join orderdetails od on o.orderid = od.orderid

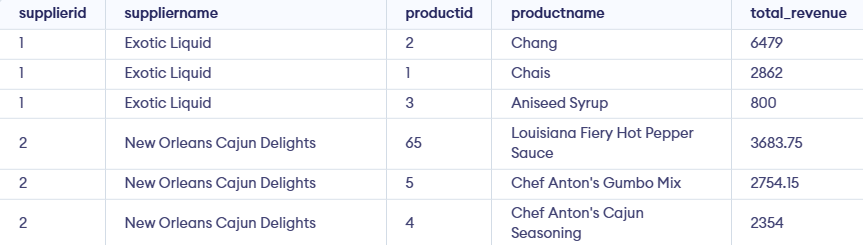
join products p on od.productid = p.productid

group by e.employeeid, e.firstname, e.lastname  
Output:

81. List the top 10 products by revenue for each supplier:

Query: with product\_revenue as ( select s.supplierid,s.suppliername,p.productid,p.productname, round(sum(od.quantity \* p.price),2) as total\_revenue from suppliers s join products p on s.supplierid = p.supplierid join orderdetails od on p.productid = od.productid join orders o on od.orderid = o.orderid group by s.supplierid, p.productid ),

ranked\_products as ( select \*, row\_number() over (partition by supplierid order by total\_revenue desc) as rank from product\_revenue )

select supplierid,suppliername,productid, productname, total\_revenue from ranked\_products where rank <= 10  
Output:

82. Calculate the total quantity ordered for each category by quarter:

Query: select c.categoryid, c.categoryname,

case

when cast(strftime('%m', o.orderdate) as integer) between 1 and 3 then 'Q1'

when cast(strftime('%m', o.orderdate) as integer) between 4 and 6 then 'Q2'

when cast(strftime('%m', o.orderdate) as integer) between 7 and 9 then 'Q3'

when cast(strftime('%m', o.orderdate) as integer) between 10 and 12 then 'Q4'

end as quarter,

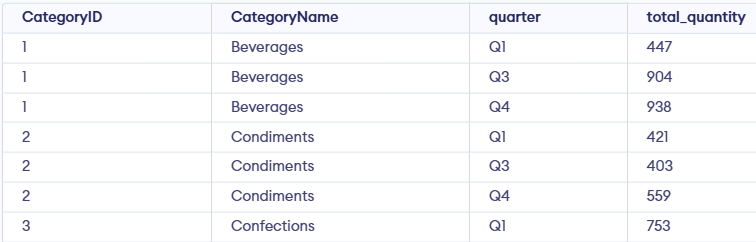
sum(od.quantity) as total\_quantity

from categories c

join products p on c.categoryid = p.categoryid

join orderdetails od on p.productid = od.productid

join orders o on od.orderid = o.orderid

group by c.categoryid, quarter  
Output:

83. List the top 5 employees by total revenue generated:

Query: select e.employeeid, e.firstname || ' ' || e.lastname as employeename,

sum(od.quantity \* p.price) as total\_revenue

from employees e

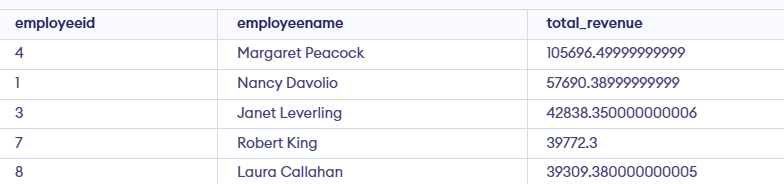
join orders o on e.employeeid = o.employeeid

join orderdetails od on o.orderid = od.orderid

join products p on od.productid = p.productid

group by e.employeeid

order by total\_revenue desc

limit 5;  
Output:

84. Find the most popular shipping method by total orders shipped:

Query: select s.shipperid,s.shippername,

count(o.orderid) as total\_orders

from shippers s

join orders o on s.shipperid = o.shipperid

group by s.shipperid, s.shippername

order by total\_orders desc

limit 1;  
Output:

85. Calculate the total revenue for each shipper by year:

Query: select s.shipperid,s.shippername,

strftime('%Y', o.orderdate) as order\_year,

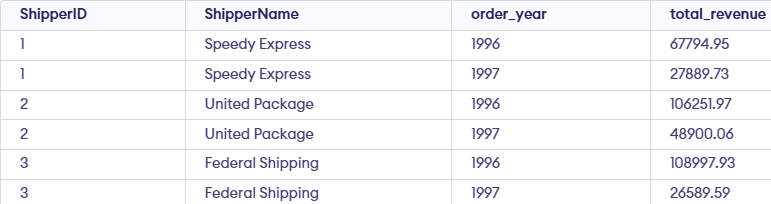
round(sum(od.quantity \* p.price),2) as total\_revenue

from shippers s

join orders o on s.shipperid = o.shipperid

join orderdetails od on o.orderid = od.orderid

join products p on od.productid = p.productid

group by s.shipperid, order\_year  
Output:

86. List all products that have never been ordered:

Query: select p.productid, p.productname

from products p

left join orderdetails od on p.productid = od.productid

where od.productid is null;  
Output:

87. List the total sales amount for each supplier by year:

Query: select s.supplierid,s.suppliername,

strftime('%Y', o.orderdate) as order\_year,

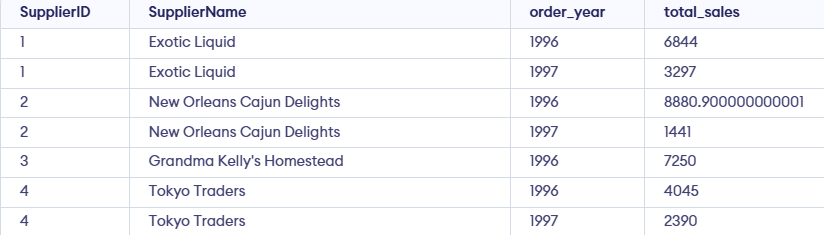
sum(od.quantity \* p.price) as total\_sales

from suppliers s

join products p on s.supplierid = p.supplierid

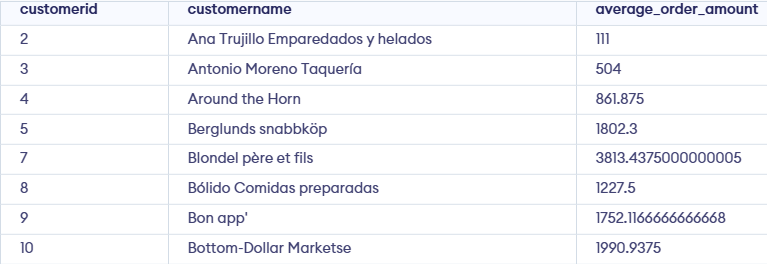
join orderdetails od on p.productid = od.productid

join orders o on od.orderid = o.orderid

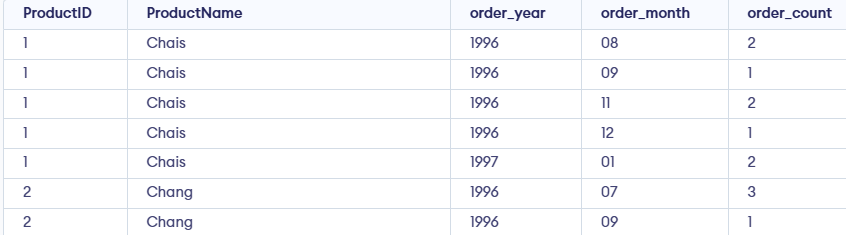
group by s.supplierid, order\_year  
Output:

88. Calculate the average order amount by customer:

Query: with customer\_orders as ( select c.customerid,c.customername,o.orderid, sum(od.quantity \* p.price) as order\_total from customers c join orders o on c.customerid = o.customerid join orderdetails od on o.orderid = od.orderid join products p on od.productid = p.productid group by c.customerid, o.orderid )

select customerid,customername, avg(order\_total) as average\_order\_amount from customer\_orders group by customerid, customername  
Output:

89. List the number of orders per product by month:

Query: select p.productid, p.productname, strftime('%Y', o.orderdate) as order\_year, strftime('%m', o.orderdate) as order\_month, count(distinct o.orderid) as order\_count from orderdetails od join orders o on od.orderid = o.orderid join products p on od.productid = p.productid group by p.productid, order\_year, order\_month  
Output:

90. Find the top 3 customers by total revenue generated:

Query: select c.customerid,c.customername,

round( sum(od.quantity \* p.price),2) as total\_revenue

from customers c

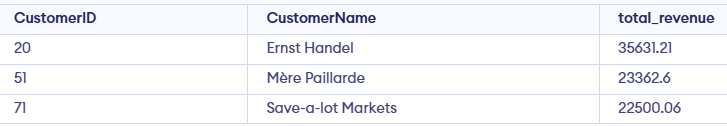
join orders o on c.customerid = o.customerid

join orderdetails od on o.orderid = od.orderid

join products p on od.productid = p.productid

group by c.customerid, c.customername

order by total\_revenue desc

limit 3  
Output:

91. Calculate the total revenue for each product by category:

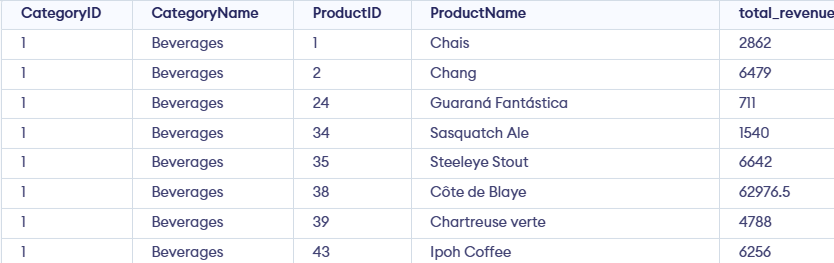
Query: select c.categoryid,c.categoryname, p.productid, p.productname,

round(sum(od.quantity \* p.price),2) as total\_revenue

from products p

join categories c on p.categoryid = c.categoryid

join orderdetails od on p.productid = od.productid

group by c.categoryid, p.productid  
Output:

92. List the total quantity ordered for each product by year:

Query: select p.productid, p.productname,

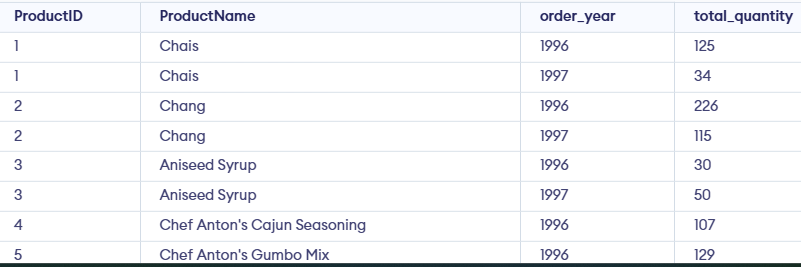
strftime('%Y', o.orderdate) as order\_year,

sum(od.quantity) as total\_quantity

from orderdetails od

join orders o on od.orderid = o.orderid

join products p on od.productid = p.productid

group by p.productid, order\_year  
Output:

93. List the top 5 products by total quantity ordered:

Query: select p.productid,p.productname,

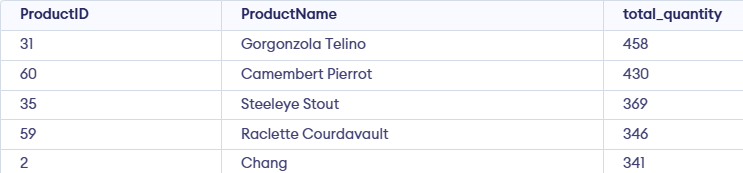
sum(od.quantity) as total\_quantity

from orderdetails od

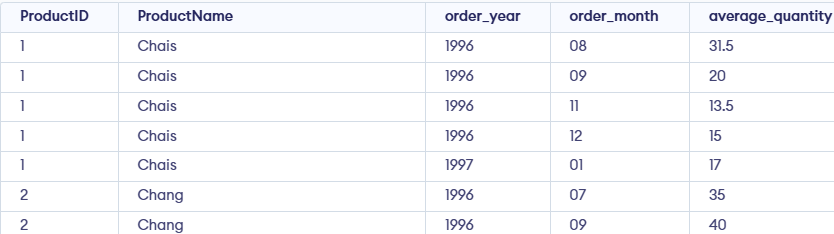
join products p on od.productid = p.productid

group by p.productid, p.productname

order by total\_quantity desc

limit 5;  
Output:

94. Calculate the average quantity ordered per product by month:

Query: select p.productid, p.productname, strftime('%Y', o.orderdate) as order\_year, strftime('%m', o.orderdate) as order\_month, round(avg(od.quantity),2) as average\_quantity from orderdetails od join orders o on od.orderid = o.orderid join products p on od.productid = p.productid group by p.productid, order\_year, order\_month  
Output:

95. Find the total revenue generated by each employee by year:

Query: select e.employeeid, e.firstname || ' ' || e.lastname as employeename,

strftime('%Y', o.orderdate) as order\_year,

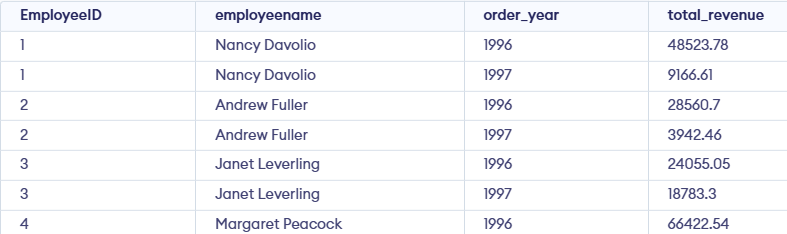
round(sum(od.quantity \* p.price),2) as total\_revenue

from employees e

join orders o on e.employeeid = o.employeeid

join orderdetails od on o.orderid = od.orderid

join products p on od.productid = p.productid

group by e.employeeid, order\_year  
Output:

96. List the top 3 categories by total quantity ordered:

Query: select c.categoryid,c.categoryname,

sum(od.quantity) as total\_quantity

from categories c

join products p on c.categoryid = p.categoryid

join orderdetails od on p.productid = od.productid

group by c.categoryid, c.categoryname

order by total\_quantity desc

limit 3;

Output:

97. List all customers who have never placed an order:

Query: select c.customerid,c.customername

from customers c

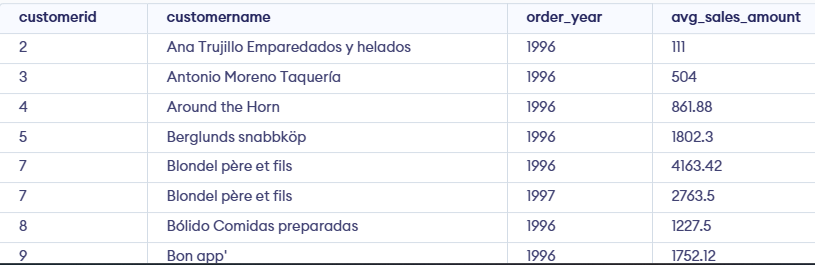
left join orders o on c.customerid = o.customerid

where o.orderid is null;  
Output:



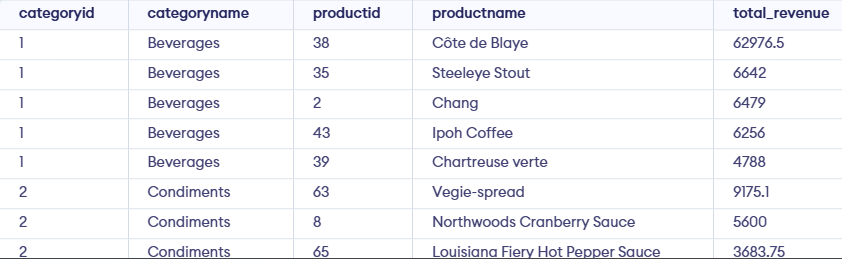
98. Calculate the average sales amount per customer by year:

Query: with customer\_sales as ( select c.customerid,c.customername,o.orderid, strftime('%Y', o.orderdate) as order\_year, sum(od.quantity \* p.price) as order\_total from customers c join orders o on c.customerid = o.customerid join orderdetails od on o.orderid = od.orderid join products p on od.productid = p.productid group by c.customerid, o.orderid)

select customerid, customername, order\_year, round(avg(order\_total), 2) as avg\_sales\_amount from customer\_sales group by customerid, order\_year  
Output:

99. List the top 5 products by total revenue for each category:

Query: with product\_revenue as ( select c.categoryid, c.categoryname,p.productid,p.productname, sum(od.quantity \* p.price) as total\_revenue, row\_number() over ( partition by c.categoryid order by sum(od.quantity \* p.price) desc) as rank from categories c join products p on c.categoryid = p.categoryid join orderdetails od on p.productid = od.productid join orders o on od.orderid = o.orderid group by c.categoryid, c.categoryname, p.productid, p.productname )

select categoryid,categoryname,productid,productname, round(total\_revenue, 2) as total\_revenue from product\_revenue where rank <= 5 order by categoryid, rank;  
Output:

100. Calculate the total revenue for each shipper by month:

Query: SELECT s.shipperid,s.shippername,

strftime('%Y-%m', o.orderdate) AS order\_month,

ROUND(SUM(od.quantity \* p.price), 2) AS total\_revenue

FROM shippers s

JOIN orders o ON s.shipperid = o.shipperid

JOIN orderdetails od ON o.orderid = od.orderid

JOIN products p ON od.productid = p.productid

GROUP BY s.shipperid, order\_month  
Output: