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
  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
- Once your editor is opened, you need to run queries from two scripts in expected order
  - DDL Script: This is a DDL script which creates table needed for your hands-on to be executed first
  - DML Script: 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

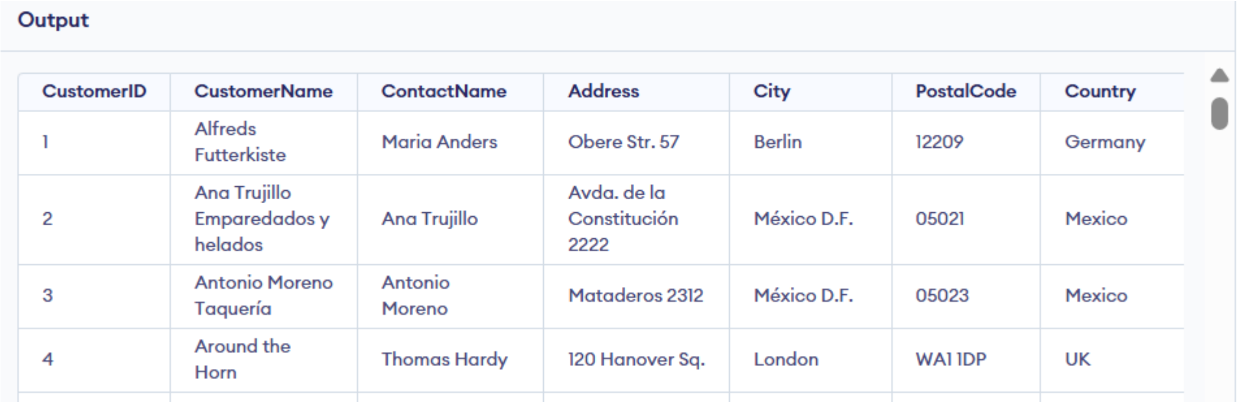
- Once data is loaded, you can proceed with solving each question.
- For each question, you are expected to provide the query and the output.

  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:

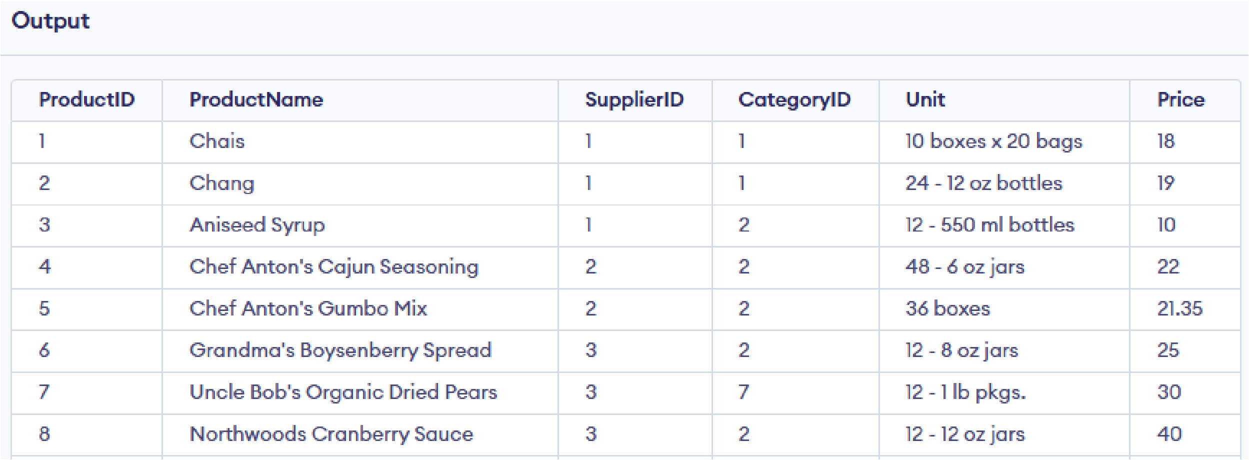


Output

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK

2. List all products:

Query: Select * from products
Output:



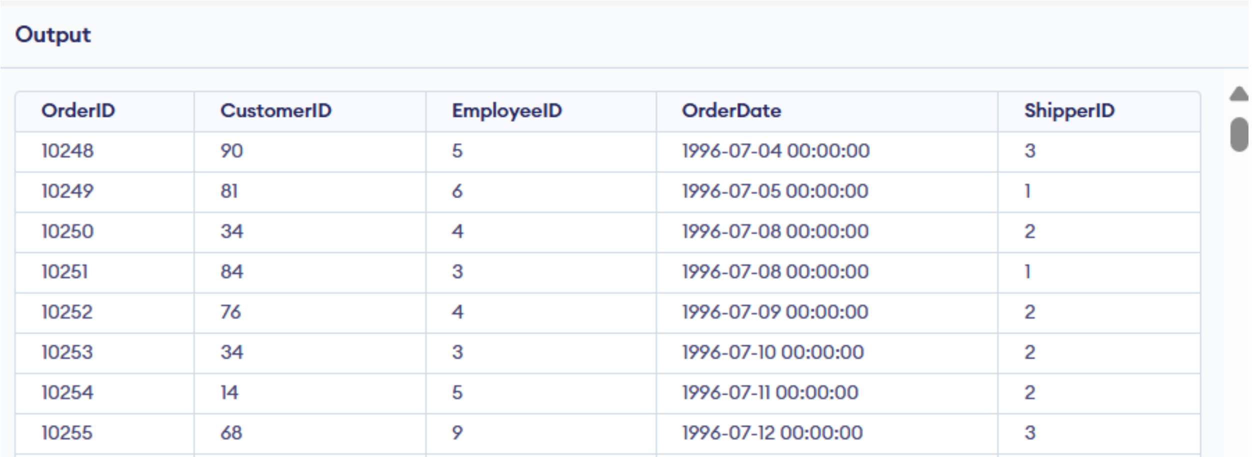
Output

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30
8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40

3. List all orders:

Query: Select * from orders

Output:



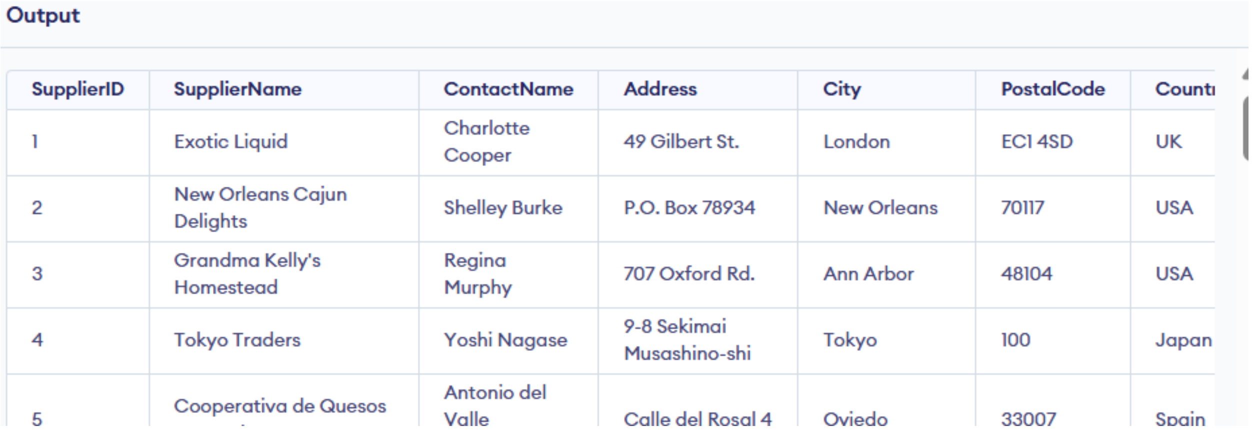
Output

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	1996-07-04 00:00:00	3
10249	81	6	1996-07-05 00:00:00	1
10250	34	4	1996-07-08 00:00:00	2
10251	84	3	1996-07-08 00:00:00	1
10252	76	4	1996-07-09 00:00:00	2
10253	34	3	1996-07-10 00:00:00	2
10254	14	5	1996-07-11 00:00:00	2
10255	68	9	1996-07-12 00:00:00	3

4. List all suppliers:

Query: Select * from suppliers

Output:



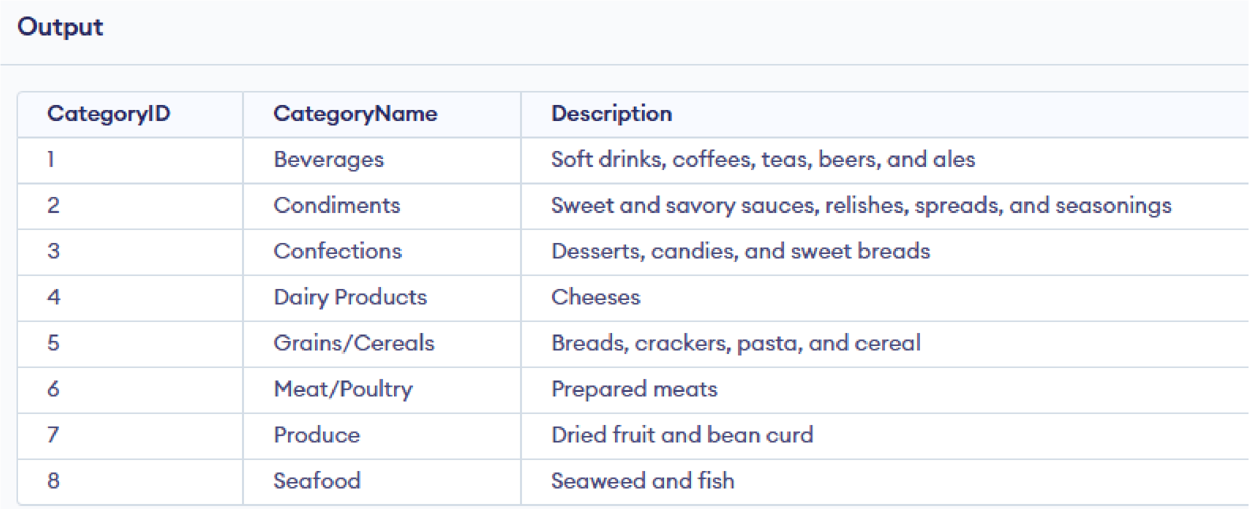
Output

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Countr
1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	London	EC1 4SD	UK
2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA
3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA
4	Tokyo Traders	Yoshi Nagase	9-8 Sekimai Musashino-shi	Tokyo	100	Japan
5	Cooperativa de Quesos	Antonio del Valle	Calle del Rosal 4	Oviedo	33007	Spain

5. List all categories:

Query: Select * from categories

Output:



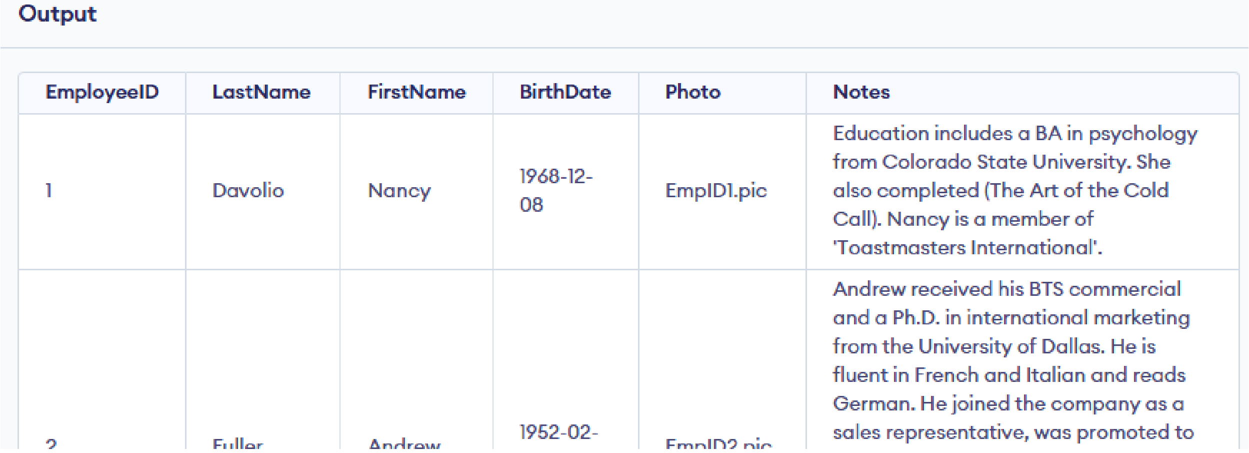
Output

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

6. List all employees:

Query: Select * from employees

Output:

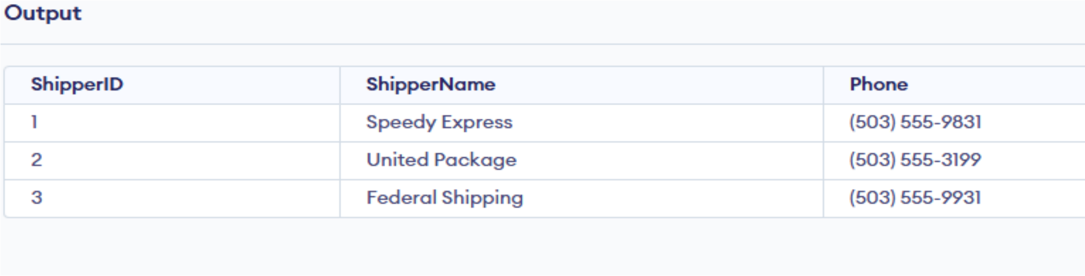


Output

EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
1	Davolio	Nancy	1968-12-08	EmpID1.pic	Education includes a BA in psychology from Colorado State University. She also completed (The Art of the Cold Call). Nancy is a member of 'Toastmasters International'.
2	Fuller	Andrew	1952-02-	EmpID2.pic	Andrew received his BTS commercial and a Ph.D. in international marketing from the University of Dallas. He is fluent in French and Italian and reads German. He joined the company as a sales representative, was promoted to

7. List all shippers:

Query: Select * from shippers



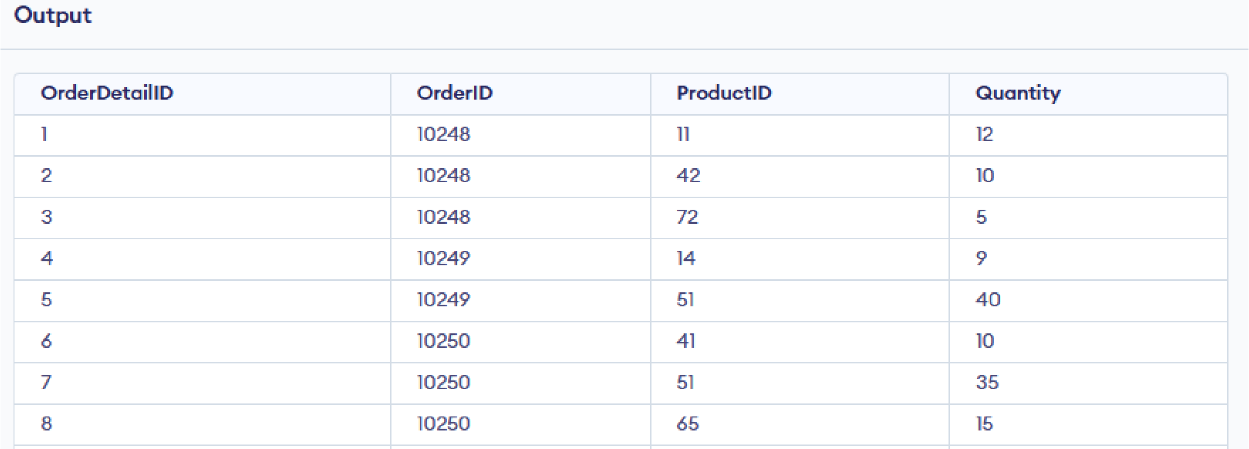
Output

ShipperID	ShipperName	Phone
1	Speedy Express	(503) 555-9831
2	United Package	(503) 555-3199
3	Federal Shipping	(503) 555-9931

8. List all order details:

Query: Select * from orderDetails

Output:



Output

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40
6	10250	41	10
7	10250	51	35
8	10250	65	15

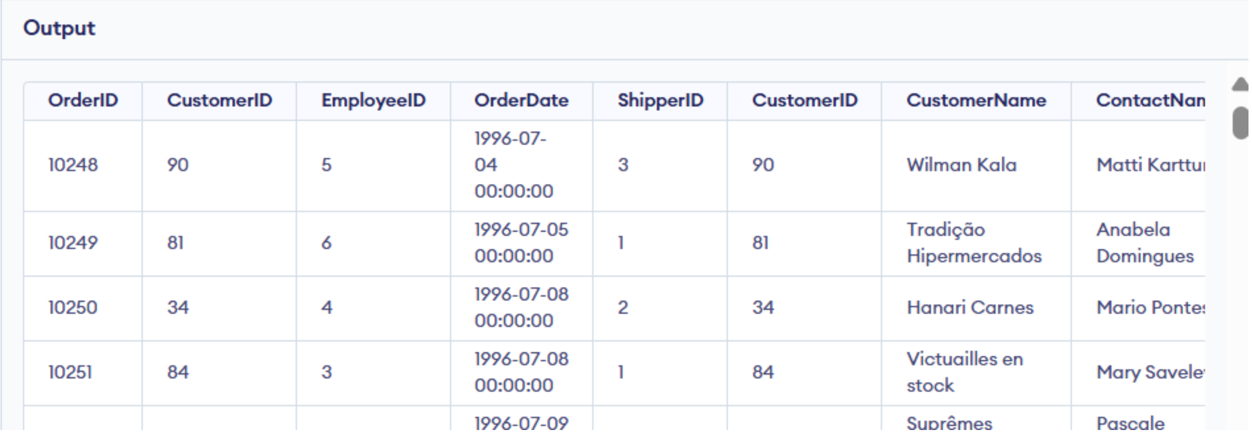
9. List all orders with customer details:

Query: Select * from orders as o

join customers as c on

o.CustomerID = c.CustomerID

Output:



Output

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	CustomerID	CustomerName	ContactNan
10248	90	5	1996-07-04 00:00:00	3	90	Wilman Kala	Matti Karttu
10249	81	6	1996-07-05 00:00:00	1	81	Tradição Hipermercados	Anabela Domingues
10250	34	4	1996-07-08 00:00:00	2	34	Hanari Carnes	Mario Ponte
10251	84	3	1996-07-08 00:00:00	1	84	Victuailles en stock	Mary Savele
			1996-07-09			Suprêmes	Pascale

10. List all orders with employee details:

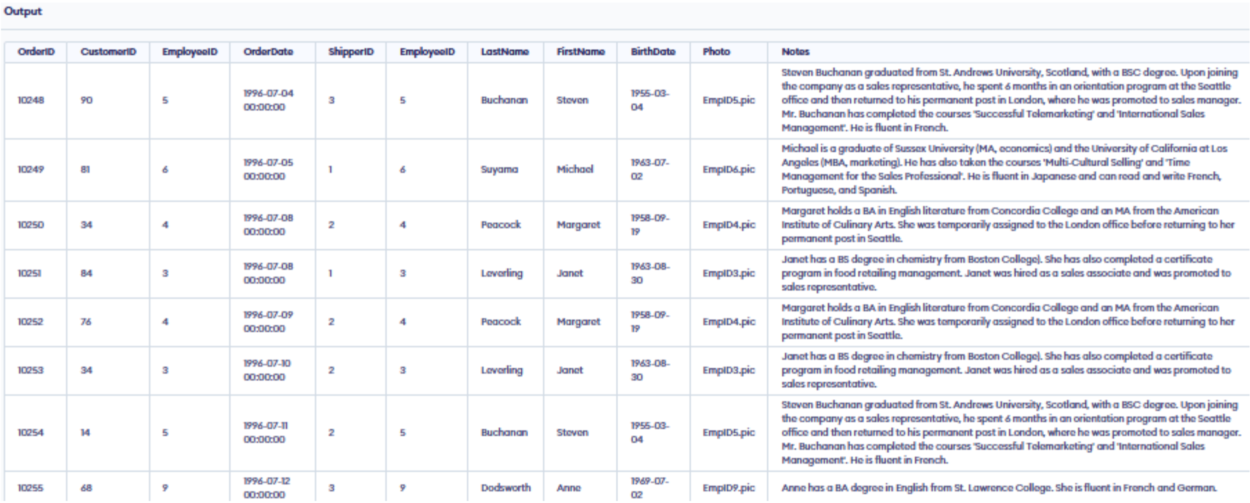
Query:

Select * from orders as o

join employees as e on

```
o.EmployeeID = e.EmployeeID
```

Output:



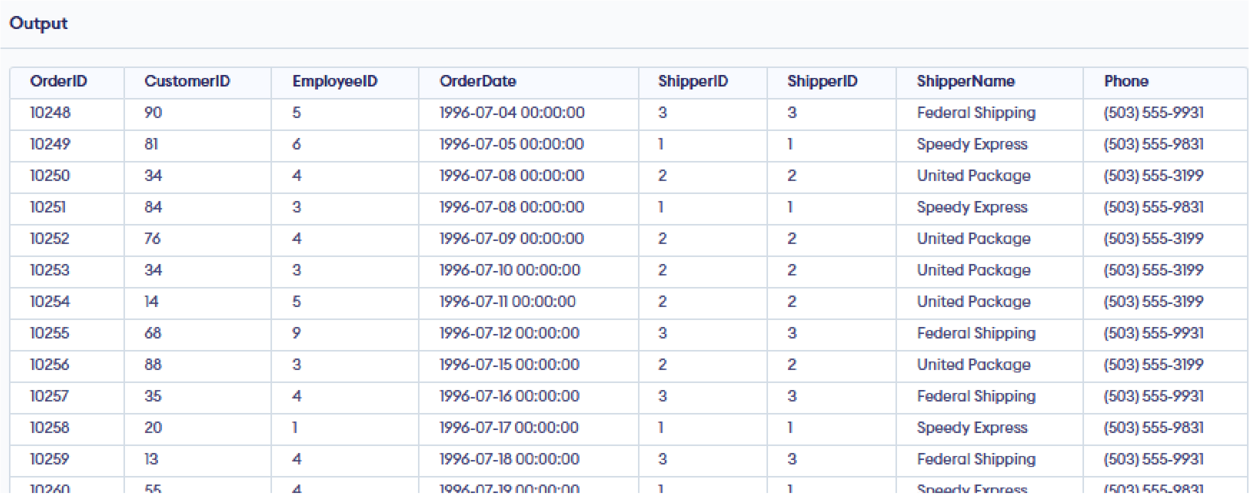
Output

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
10248	90	5	1996-07-04 00:00:00	3	5	Buchanan	Steven	1955-03-04	EmpID5.pic	Steven Buchanan graduated from St. Andrews University, Scotland, with a BSC degree. Upon joining the company as a sales representative, he spent 6 months in an orientation program at the Seattle office and then returned to his permanent post in London, where he was promoted to sales manager. Mr. Buchanan has completed the courses 'Successful Telemarketing' and 'International Sales Management'. He is fluent in French.
10249	81	6	1996-07-05 00:00:00	1	6	Suyama	Michael	1963-07-02	EmpID6.pic	Michael is a graduate of Sussex University (MA, economics) and the University of California at Los Angeles (MBA, marketing). He has also taken the courses 'Multi-Cultural Selling' and 'Time Management for the Sales Professional'. He is fluent in Japanese and can read and write French, Portuguese, and Spanish.
10250	34	4	1996-07-08 00:00:00	2	4	Peacock	Margaret	1958-09-19	EmpID4.pic	Margaret holds a BA in English literature from Concordia College and an MA from the American Institute of Culinary Arts. She was temporarily assigned to the London office before returning to her permanent post in Seattle.
10251	84	3	1996-07-08 00:00:00	1	3	Leverling	Janet	1963-08-30	EmpID3.pic	Janet has a BS degree in chemistry from Boston College). She has also completed a certificate program in food retailing management. Janet was hired as a sales associate and was promoted to sales representative.
10252	76	4	1996-07-09 00:00:00	2	4	Peacock	Margaret	1958-09-19	EmpID4.pic	Margaret holds a BA in English literature from Concordia College and an MA from the American Institute of Culinary Arts. She was temporarily assigned to the London office before returning to her permanent post in Seattle.
10253	34	3	1996-07-10 00:00:00	2	3	Leverling	Janet	1963-08-30	EmpID3.pic	Janet has a BS degree in chemistry from Boston College). She has also completed a certificate program in food retailing management. Janet was hired as a sales associate and was promoted to sales representative.
10254	14	5	1996-07-11 00:00:00	2	5	Buchanan	Steven	1955-03-04	EmpID5.pic	Steven Buchanan graduated from St. Andrews University, Scotland, with a BSC degree. Upon joining the company as a sales representative, he spent 6 months in an orientation program at the Seattle office and then returned to his permanent post in London, where he was promoted to sales manager. Mr. Buchanan has completed the courses 'Successful Telemarketing' and 'International Sales Management'. He is fluent in French.
10255	68	9	1996-07-12 00:00:00	3	9	Dodsworth	Anne	1969-07-02	EmpID9.pic	Anne has a BA degree in English from St. Lawrence College. She is fluent in French and German.

11. List all orders with shipper details:

```
Query:Select * from orders as o

             join Shippers  as e on

            o.ShipperID = e.ShipperID
```

Output:



Output

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	ShipperID	ShipperName	Phone
10248	90	5	1996-07-04 00:00:00	3	3	Federal Shipping	(503) 555-9931
10249	81	6	1996-07-05 00:00:00	1	1	Speedy Express	(503) 555-9831
10250	34	4	1996-07-08 00:00:00	2	2	United Package	(503) 555-3199
10251	84	3	1996-07-08 00:00:00	1	1	Speedy Express	(503) 555-9831
10252	76	4	1996-07-09 00:00:00	2	2	United Package	(503) 555-3199
10253	34	3	1996-07-10 00:00:00	2	2	United Package	(503) 555-3199
10254	14	5	1996-07-11 00:00:00	2	2	United Package	(503) 555-3199
10255	68	9	1996-07-12 00:00:00	3	3	Federal Shipping	(503) 555-9931
10256	88	3	1996-07-15 00:00:00	2	2	United Package	(503) 555-3199
10257	35	4	1996-07-16 00:00:00	3	3	Federal Shipping	(503) 555-9931
10258	20	1	1996-07-17 00:00:00	1	1	Speedy Express	(503) 555-9831
10259	13	4	1996-07-18 00:00:00	3	3	Federal Shipping	(503) 555-9931
10260	55	4	1996-07-19 00:00:00	1	1	Speedy Express	(503) 555-9831

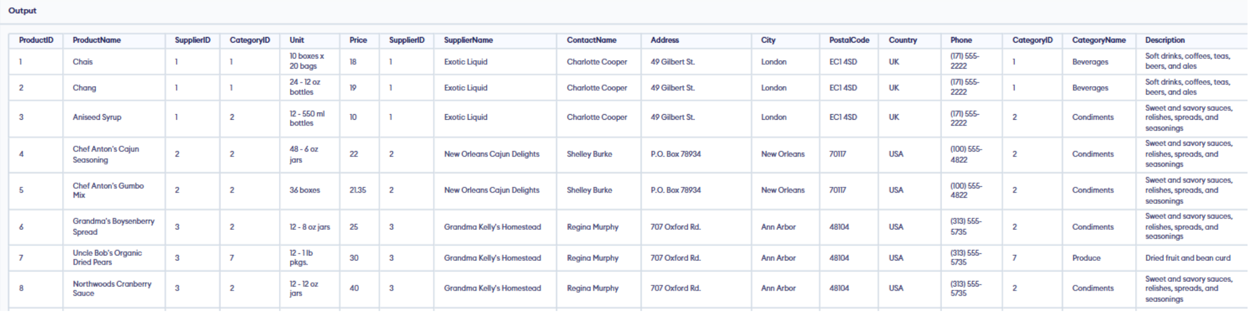
12. List all products along with their supplier and category:

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:



Output

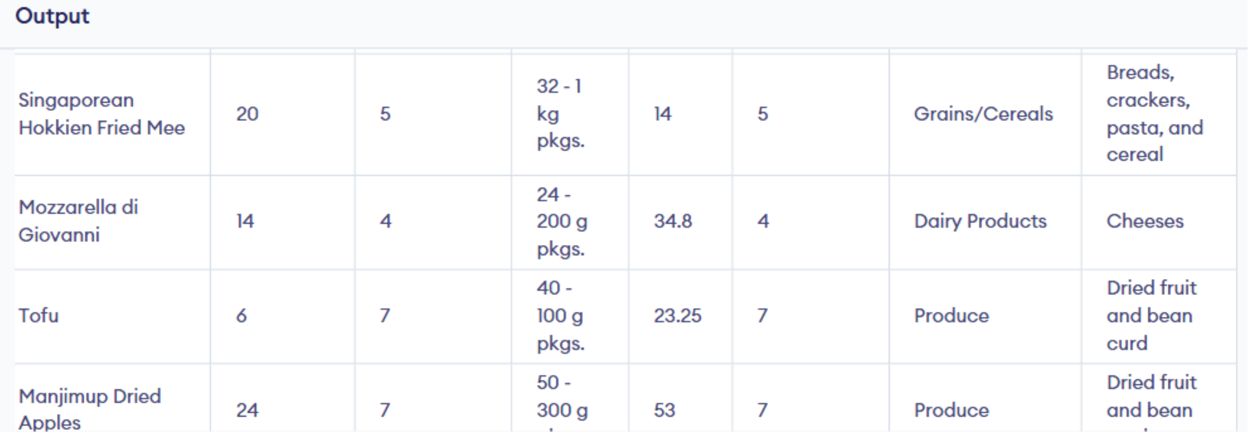
ProductID	ProductName	SupplierID	CategoryID	Unit	Price	SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone	CategoryID	CategoryName	Description
1	Chais	1	1	10 boxes x 20 bags	18	1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	London	EC1 4SD	UK	(171) 555-2222	1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Chang	1	1	24 - 12 oz bottles	19	1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	London	EC1 4SD	UK	(171) 555-2222	1	Beverages	Soft drinks, coffees, teas, beers, and ales
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	London	EC1 4SD	UK	(171) 555-2222	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22	2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA	(100) 555-4822	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35	2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA	(100) 555-4822	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25	3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA	(313) 555-5735	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30	3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA	(313) 555-5735	7	Produce	Dried fruit and bean curd
8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40	3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA	(313) 555-5735	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings

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:



Output

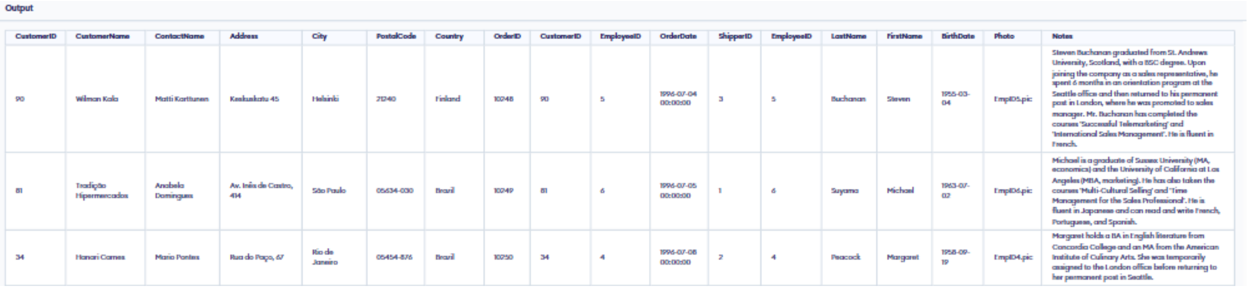
Singaporean Hokkien Fried Mee	20	5	32 - 1 kg pkgs.	14	5	Grains/Cereals	Breads, crackers, pasta, and cereal
Mozzarella di Giovanni	14	4	24 - 200 g pkgs.	34.8	4	Dairy Products	Cheeses
Tofu	6	7	40 - 100 g pkgs.	23.25	7	Produce	Dried fruit and bean curd
Manjimup Dried Apples	24	7	50 - 300 g	53	7	Produce	Dried fruit and bean

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:



Output

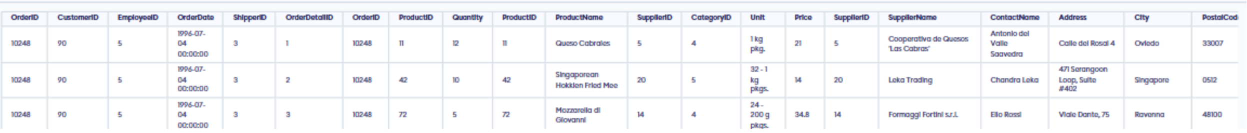
CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country	OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland	10248	90	5	1996-07-04 00:00:00	3	5	Buchanan	Steven	1955-03-04	EmpID5.pic	Steven Buchanan graduated from St. Andrews University, Scotland, with a BSC degree. Upon joining the company as a sales representative, he spent 6 months in an orientation program at the Seattle office and then returned to his permanent post in London, where he was promoted to sales manager. Mr. Buchanan has completed the courses 'Successful Telemarketing' and 'International Sales Management'. He is fluent in French.
81	Tradição Hipermercados	Anabela Domingues	Av. Inês de Castro, 414	São Paulo	05634-030	Brazil	10249	81	6	1996-07-05 00:00:00	1	6	Suyama	Michael	1963-07-02	EmpID6.pic	Michael is a graduate of Sussex University (MA, economics) and the University of California at Los Angeles (MBA, marketing). He has also taken the courses 'Multi-Cultural Selling' and 'Time Management for the Sales Professional'. He is fluent in Japanese and can read and write French, Portuguese, and Spanish.
34	Hanari Carnes	Mario Pontes	Rua do Paço, 67	Rio de Janeiro	05454-876	Brazil	10250	34	4	1996-07-08 00:00:00	2	4	Peacock	Margaret	1958-09-19	EmpID4.pic	Margaret holds a BA in English literature from Concordia College and an MA from the American Institute of Culinary Arts. She was temporarily assigned to the London office before returning to her permanent post in Seattle.

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:



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	OrderDetailID	OrderID	ProductID	Quantity	ProductID	ProductName	SupplierID	CategoryID	Unit	Price	SupplierID	SupplierName	ContactName	Address	City	PostalCod
10248	90	5	1996-07-04 00:00:00	3	1	10248	11	12	11	Queso Cabrales	5	4	1 kg pkg.	21	5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Calle del Rosal 4	Oviedo	33007
10248	90	5	1996-07-04 00:00:00	3	2	10248	42	10	42	Singaporean Hokkien Fried Mee	20	5	32 - 1 kg pkgs.	14	20	Leka Trading	Chandra Leka	471 Serangoon Loop, Suite #402	Singapore	0512
10248	90	5	1996-07-04 00:00:00	3	3	10248	72	5	72	Mozzarella di Giovanni	14	4	24 - 200 g pkgs.	34.8	14	Formaggi Fortini s.r.l.	Elio Rossi	Viale Dante, 75	Ravenna	48100

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

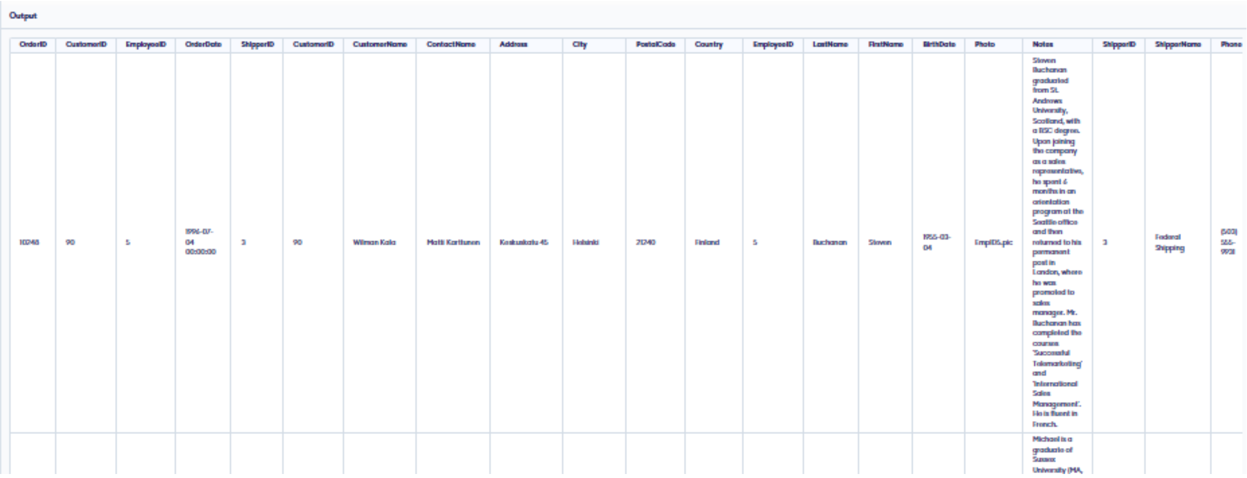
join Employees   as e

on o.EmployeeID = e.EmployeeID

join Shippers as s

on o.ShipperID = s.ShipperID

Output:

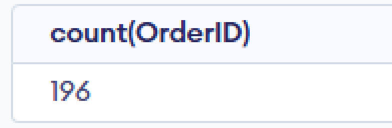


Output

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country	EmployeeID	LastName	FirstName	BirthDate	Photo	Notes	ShipperID	ShipperName	Phone
10248	90	5	1996-07-04 00:00:00	3	90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland	5	Buchanan	Steven	1955-03-04	EmpID5.pic	Steven Buchanan graduated from St. Andrews University, Scotland, with a BSC degree. Upon joining the company as a sales representative, he spent 6 months in an orientation program at the Seattle office and then returned to his permanent post in London, where he was promoted to sales manager. Mr. Buchanan has completed the courses 'Successful Telemarketing' and 'International Sales Management'. He is fluent in French.	3	Federal Shipping	(503) 555-9931
																	Michael is a graduate of Sussex University (MA,			

17. Count total number of orders:

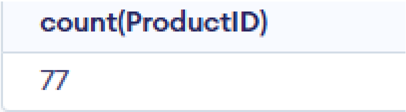
Query: Select count(OrderID) from Orders

Output: 

count(OrderID)
196

18. Count total number of products:

Query: Select count(ProductID) from Products

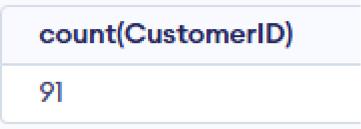
Output: 

count(ProductID)
77

19. Count total number of customers:

Query: Select count(CustomerID) from Customers

Output:



count(CustomerID)
91

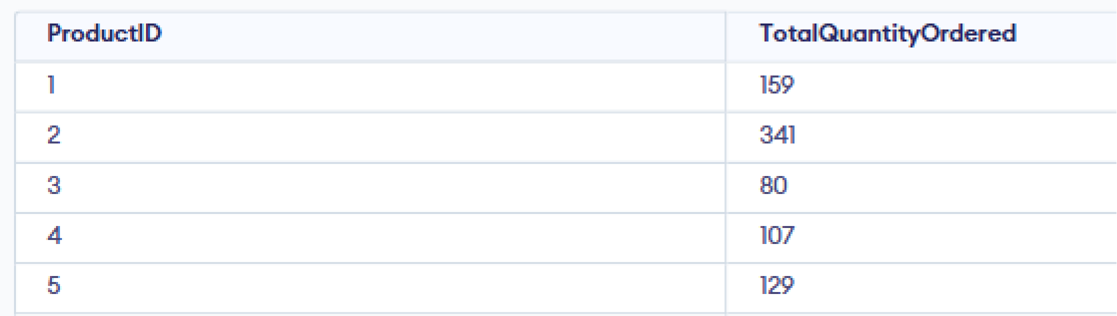
20. Find the total quantity of each product ordered:

Query: SELECT ProductID, SUM(Quantity) AS TotalQuantityOrdered

FROM OrderDetails

GROUP BY ProductID;

Output:



ProductID	TotalQuantityOrdered
1	159
2	341
3	80
4	107
5	129

21. Find the total sales amount for each product:

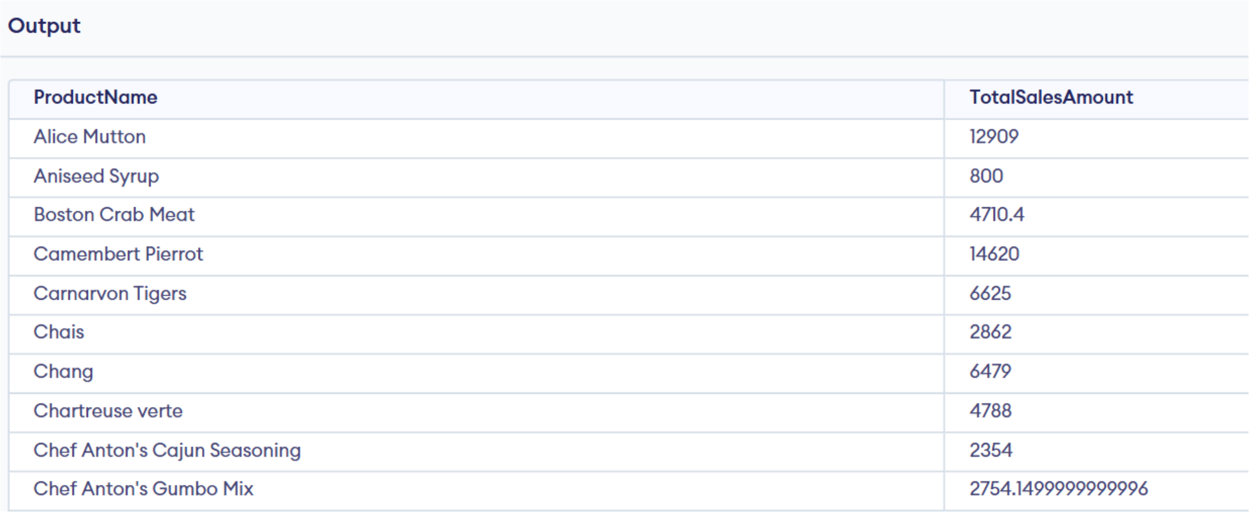
Query: SELECT p.ProductName,SUM(od.Quantity * p.Price) AS TotalSalesAmount

FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY p.ProductName;

Output:



Output

ProductName	TotalSalesAmount
Alice Mutton	12909
Aniseed Syrup	800
Boston Crab Meat	4710.4
Camembert Pierrot	14620
Carnarvon Tigers	6625
Chais	2862
Chang	6479
Chartreuse verte	4788
Chef Anton's Cajun Seasoning	2354
Chef Anton's Gumbo Mix	2754.1499999999996

22. Find the average order total:

Query: SELECT avg(TotalSalesAmount)

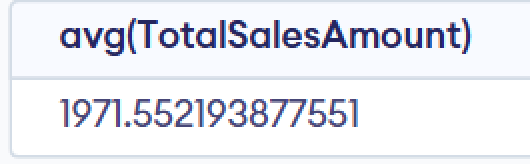
from (select od.orderid, SUM(od.Quantity * p.Price) AS TotalSalesAmount

FROM OrderDetails od

JOIN Products p ON od.ProductID = p.ProductID

GROUP BY od.orderid) as orderTotal ;

Output:

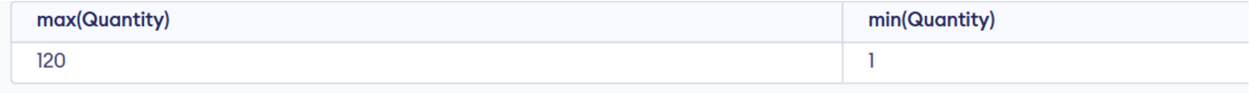


avg(TotalSalesAmount)
1971.552193877551

23. Find the maximum and minimum order quantities:

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

Output:



max(Quantity)	min(Quantity)
120	1

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

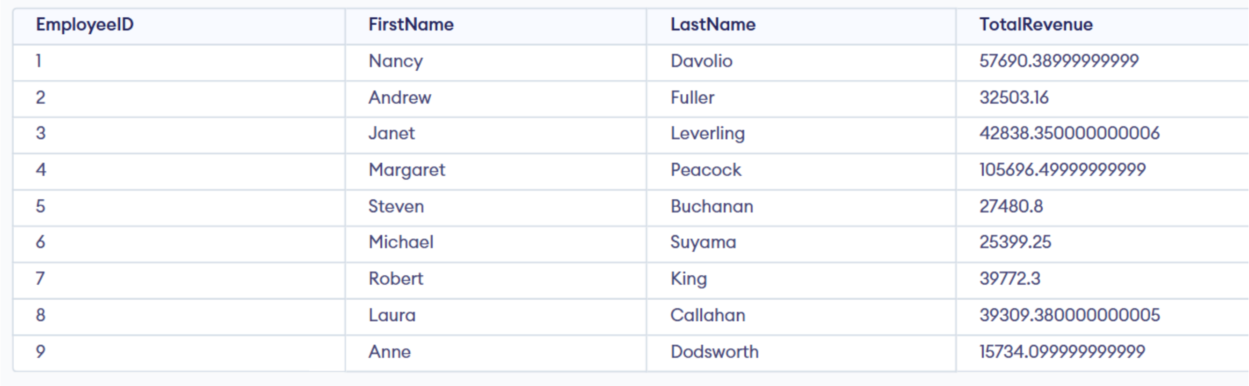
join orders as o on e.EmployeeID = o.EmployeeID

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:



EmployeeID	FirstName	LastName	TotalRevenue
1	Nancy	Davolio	57690.38999999999
2	Andrew	Fuller	32503.16
3	Janet	Leverling	42838.350000000006
4	Margaret	Peacock	105696.49999999999
5	Steven	Buchanan	27480.8
6	Michael	Suyama	25399.25
7	Robert	King	39772.3
8	Laura	Callahan	39309.380000000005
9	Anne	Dodsworth	15734.099999999999

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

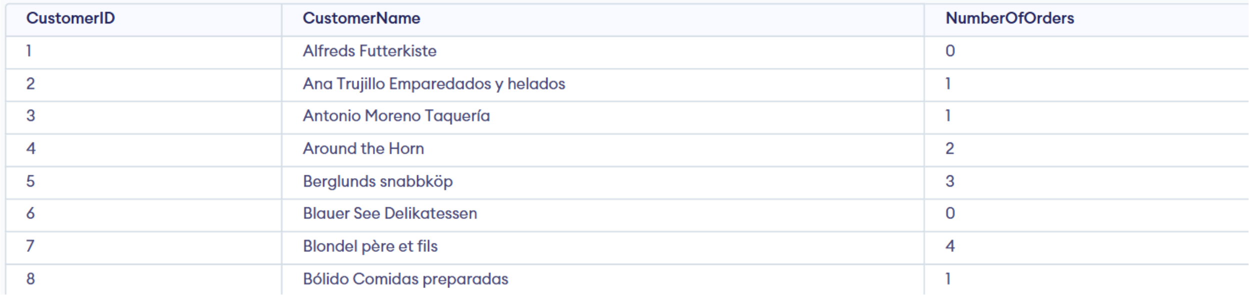
Query: SELECT c.CustomerID, c.CustomerName,COUNT(o.OrderID) AS NumberOfOrders

FROM Customers c

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

GROUP BY c.CustomerID, c.CustomerName;

Output:



CustomerID	CustomerName	NumberOfOrders
1	Alfreds Futterkiste	0
2	Ana Trujillo Emparedados y helados	1
3	Antonio Moreno Taquería	1
4	Around the Horn	2
5	Berglunds snabbköp	3
6	Blauer See Delikatessen	0
7	Blondel père et fils	4
8	Bólido Comidas preparadas	1

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

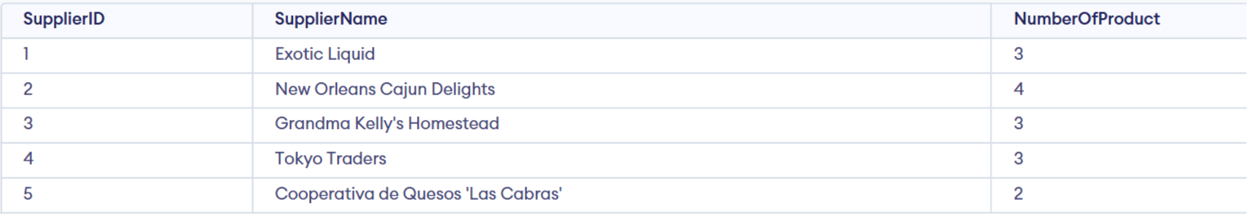
Query: SELECT s.SupplierID, s.SupplierName,COUNT(p.ProductID) AS NumberOfProduct

FROM Suppliers s

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

GROUP BY s.SupplierID, s.SupplierName;

Output:



SupplierID	SupplierName	NumberOfProduct
1	Exotic Liquid	3
2	New Orleans Cajun Delights	4
3	Grandma Kelly's Homestead	3
4	Tokyo Traders	3
5	Cooperativa de Quesos 'Las Cabras'	2

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

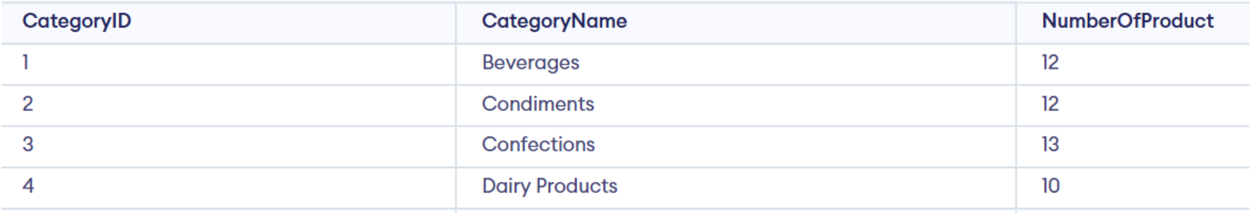
Query: SELECT c.CategoryID, c.CategoryName,COUNT(p.ProductID) AS NumberOfProduct

FROM Categories   c

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

GROUP BY c.CategoryID, c.CategoryName;

Output:



CategoryID	CategoryName	NumberOfProduct
1	Beverages	12
2	Condiments	12
3	Confections	13
4	Dairy Products	10

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

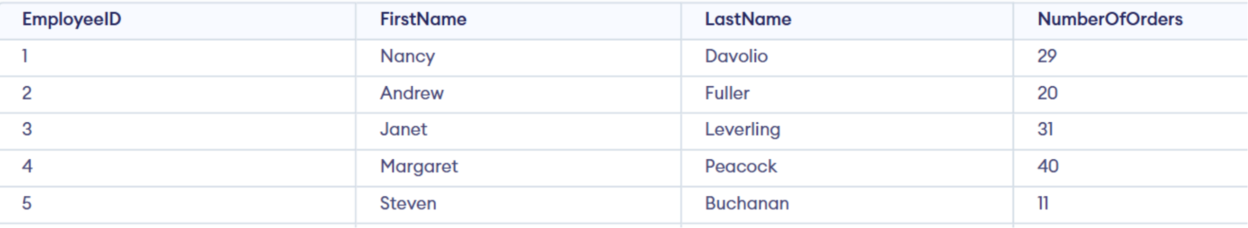
Query: SELECT e.EmployeeID,e.FirstName, e.LastName,COUNT(o.OrderID) AS NumberOfOrders

FROM Employees   e

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

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

Output:



EmployeeID	FirstName	LastName	NumberOfOrders
1	Nancy	Davolio	29
2	Andrew	Fuller	20
3	Janet	Leverling	31
4	Margaret	Peacock	40
5	Steven	Buchanan	11

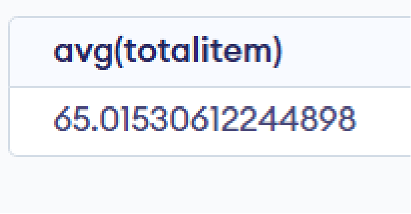
29. Find the average number of items per order:

Query: select avg(totalitem) from

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

group by OrderID)

Output:



avg(totalitem)
65.01530612244898

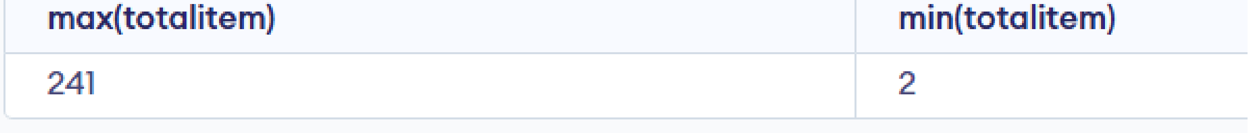
30. Find the highest and lowest order totals:

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

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

group by OrderID)

Output:



max(totalitem)	min(totalitem)
241	2

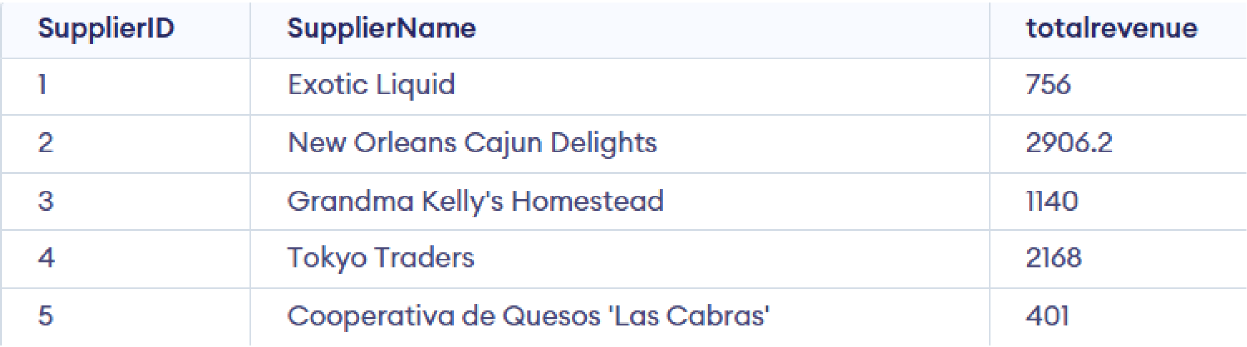
31. Find the total revenue generated by each supplier:

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:



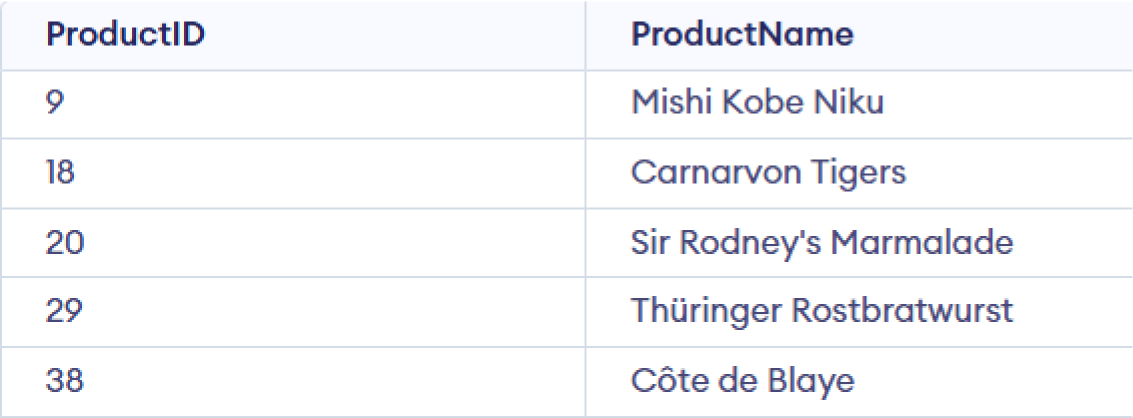
SupplierID	SupplierName	totalrevenue
1	Exotic Liquid	756
2	New Orleans Cajun Delights	2906.2
3	Grandma Kelly's Homestead	1140
4	Tokyo Traders	2168
5	Cooperativa de Quesos 'Las Cabras'	401

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

Query: select ProductID,ProductName from Products

where Price> 50

Output:



ProductID	ProductName
9	Mishi Kobe Niku
18	Carnarvon Tigers
20	Sir Rodney's Marmalade
29	Thüringer Rostbratwurst
38	Côte de Blaye

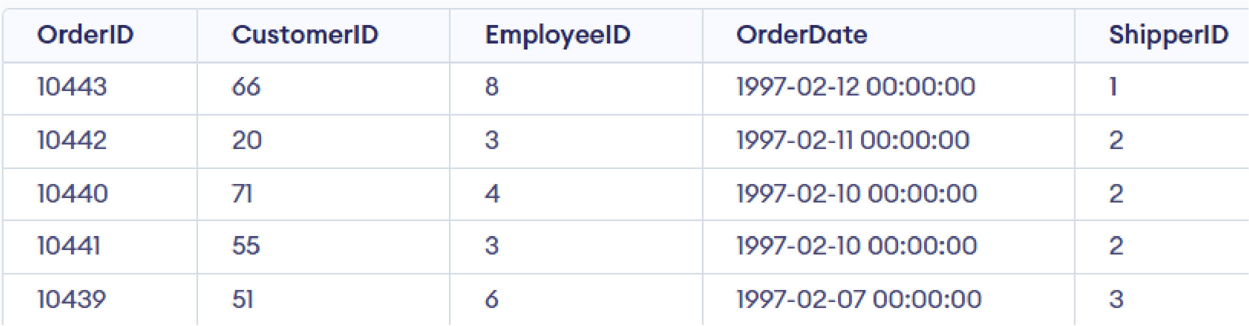
33. List all orders placed in the last 30 days:

Query: SELECT * FROM Orders

order by OrderDate DESC

limit 30;

Output:



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10443	66	8	1997-02-12 00:00:00	1
10442	20	3	1997-02-11 00:00:00	2
10440	71	4	1997-02-10 00:00:00	2
10441	55	3	1997-02-10 00:00:00	2
10439	51	6	1997-02-07 00:00:00	3

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

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

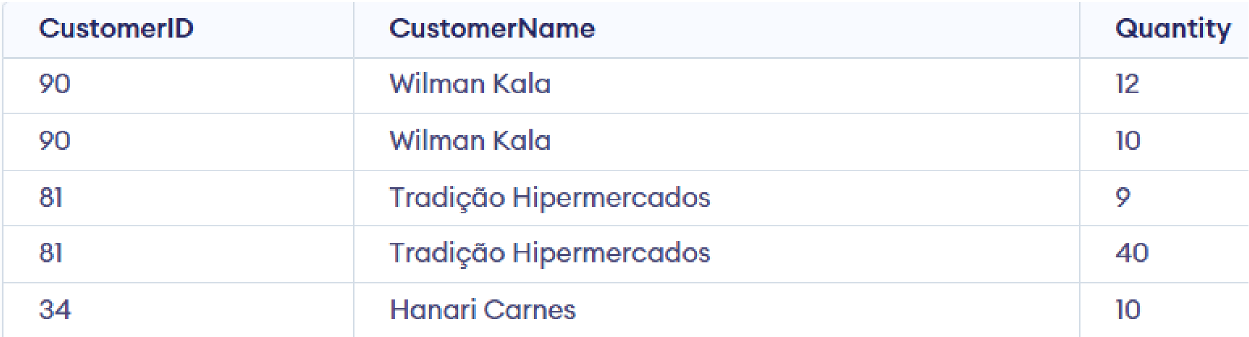
FROM Customers as c

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

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

where od.Quantity>5

Output:



CustomerID	CustomerName	Quantity
90	Wilman Kala	12
90	Wilman Kala	10
81	Tradição Hipermercados	9
81	Tradição Hipermercados	40
34	Hanari Carnes	10

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

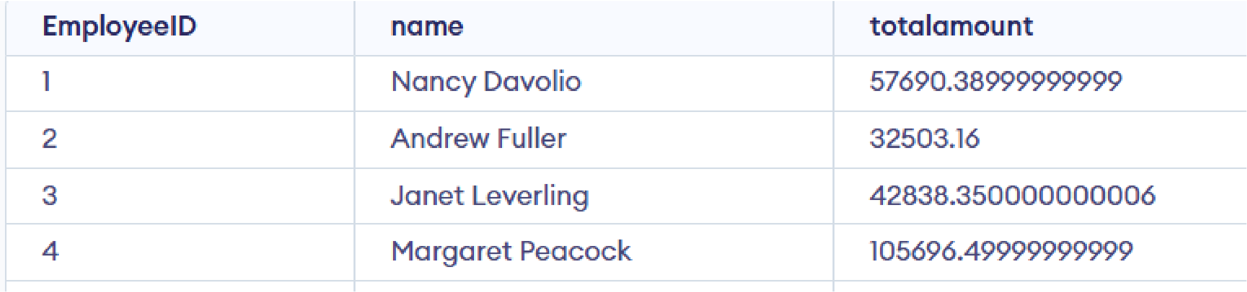
from Employees as e

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:



EmployeeID	name	totalamount
1	Nancy Davolio	57690.38999999999
2	Andrew Fuller	32503.16
3	Janet Leverling	42838.350000000006
4	Margaret Peacock	105696.49999999999

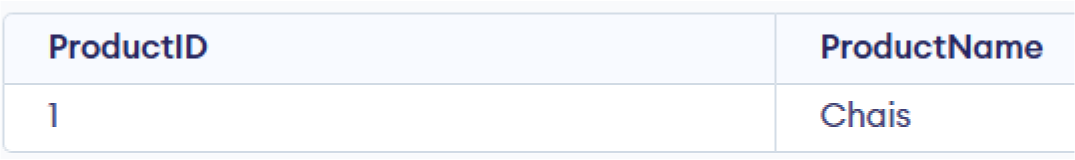
36. List all products supplied by 'Supplier A':

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:



ProductID	ProductName
1	Chais

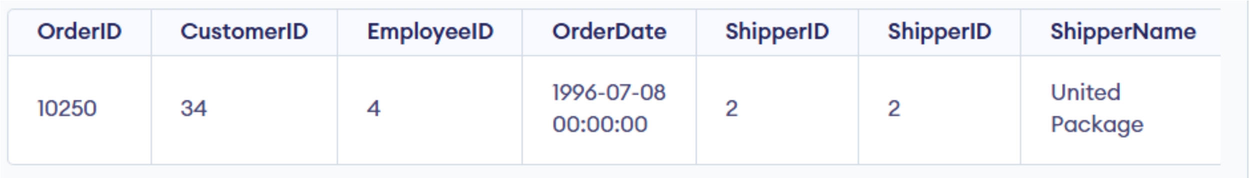
37. List all orders shipped by 'Shipper B':

Query:
select * from Orders as o

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

group by s.ShipperID

having s.ShipperID = 2
Output:



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	ShipperID	ShipperName
10250	34	4	1996-07-08 00:00:00	2	2	United Package

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

Query:

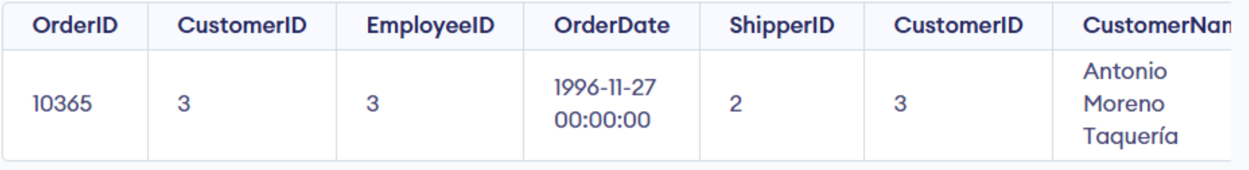
select * from Orders as o

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

group by c.CustomerID

having c.CustomerID = 3

Output:



OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	CustomerID	CustomerNan
10365	3	3	1996-11-27 00:00:00	2	3	Antonio Moreno Taquería

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

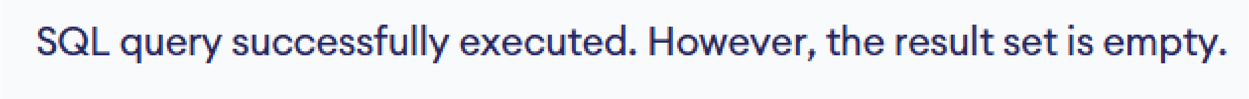
Query: select * from Products as p

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

group by c.CategoryName

having c.CategoryName = "Electronics"

Output:



SQL query successfully executed. However, the result set is empty.

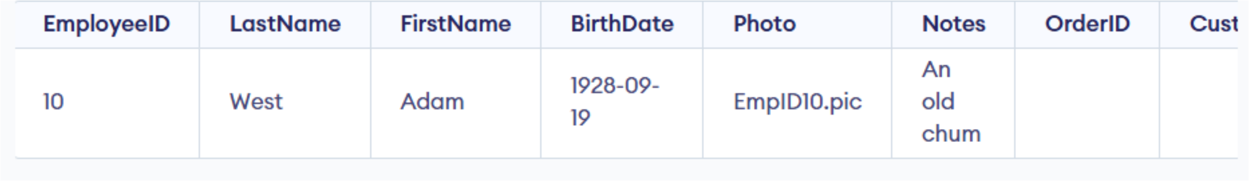
40. List all employees who have not handled any orders:

Query: select * from Employees as e

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

WHERE o.OrderID IS NULL;

Output:



EmployeeID	LastName	FirstName	BirthDate	Photo	Notes	OrderID	Cust
10	West	Adam	1928-09-19	EmpID10.pic	An old chum		

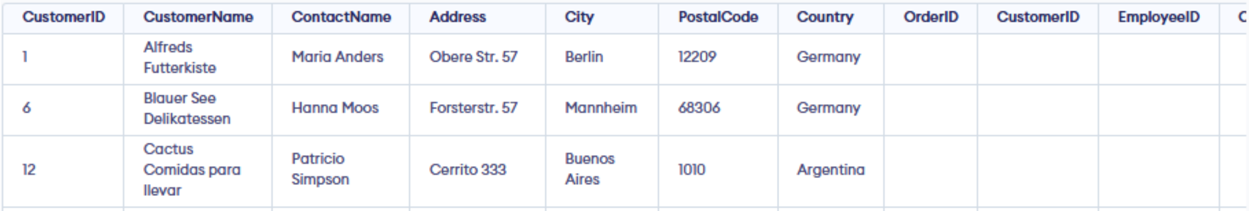
41. List all customers who have not placed any orders:

Query: select * from Customers as c

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

WHERE o.OrderID IS NULL;

Output:



CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country	OrderID	CustomerID	EmployeeID	C
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany				
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany				
12	Cactus Comidas para llevar	Patricio Simpson	Cerrito 333	Buenos Aires	1010	Argentina				

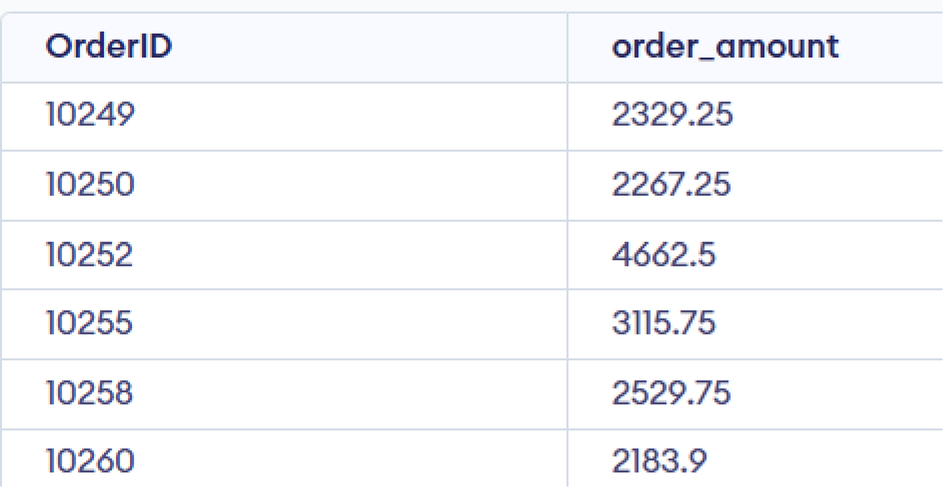
42. List all orders where the total order amount is greater than the average order amount:

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:

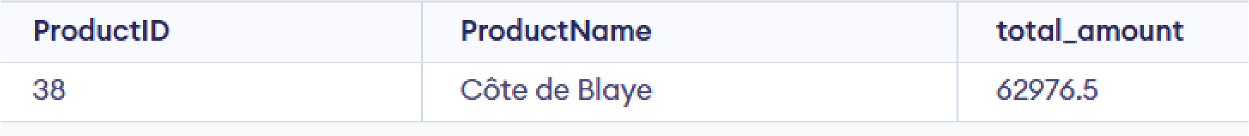


OrderID	order_amount
10249	2329.25
10250	2267.25
10252	4662.5
10255	3115.75
10258	2529.75
10260	2183.9

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:

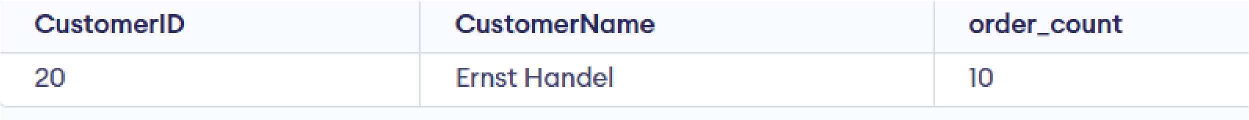


ProductID	ProductName	total_amount
38	Côte de Blaye	62976.5

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:

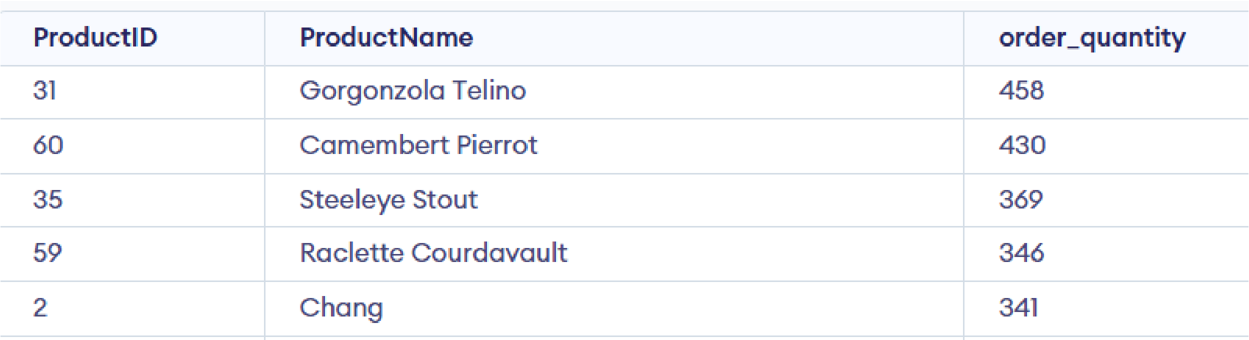


CustomerID	CustomerName	order_count
20	Ernst Handel	10

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:



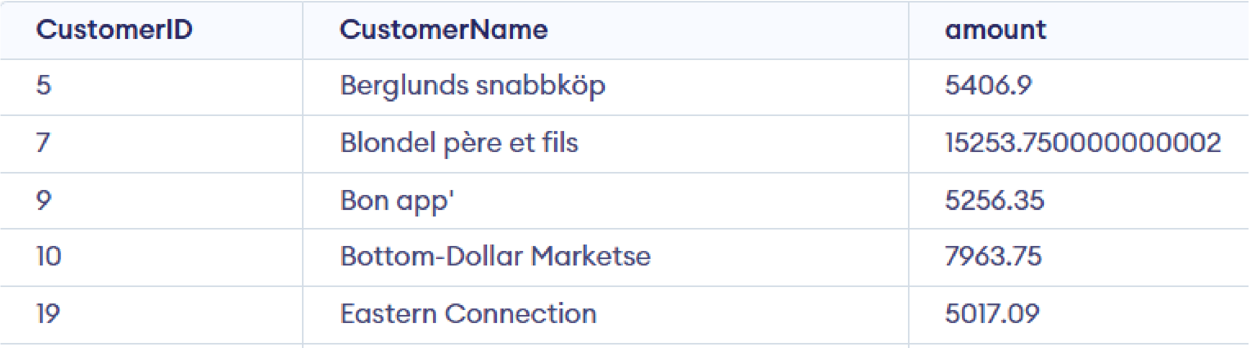
ProductID	ProductName	order_quantity
31	Gorgonzola Telino	458
60	Camembert Pierrot	430
35	Steeleye Stout	369
59	Raclette Courdavault	346
2	Chang	341

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:



CustomerID	CustomerName	amount
5	Berglunds snabbköp	5406.9
7	Blondel père et fils	15253.750000000002
9	Bon app'	5256.35
10	Bottom-Dollar Marketse	7963.75
19	Eastern Connection	5017.09

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

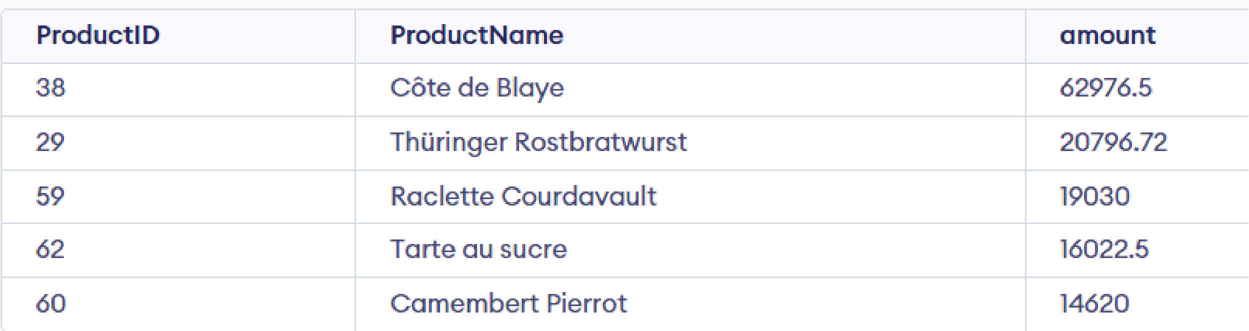
join OrderDetails as od on p.ProductID = od.ProductID

group by p.ProductID, p.ProductName

order by amount desc

limit 5

Output:



ProductID	ProductName	amount
38	Côte de Blaye	62976.5
29	Thüringer Rostbratwurst	20796.72
59	Raclette Courdavault	19030
62	Tarte au sucre	16022.5
60	Camembert Pierrot	14620

48. List all products that have never been ordered:

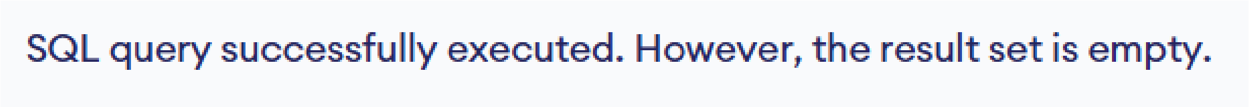
Query: select p.ProductID,p.ProductName from Products as p

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

group by p.ProductID,p.ProductName

having od.OrderID is null

Output:



SQL query successfully executed. However, the result set is empty.

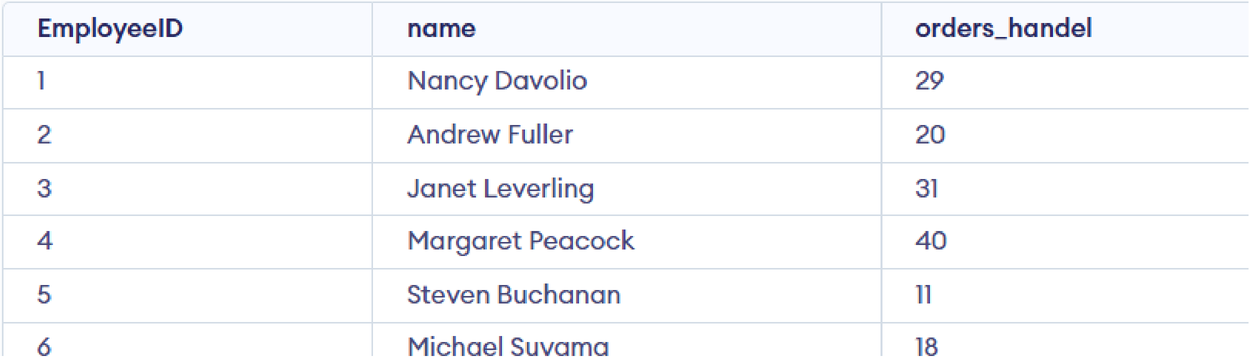
49. List all employees along with the number of orders they have handled:

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:



EmployeeID	name	orders_handel
1	Nancy Davolio	29
2	Andrew Fuller	20
3	Janet Leverling	31
4	Margaret Peacock	40
5	Steven Buchanan	11
6	Michael Suyama	18

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

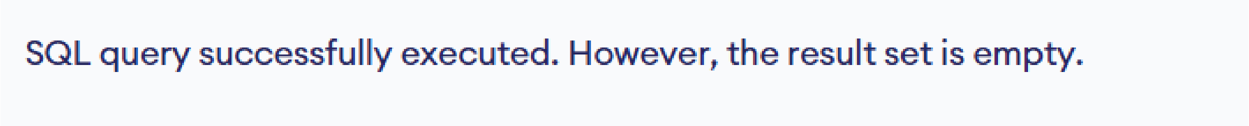
Query: select s.SupplierID, s.SupplierName from Suppliers as s

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:



SQL query successfully executed. However, the result set is empty.

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

Query:

with orderCount as

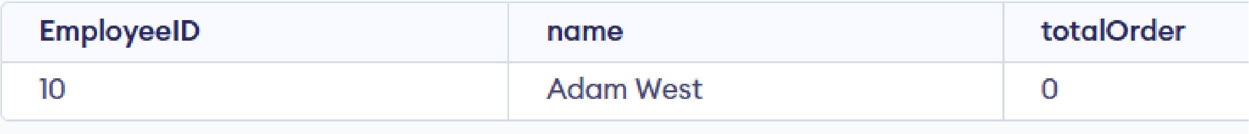
(select e.EmployeeID, e.FirstName || ' ' || e.LastName as name,count(o.OrderID) as totalOrder from employees as e

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:

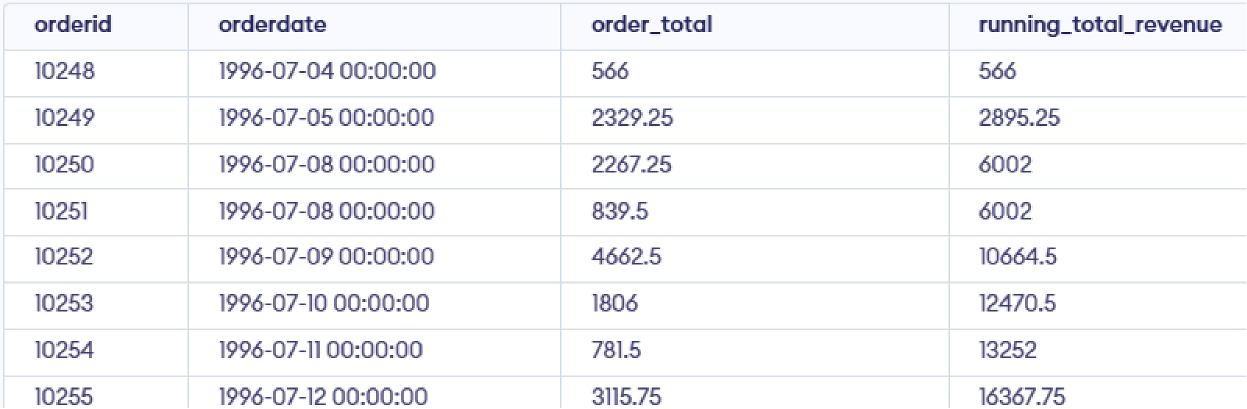


EmployeeID	name	totalOrder
10	Adam West	0

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:



orderid	orderdate	order_total	running_total_revenue
10248	1996-07-04 00:00:00	566	566
10249	1996-07-05 00:00:00	2329.25	2895.25
10250	1996-07-08 00:00:00	2267.25	6002
10251	1996-07-08 00:00:00	839.5	6002
10252	1996-07-09 00:00:00	4662.5	10664.5
10253	1996-07-10 00:00:00	1806	12470.5
10254	1996-07-11 00:00:00	781.5	13252
10255	1996-07-12 00:00:00	3115.75	16367.75

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

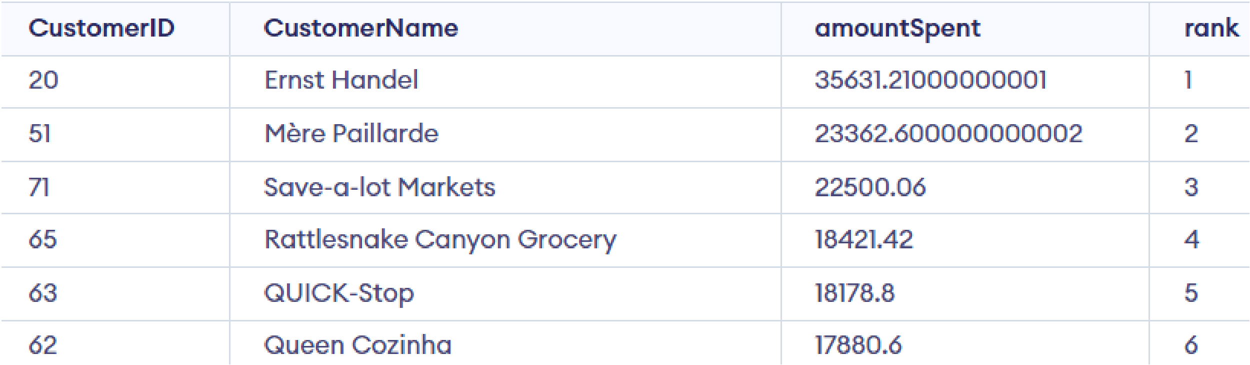
from Customers as c join Orders as o on c.CustomerID = o.CustomerID

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

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

group by c.CustomerID

Output:



CustomerID	CustomerName	amountSpent	rank
20	Ernst Handel	35631.21000000001	1
51	Mère Paillarde	23362.600000000002	2
71	Save-a-lot Markets	22500.06	3
65	Rattlesnake Canyon Grocery	18421.42	4
63	QUICK-Stop	18178.8	5
62	Queen Cozinha	17880.6	6

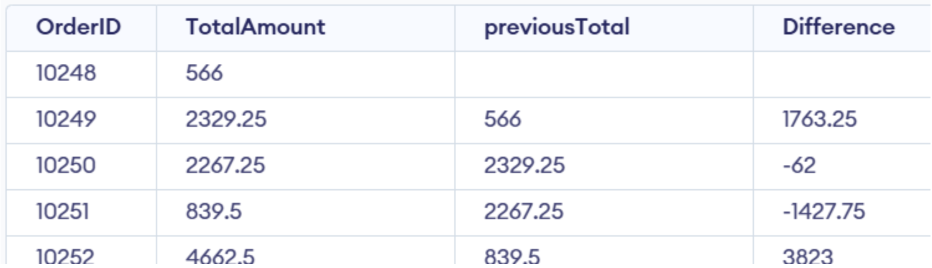
54. Calculate the difference in order totals between consecutive orders:

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:



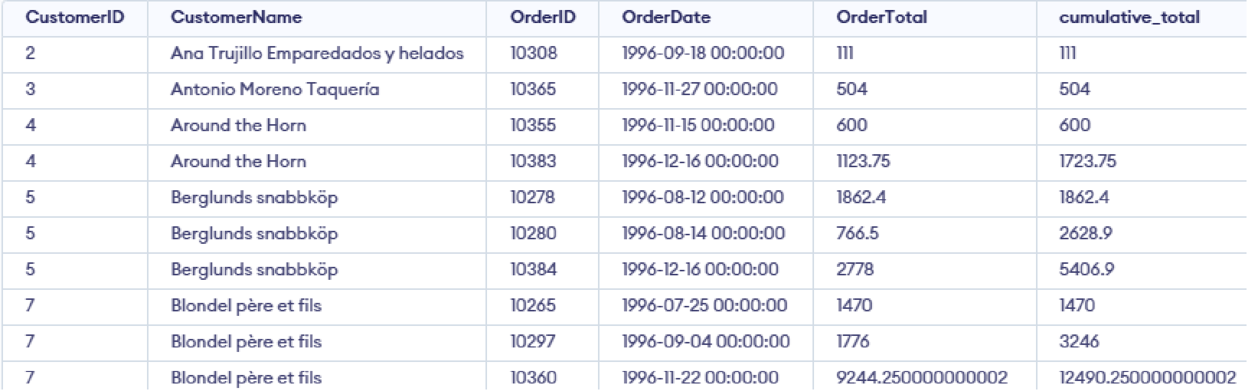
OrderID	TotalAmount	previousTotal	Difference
10248	566		
10249	2329.25	566	1763.25
10250	2267.25	2329.25	-62
10251	839.5	2267.25	-1427.75
10252	4662.5	839.5	3823

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:

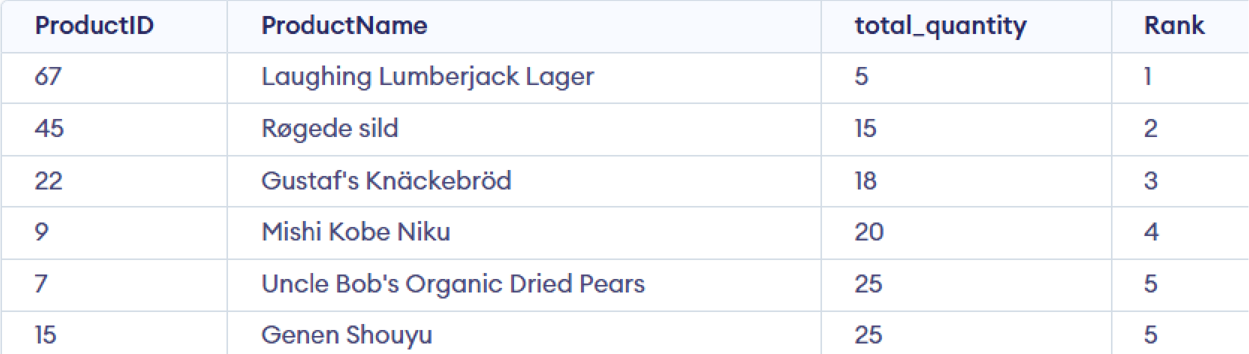


CustomerID	CustomerName	OrderID	OrderDate	OrderTotal	cumulative_total
2	Ana Trujillo Emparedados y helados	10308	1996-09-18 00:00:00	111	111
3	Antonio Moreno Taquería	10365	1996-11-27 00:00:00	504	504
4	Around the Horn	10355	1996-11-15 00:00:00	600	600
4	Around the Horn	10383	1996-12-16 00:00:00	1123.75	1723.75
5	Berglunds snabbköp	10278	1996-08-12 00:00:00	1862.4	1862.4
5	Berglunds snabbköp	10280	1996-08-14 00:00:00	766.5	2628.9
5	Berglunds snabbköp	10384	1996-12-16 00:00:00	2778	5406.9
7	Blondel père et fils	10265	1996-07-25 00:00:00	1470	1470
7	Blondel père et fils	10297	1996-09-04 00:00:00	1776	3246
7	Blondel père et fils	10360	1996-11-22 00:00:00	9244.250000000002	12490.250000000002

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:



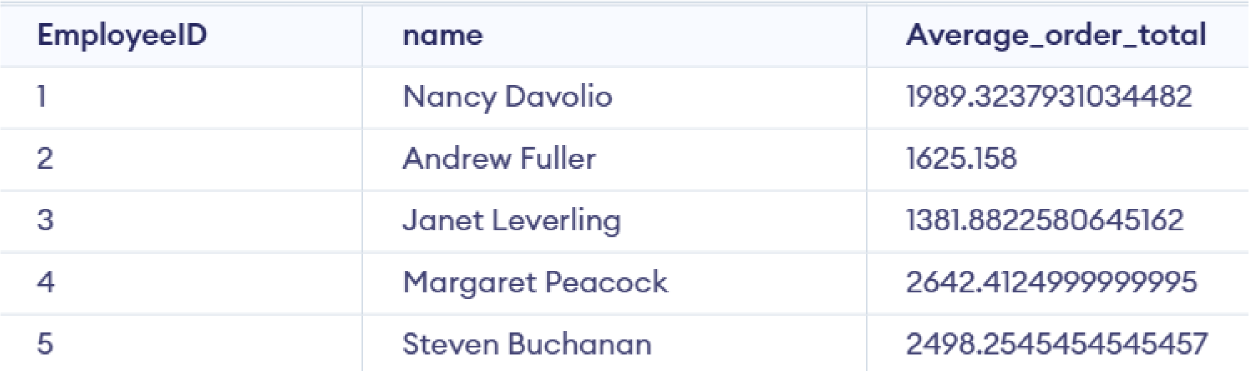
ProductID	ProductName	total_quantity	Rank
67	Laughing Lumberjack Lager	5	1
45	Røgede sild	15	2
22	Gustaf's Knäckebröd	18	3
9	Mishi Kobe Niku	20	4
7	Uncle Bob's Organic Dried Pears	25	5
15	Genen Shouyu	25	5

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:

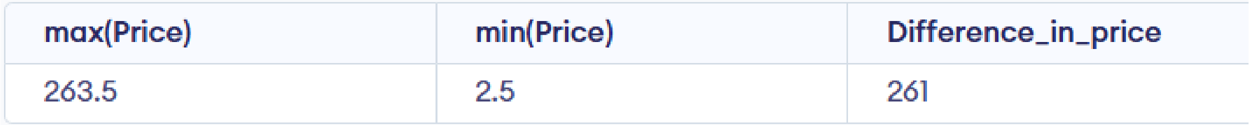


EmployeeID	name	Average_order_total
1	Nancy Davolio	1989.3237931034482
2	Andrew Fuller	1625.158
3	Janet Leverling	1381.8822580645162
4	Margaret Peacock	2642.4124999999995
5	Steven Buchanan	2498.2545454545457

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:



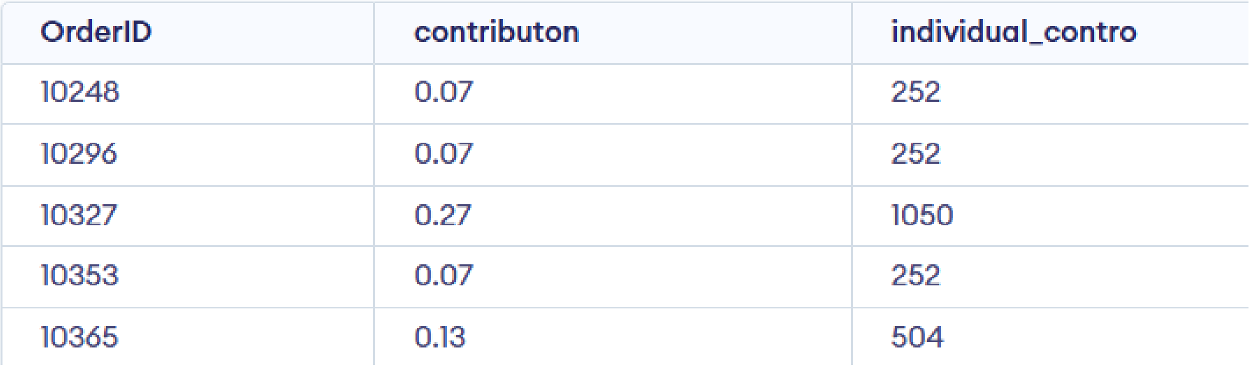
max(Price)	min(Price)	Difference_in_price
263.5	2.5	261

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.Quantityp.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:



OrderID	contributon	individual_contro
10248	0.07	252
10296	0.07	252
10327	0.27	1050
10353	0.07	252
10365	0.13	504

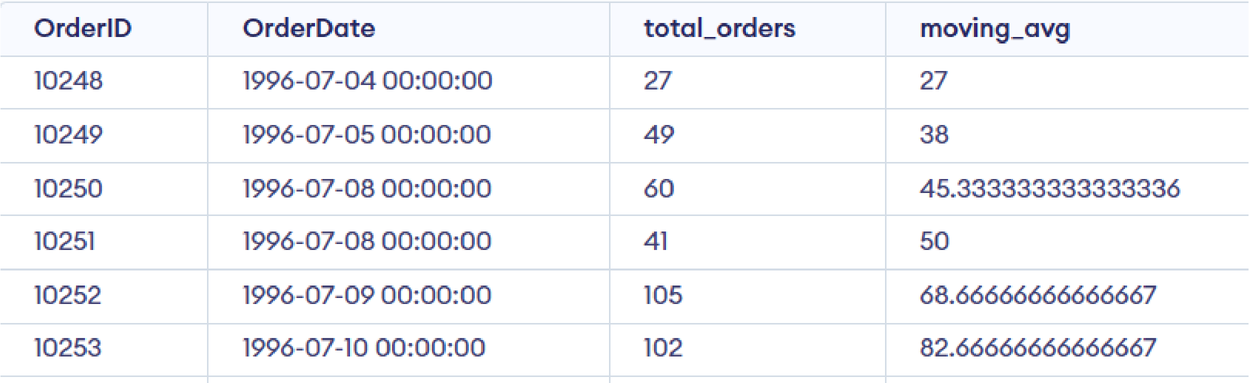
60. Calculate the moving average of order totals over a 3-order window:

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:

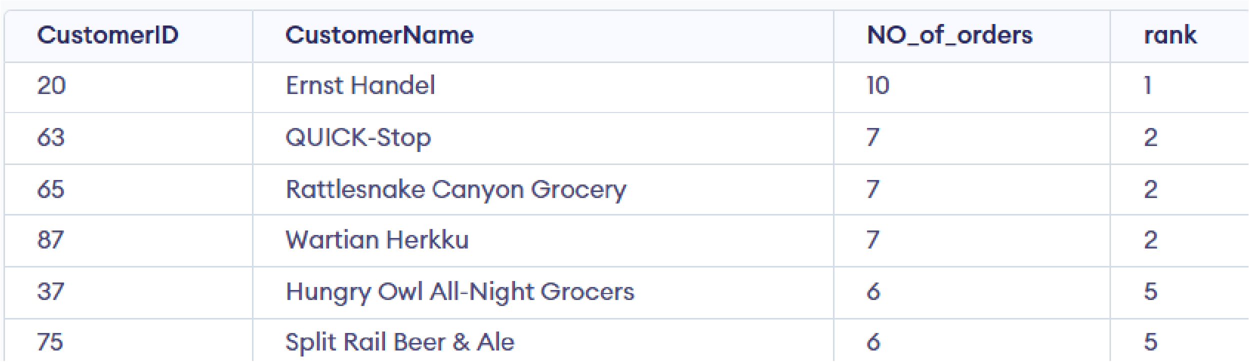


OrderID	OrderDate	total_orders	moving_avg
10248	1996-07-04 00:00:00	27	27
10249	1996-07-05 00:00:00	49	38
10250	1996-07-08 00:00:00	60	45.333333333333336
10251	1996-07-08 00:00:00	41	50
10252	1996-07-09 00:00:00	105	68.66666666666667
10253	1996-07-10 00:00:00	102	82.66666666666667

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:



CustomerID	CustomerName	NO_of_orders	rank
20	Ernst Handel	10	1
63	QUICK-Stop	7	2
65	Rattlesnake Canyon Grocery	7	2
87	Wartian Herkku	7	2
37	Hungry Owl All-Night Grocers	6	5
75	Split Rail Beer & Ale	6	5

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

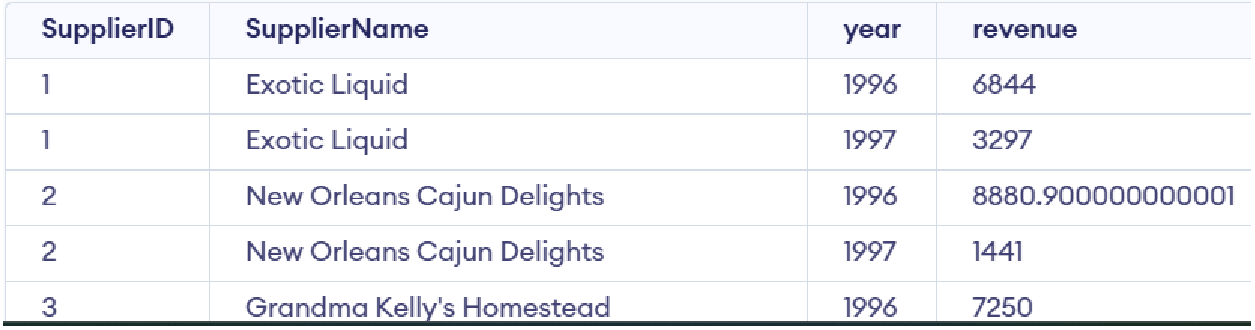
Query: select s.SupplierID,s.SupplierName,strftime("%Y",o.OrderDate) as year, sum(od.Quantity*p.price) as revenue from Suppliers s

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:



SupplierID	SupplierName	year	revenue
1	Exotic Liquid	1996	6844
1	Exotic Liquid	1997	3297
2	New Orleans Cajun Delights	1996	8880.900000000001
2	New Orleans Cajun Delights	1997	1441
3	Grandma Kelly's Homestead	1996	7250

63. List the average order amount by employee:

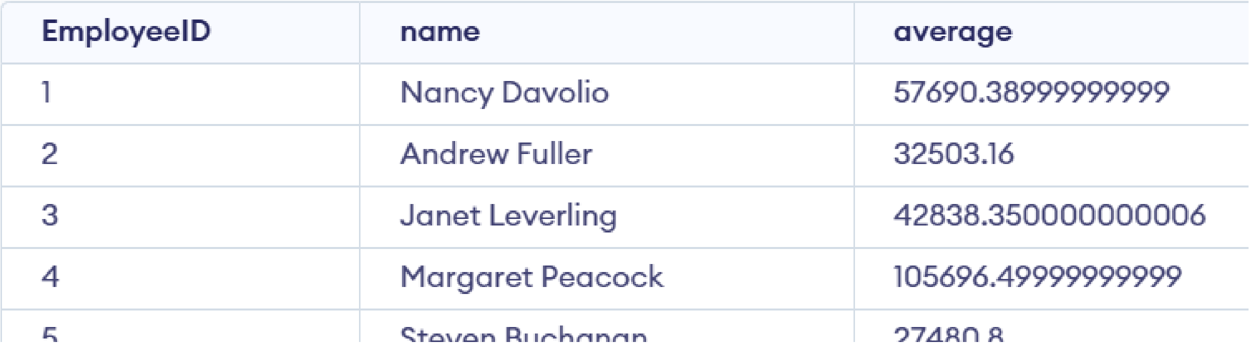
Query:

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:



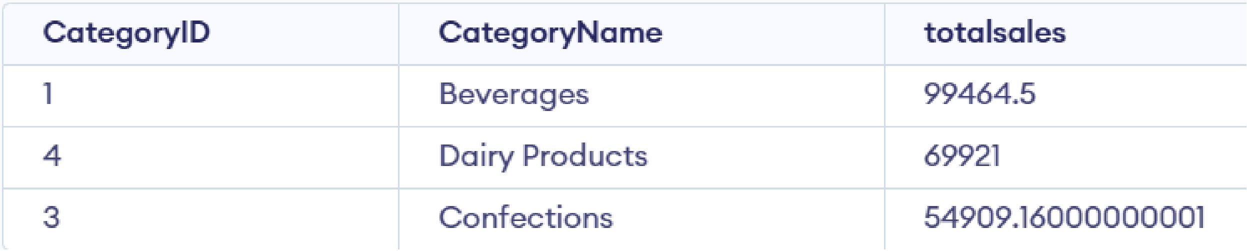
EmployeeID	name	average
1	Nancy Davolio	57690.38999999999
2	Andrew Fuller	32503.16
3	Janet Leverling	42838.350000000006
4	Margaret Peacock	105696.49999999999
5	Steven Buchanan	27480.8

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:



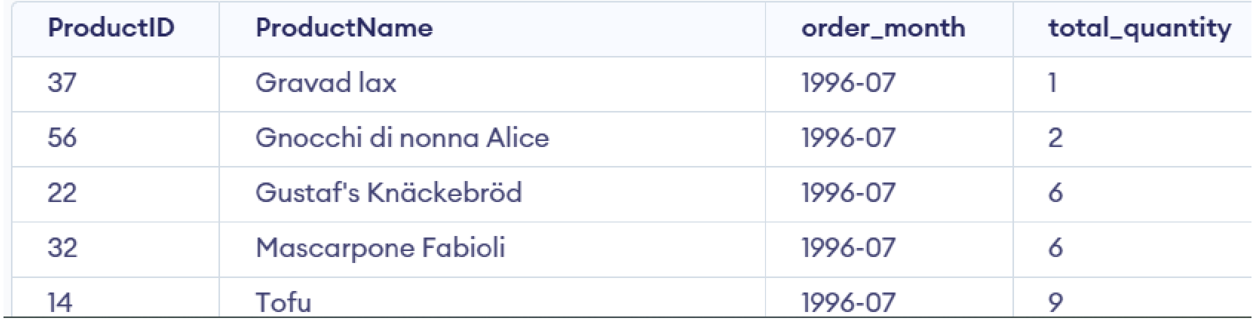
CategoryID	CategoryName	totalsales
1	Beverages	99464.5
4	Dairy Products	69921
3	Confections	54909.16000000001

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:



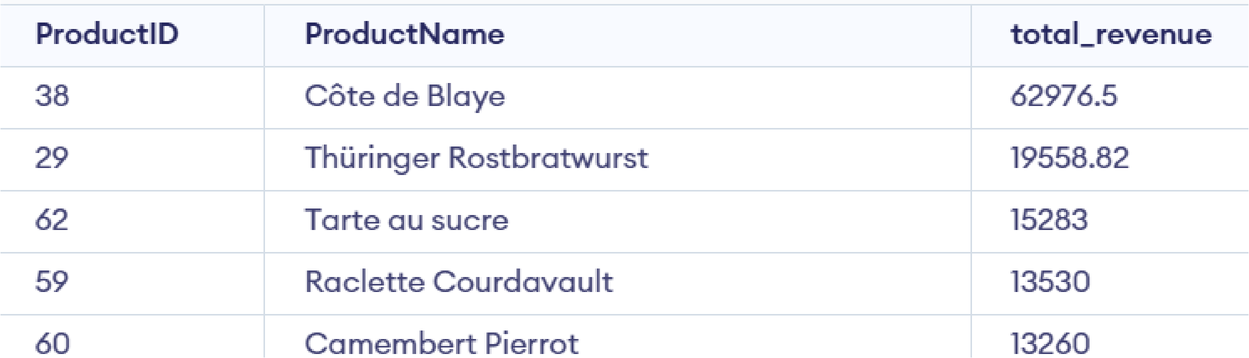
ProductID	ProductName	order_month	total_quantity
37	Gravad lax	1996-07	1
56	Gnocchi di nonna Alice	1996-07	2
22	Gustaf's Knäckebröd	1996-07	6
32	Mascarpone Fabioli	1996-07	6
14	Tofu	1996-07	9

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:



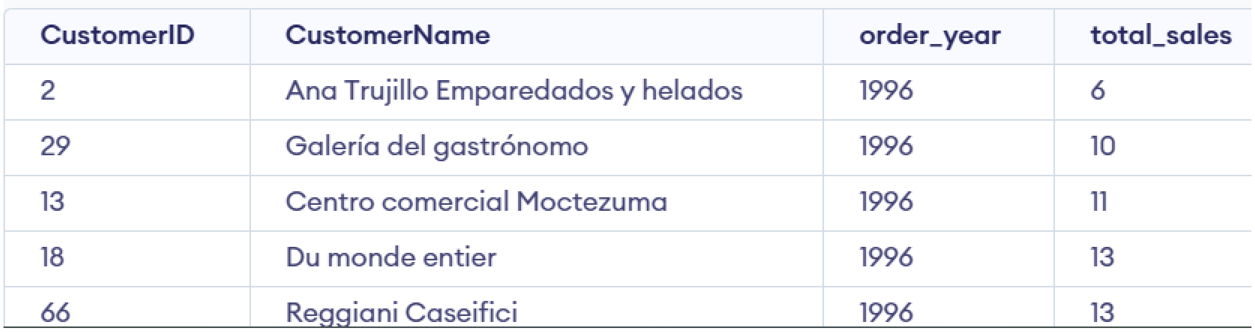
ProductID	ProductName	total_revenue
38	Côte de Blaye	62976.5
29	Thüringer Rostbratwurst	19558.82
62	Tarte au sucre	15283
59	Raclette Courdavault	13530
60	Camembert Pierrot	13260

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:



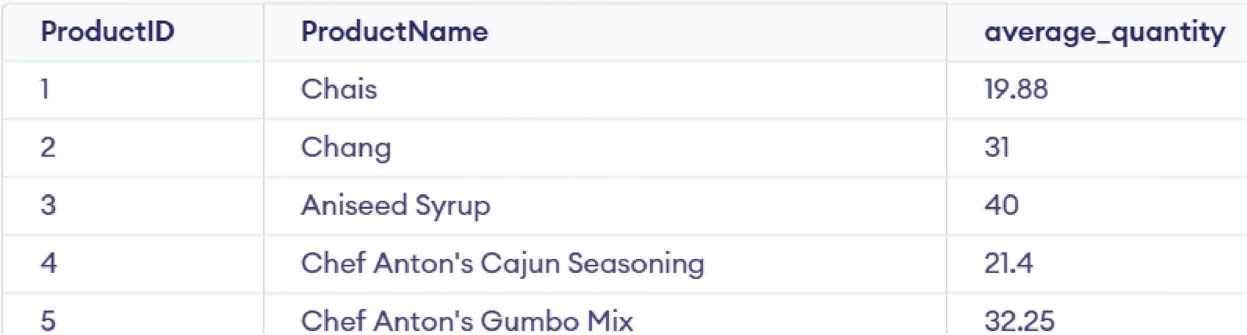
CustomerID	CustomerName	order_year	total_sales
2	Ana Trujillo Emparedados y helados	1996	6
29	Galería del gastrónomo	1996	10
13	Centro comercial Moctezuma	1996	11
18	Du monde entier	1996	13
66	Reggiani Caseifici	1996	13

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
```

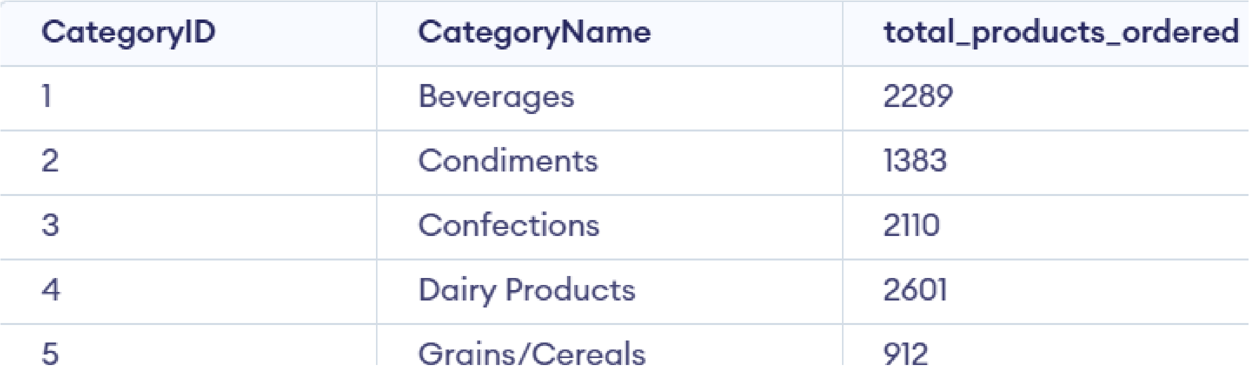
order by p.productid;
Output:



ProductID	ProductName	average_quantity
1	Chais	19.88
2	Chang	31
3	Aniseed Syrup	40
4	Chef Anton's Cajun Seasoning	21.4
5	Chef Anton's Gumbo Mix	32.25

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:

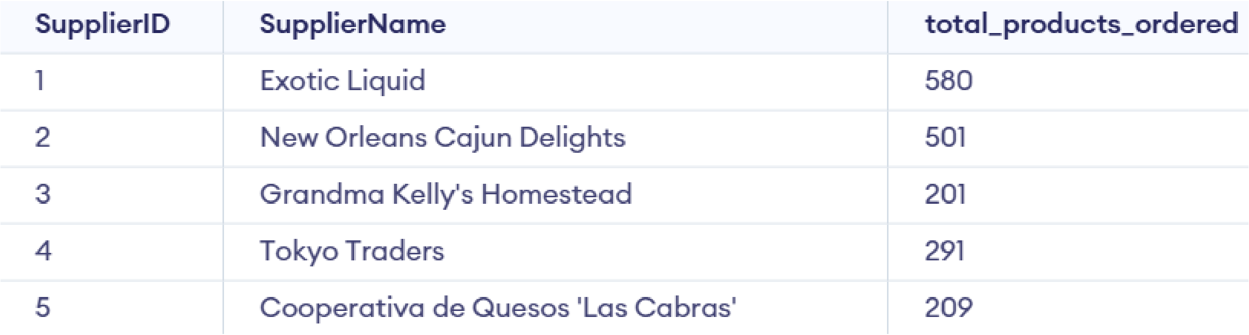


CategoryID	CategoryName	total_products_ordered
1	Beverages	2289
2	Condiments	1383
3	Confections	2110
4	Dairy Products	2601
5	Grains/Cereals	912

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:



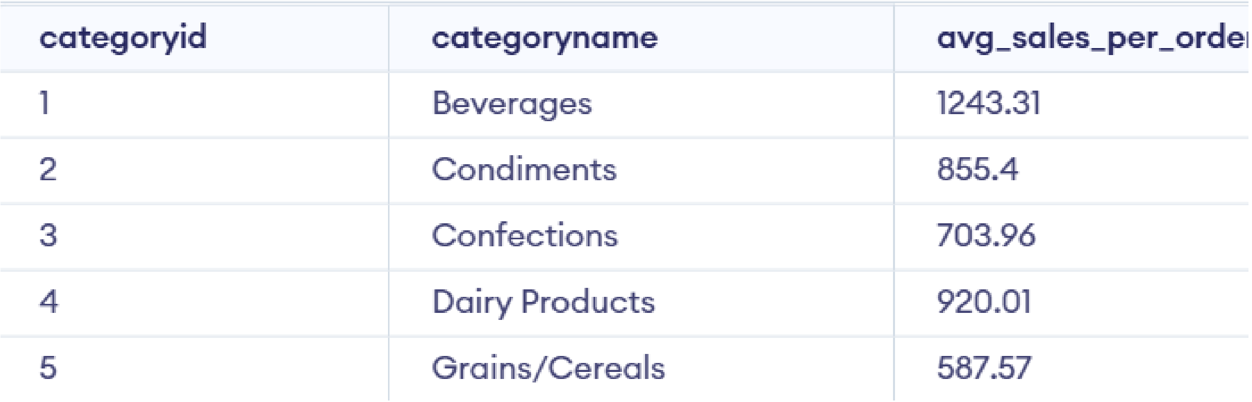
SupplierID	SupplierName	total_products_ordered
1	Exotic Liquid	580
2	New Orleans Cajun Delights	501
3	Grandma Kelly's Homestead	201
4	Tokyo Traders	291
5	Cooperativa de Quesos 'Las Cabras'	209

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:

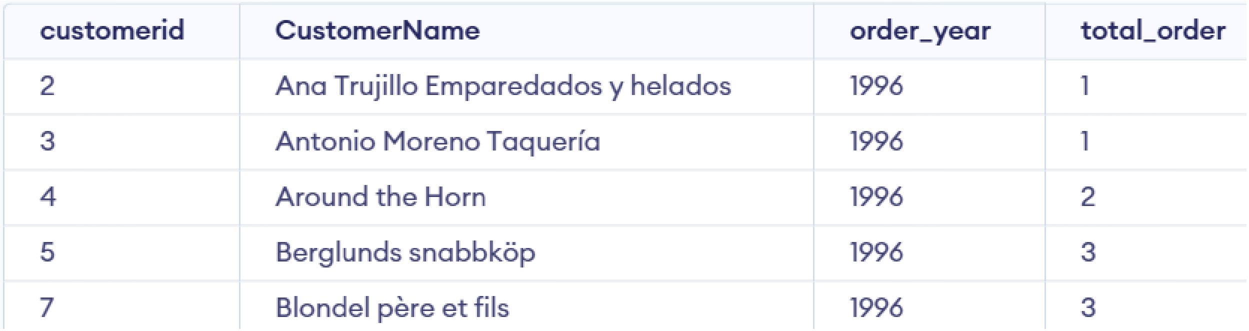


categoryid	categoryname	avg_sales_per_orde
1	Beverages	1243.31
2	Condiments	855.4
3	Confections	703.96
4	Dairy Products	920.01
5	Grains/Cereals	587.57

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)

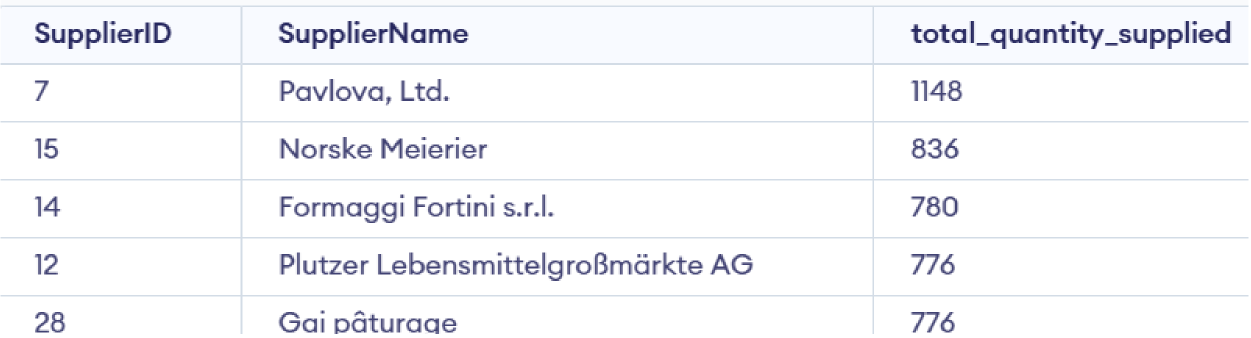
select customerid,CustomerName,order_year,total_order from total_no_orders
Output:



customerid	CustomerName	order_year	total_order
2	Ana Trujillo Emparedados y helados	1996	1
3	Antonio Moreno Taquería	1996	1
4	Around the Horn	1996	2
5	Berglunds snabbköp	1996	3
7	Blondel père et fils	1996	3

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:



SupplierID	SupplierName	total_quantity_supplied
7	Pavlova, Ltd.	1148
15	Norske Meierier	836
14	Formaggi Fortini s.r.l.	780
12	Plutzer Lebensmittelgroßmärkte AG	776
28	Gai pâturage	776

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

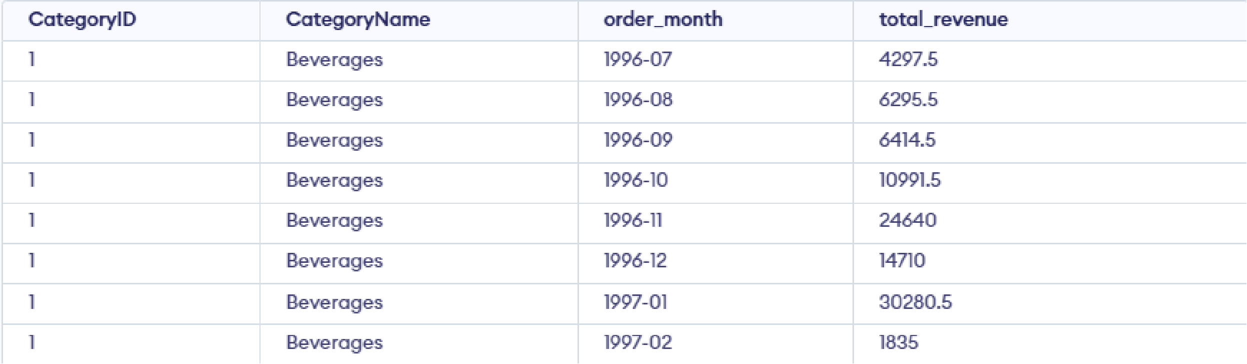
join products p on c.categoryid = p.categoryid

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:



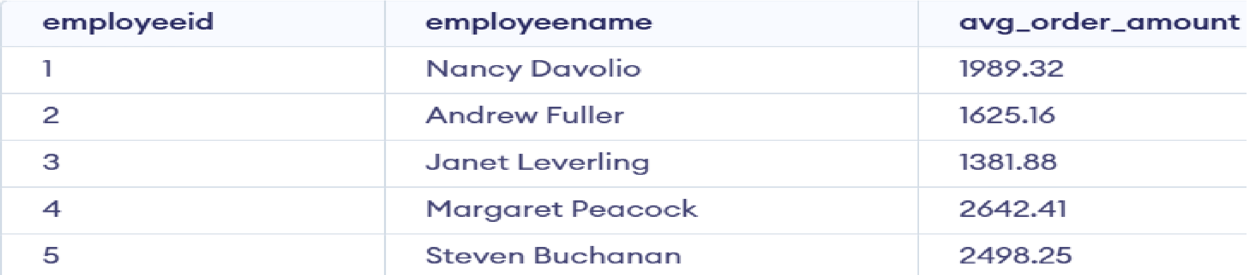
CategoryID	CategoryName	order_month	total_revenue
1	Beverages	1996-07	4297.5
1	Beverages	1996-08	6295.5
1	Beverages	1996-09	6414.5
1	Beverages	1996-10	10991.5
1	Beverages	1996-11	24640
1	Beverages	1996-12	14710
1	Beverages	1997-01	30280.5
1	Beverages	1997-02	1835

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:



employeeid	employeename	avg_order_amount
1	Nancy Davolio	1989.32
2	Andrew Fuller	1625.16
3	Janet Leverling	1381.88
4	Margaret Peacock	2642.41
5	Steven Buchanan	2498.25

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

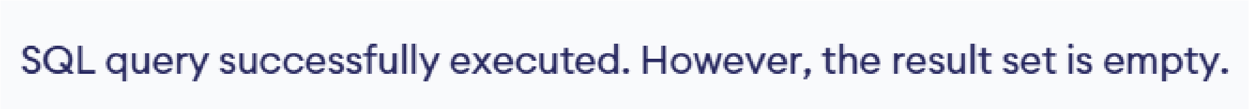
Query: select productid, count(distinct supplierid) as supplier_count

from products

group by productid

```
having supplier_count > 1;
```

Output:

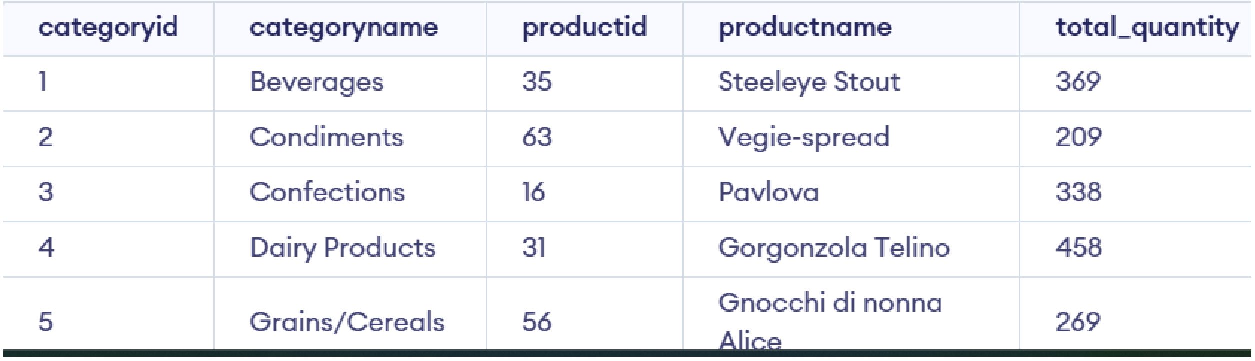


SQL query successfully executed. However, the result set is empty.

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:

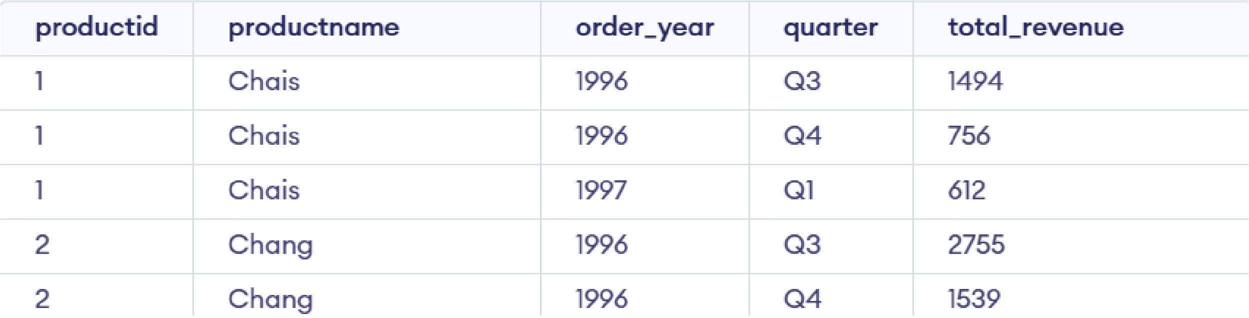


categoryid	categoryname	productid	productname	total_quantity
1	Beverages	35	Steeleye Stout	369
2	Condiments	63	Vegie-spread	209
3	Confections	16	Pavlova	338
4	Dairy Products	31	Gorgonzola Telino	458
5	Grains/Cereals	56	Gnocchi di nonna Alice	269

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:



productid	productname	order_year	quarter	total_revenue
1	Chais	1996	Q3	1494
1	Chais	1996	Q4	756
1	Chais	1997	Q1	612
2	Chang	1996	Q3	2755
2	Chang	1996	Q4	1539

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

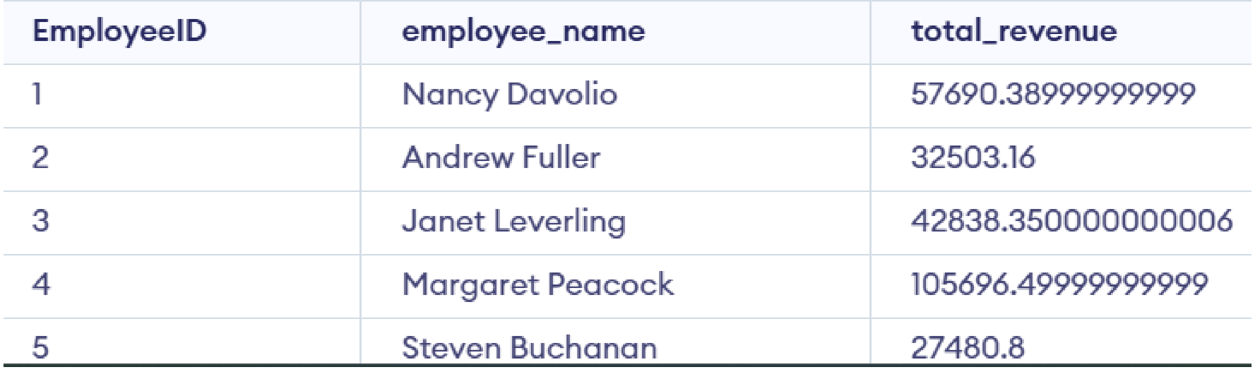
```
Query: with customer_quarters as (
select c.customerid,c.customername,
strftime('%Y', o.orderdate) as year,
        case
            when cast(strftime('%m', o.orderdate) as integer) between 1 and 3 then 'Q1'
            when cast(strftime('%m', o.orderdate) as integer) between 4 and 6 then 'Q2'
            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:

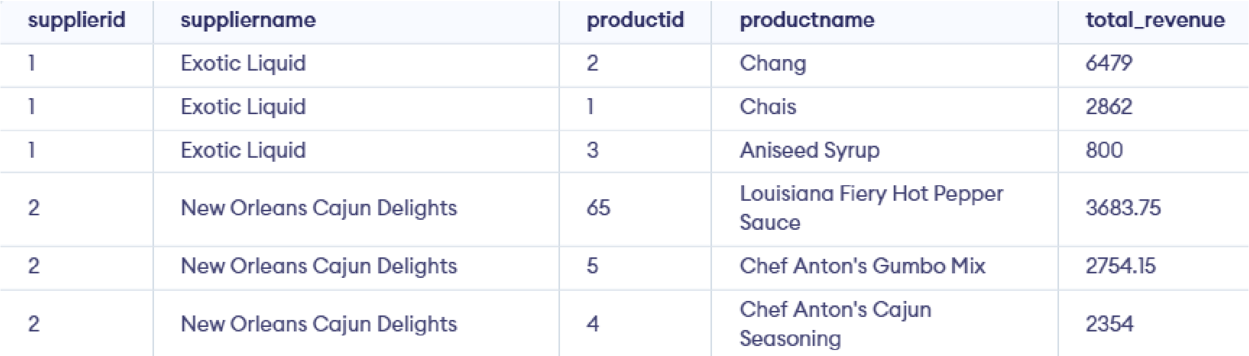


EmployeeID	employee_name	total_revenue
1	Nancy Davolio	57690.38999999999
2	Andrew Fuller	32503.16
3	Janet Leverling	42838.350000000006
4	Margaret Peacock	105696.49999999999
5	Steven Buchanan	27480.8

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:



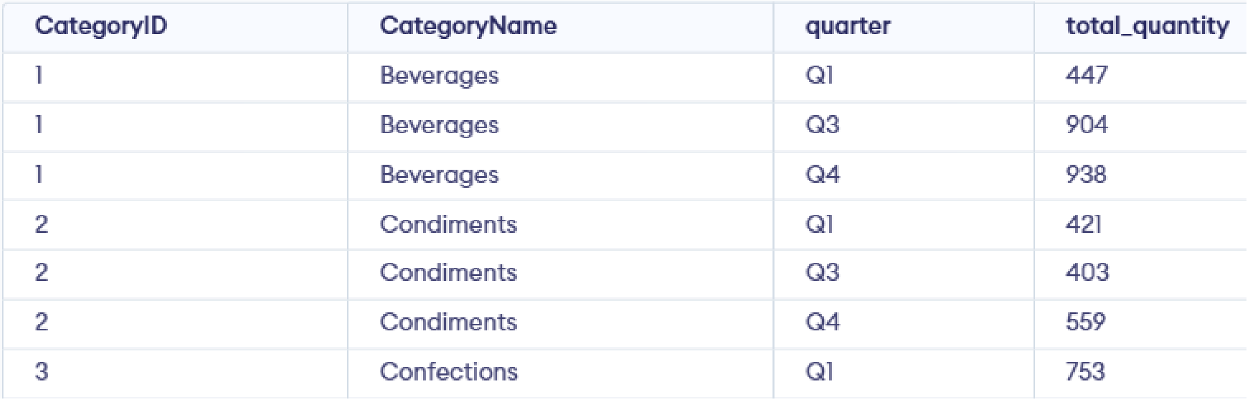
supplierid	suppliername	productid	productname	total_revenue
1	Exotic Liquid	2	Chang	6479
1	Exotic Liquid	1	Chais	2862
1	Exotic Liquid	3	Aniseed Syrup	800
2	New Orleans Cajun Delights	65	Louisiana Fiery Hot Pepper Sauce	3683.75
2	New Orleans Cajun Delights	5	Chef Anton's Gumbo Mix	2754.15
2	New Orleans Cajun Delights	4	Chef Anton's Cajun Seasoning	2354

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:

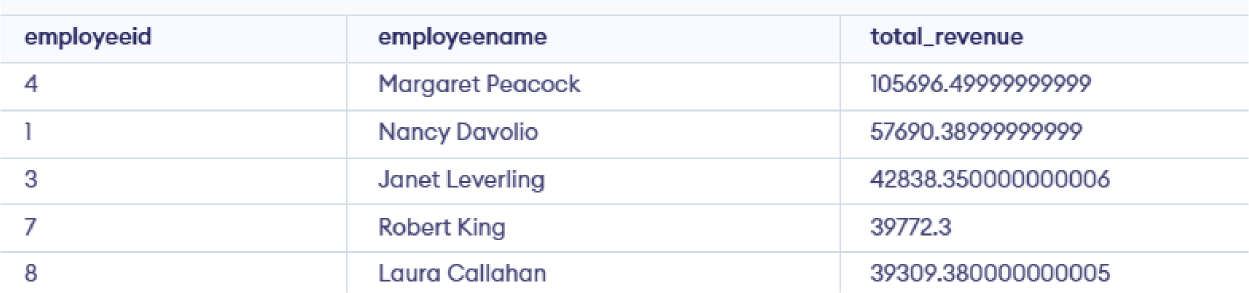


CategoryID	CategoryName	quarter	total_quantity
1	Beverages	Q1	447
1	Beverages	Q3	904
1	Beverages	Q4	938
2	Condiments	Q1	421
2	Condiments	Q3	403
2	Condiments	Q4	559
3	Confections	Q1	753

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:

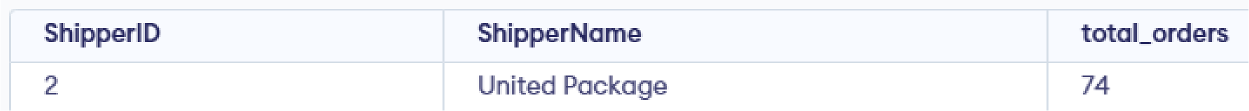


employeeid	employeename	total_revenue
4	Margaret Peacock	105696.49999999999
1	Nancy Davolio	57690.38999999999
3	Janet Leverling	42838.350000000006
7	Robert King	39772.3
8	Laura Callahan	39309.380000000005

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:

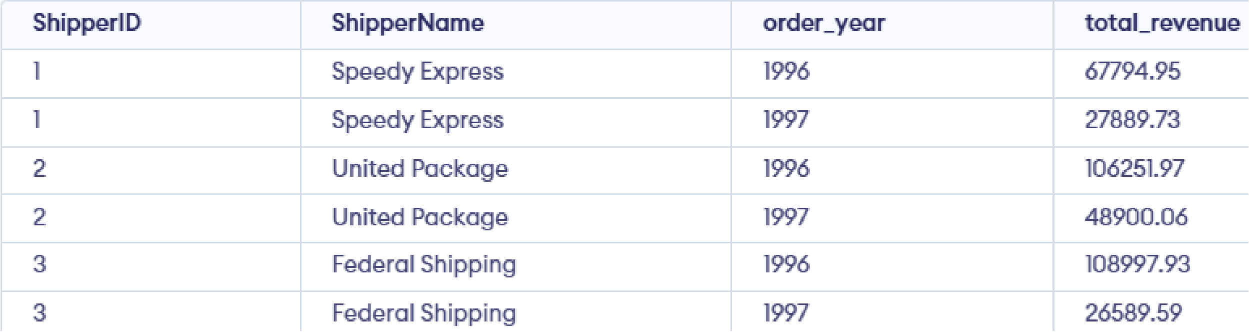


ShipperID	ShipperName	total_orders
2	United Package	74

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:



ShipperID	ShipperName	order_year	total_revenue
1	Speedy Express	1996	67794.95
1	Speedy Express	1997	27889.73
2	United Package	1996	106251.97
2	United Package	1997	48900.06
3	Federal Shipping	1996	108997.93
3	Federal Shipping	1997	26589.59

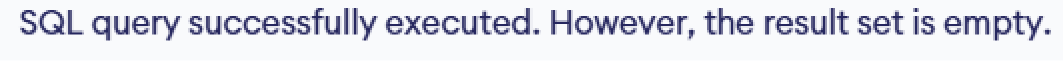
86. List all products that have never been ordered:

```
Query: select p.productid, p.productname
```

from products p

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

where od.productid is null;

Output: 

SQL query successfully executed. However, the result set is empty.

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

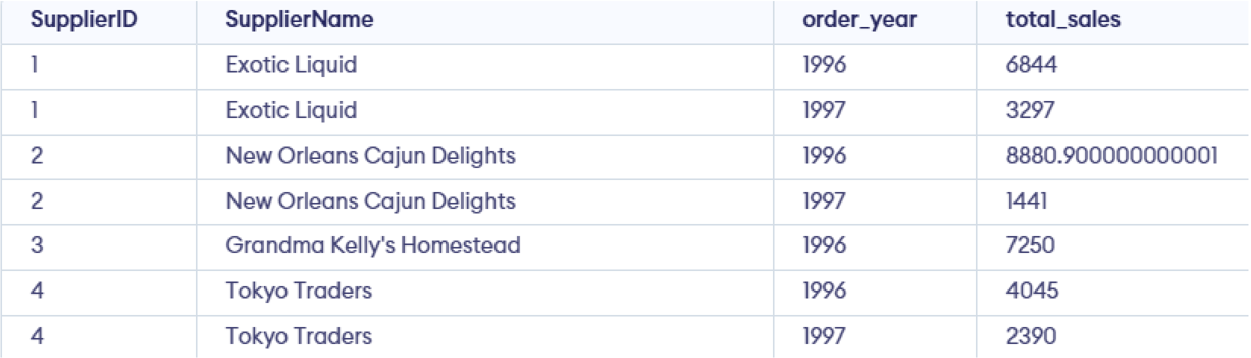
Query: select s.supplierid,s.suppliername,

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:

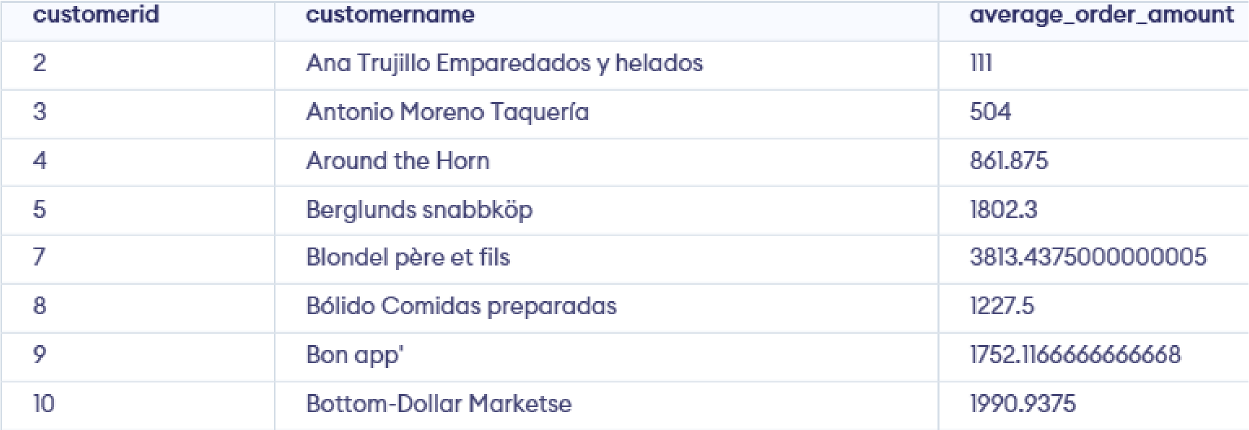


SupplierID	SupplierName	order_year	total_sales
1	Exotic Liquid	1996	6844
1	Exotic Liquid	1997	3297
2	New Orleans Cajun Delights	1996	8880.900000000001
2	New Orleans Cajun Delights	1997	1441
3	Grandma Kelly's Homestead	1996	7250
4	Tokyo Traders	1996	4045
4	Tokyo Traders	1997	2390

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:

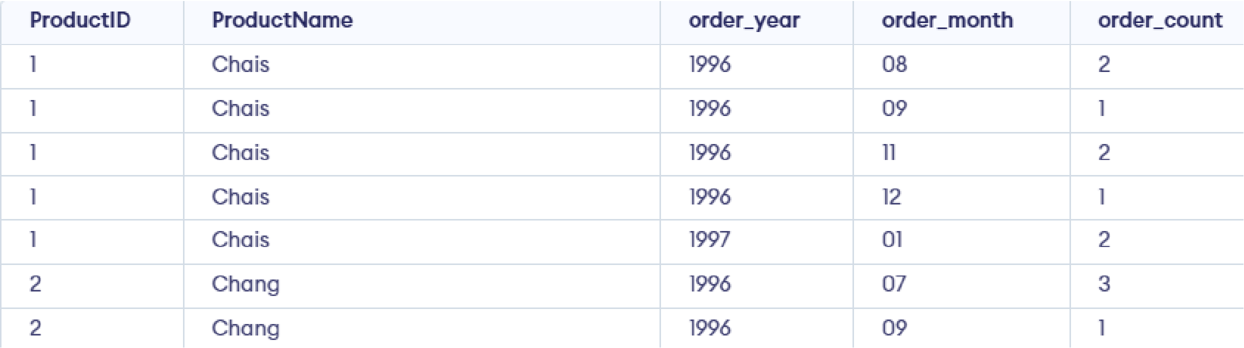


customerid	customername	average_order_amount
2	Ana Trujillo Emparedados y helados	111
3	Antonio Moreno Taquería	504
4	Around the Horn	861.875
5	Berglunds snabbköp	1802.3
7	Blondel père et fils	3813.4375000000005
8	Bólido Comidas preparadas	1227.5
9	Bon app'	1752.1166666666668
10	Bottom-Dollar Marketse	1990.9375

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:



ProductID	ProductName	order_year	order_month	order_count
1	Chais	1996	08	2
1	Chais	1996	09	1
1	Chais	1996	11	2
1	Chais	1996	12	1
1	Chais	1997	01	2
2	Chang	1996	07	3
2	Chang	1996	09	1

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

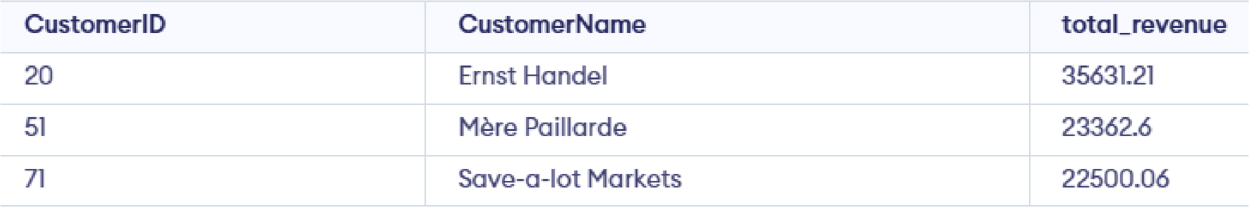
from customers c

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:

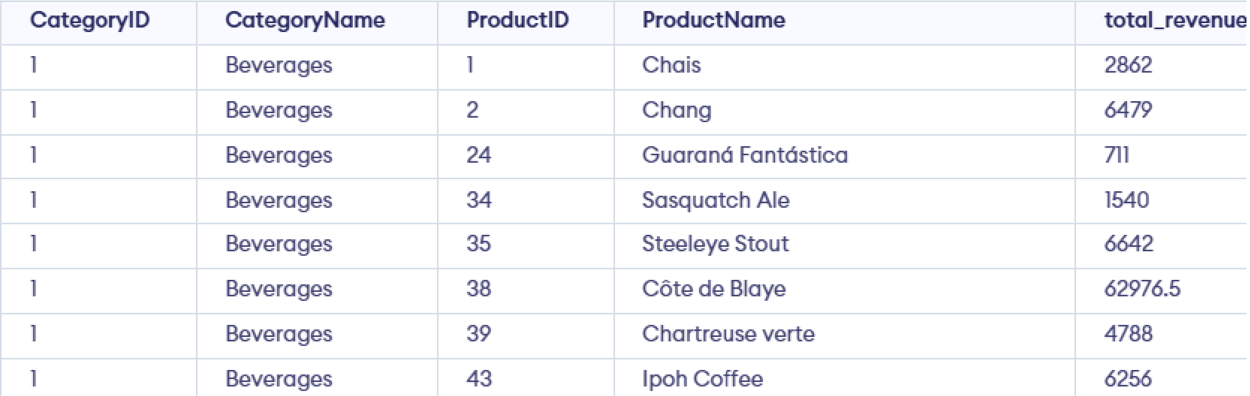


CustomerID	CustomerName	total_revenue
20	Ernst Handel	35631.21
51	Mère Paillarde	23362.6
71	Save-a-lot Markets	22500.06

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:

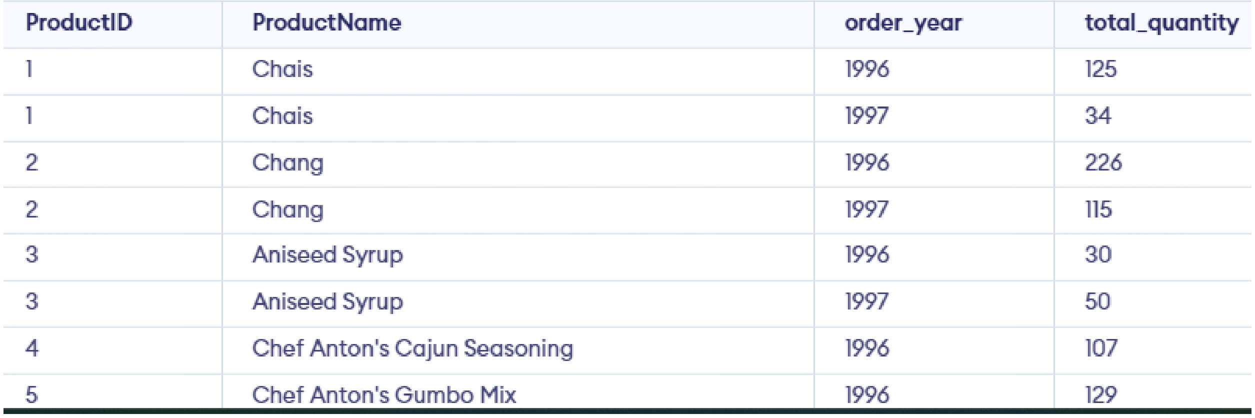


CategoryID	CategoryName	ProductID	ProductName	total_revenue
1	Beverages	1	Chais	2862
1	Beverages	2	Chang	6479
1	Beverages	24	Guaraná Fantástica	711
1	Beverages	34	Sasquatch Ale	1540
1	Beverages	35	Steeleye Stout	6642
1	Beverages	38	Côte de Blaye	62976.5
1	Beverages	39	Chartreuse verte	4788
1	Beverages	43	Ipoh Coffee	6256

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:

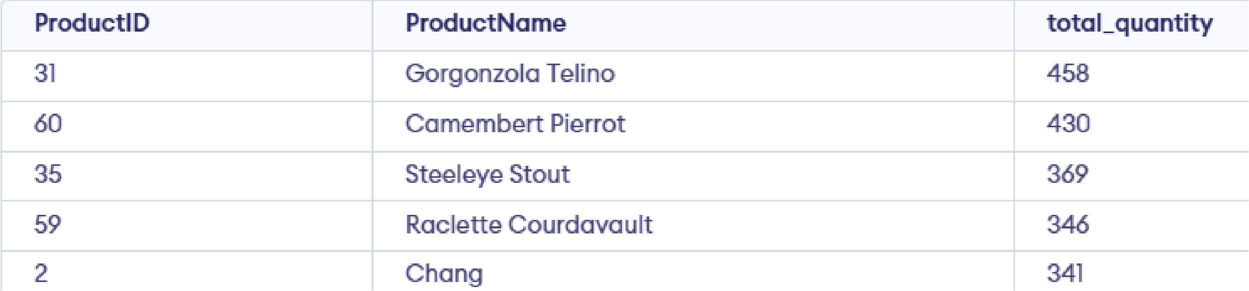


ProductID	ProductName	order_year	total_quantity
1	Chais	1996	125
1	Chais	1997	34
2	Chang	1996	226
2	Chang	1997	115
3	Aniseed Syrup	1996	30
3	Aniseed Syrup	1997	50
4	Chef Anton's Cajun Seasoning	1996	107
5	Chef Anton's Gumbo Mix	1996	129

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:

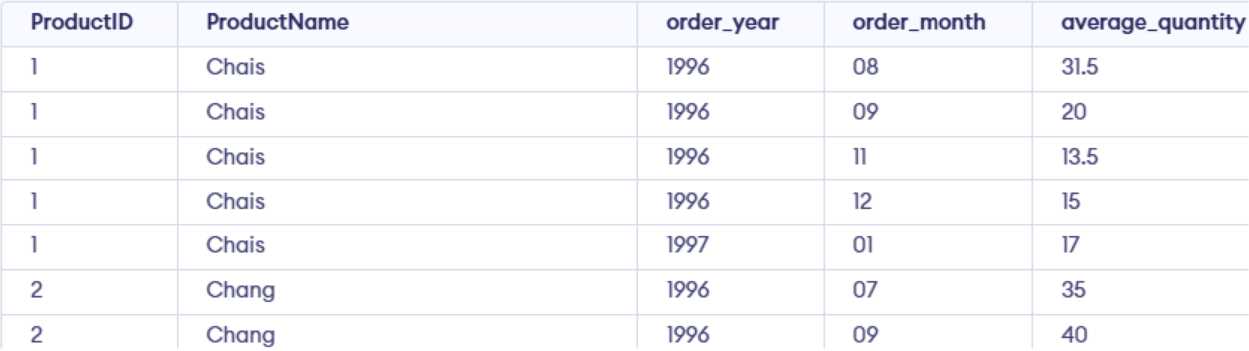


ProductID	ProductName	total_quantity
31	Gorgonzola Telino	458
60	Camembert Pierrot	430
35	Steeleye Stout	369
59	Raclette Courdavault	346
2	Chang	341

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:



ProductID	ProductName	order_year	order_month	average_quantity
1	Chais	1996	08	31.5
1	Chais	1996	09	20
1	Chais	1996	11	13.5
1	Chais	1996	12	15
1	Chais	1997	01	17
2	Chang	1996	07	35
2	Chang	1996	09	40

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

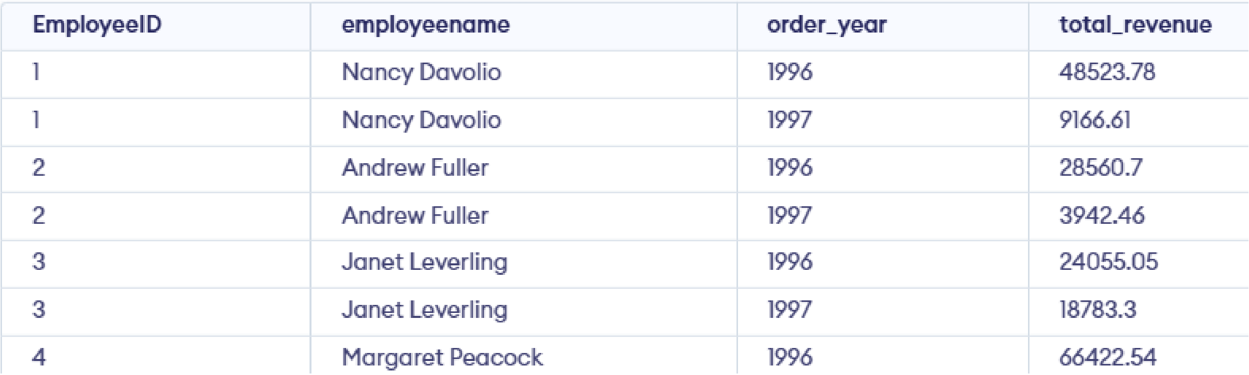
join orders o on e.employeeid = o.employeeid

join orderdetails od on o.orderid = od.orderid

join products p on od.productid = p.productid

```
    group by e.employeeid, order_year
```

Output:

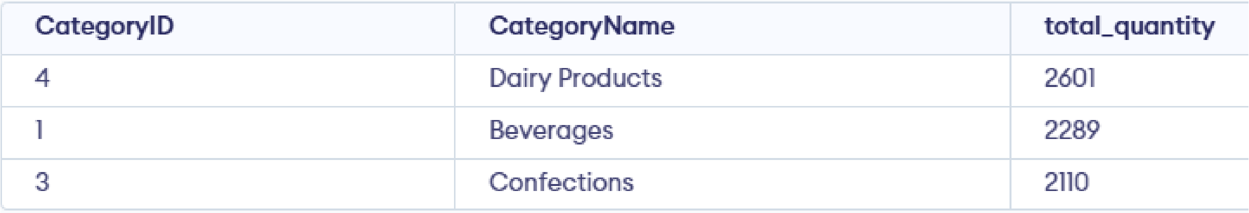


EmployeeID	employeename	order_year	total_revenue
1	Nancy Davolio	1996	48523.78
1	Nancy Davolio	1997	9166.61
2	Andrew Fuller	1996	28560.7
2	Andrew Fuller	1997	3942.46
3	Janet Leverling	1996	24055.05
3	Janet Leverling	1997	18783.3
4	Margaret Peacock	1996	66422.54

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:

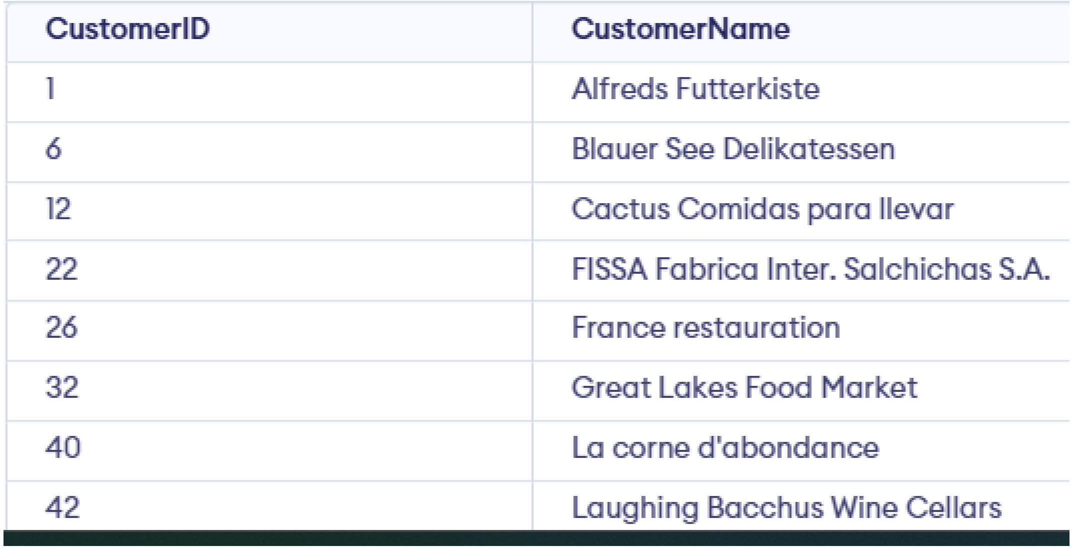


CategoryID	CategoryName	total_quantity
4	Dairy Products	2601
1	Beverages	2289
3	Confections	2110

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:

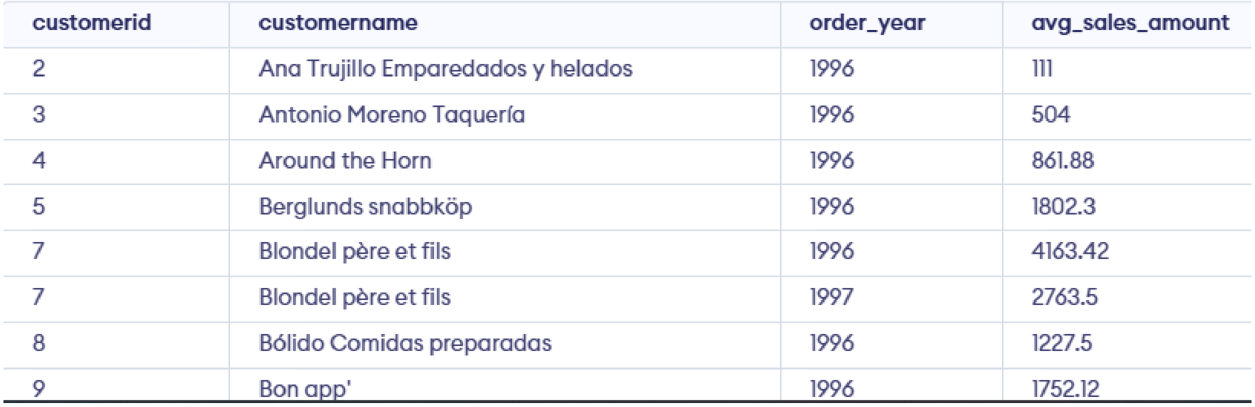


CustomerID	CustomerName
1	Alfreds Futterkiste
6	Blauer See Delikatessen
12	Cactus Comidas para llevar
22	FISSA Fabrica Inter. Salchichas S.A.
26	France restauration
32	Great Lakes Food Market
40	La corne d'abondance
42	Laughing Bacchus Wine Cellars

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:

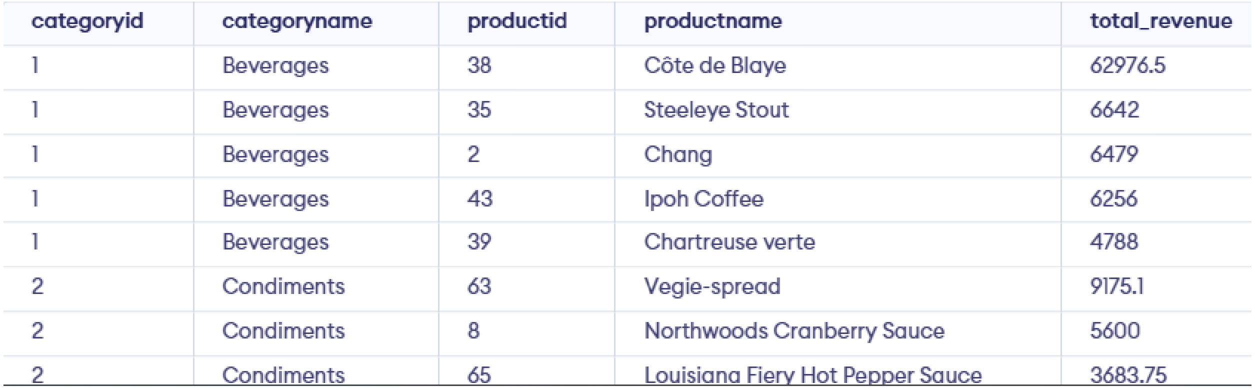


customerid	customername	order_year	avg_sales_amount
2	Ana Trujillo Emparedados y helados	1996	111
3	Antonio Moreno Taquería	1996	504
4	Around the Horn	1996	861.88
5	Berglunds snabbköp	1996	1802.3
7	Blondel père et fils	1996	4163.42
7	Blondel père et fils	1997	2763.5
8	Bólido Comidas preparadas	1996	1227.5
9	Bon app'	1996	1752.12

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:



categoryid	categoryname	productid	productname	total_revenue
1	Beverages	38	Côte de Blaye	62976.5
1	Beverages	35	Steeleye Stout	6642
1	Beverages	2	Chang	6479
1	Beverages	43	Ipoh Coffee	6256
1	Beverages	39	Chartreuse verte	4788
2	Condiments	63	Vegie-spread	9175.1
2	Condiments	8	Northwoods Cranberry Sauce	5600
2	Condiments	65	Louisiana Fiery Hot Pepper Sauce	3683.75

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

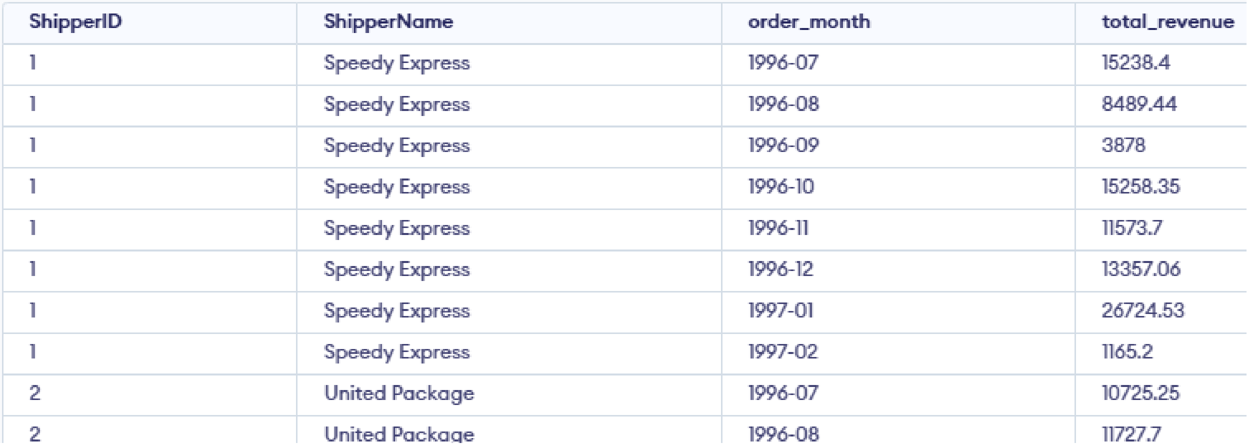
JOIN orders o ON s.shipperid = o.shipperid

JOIN orderdetails od ON o.orderid = od.orderid

JOIN products p ON od.productid = p.productid

```
GROUP BY s.shipperid, order_month
```

Output:



ShipperID	ShipperName	order_month	total_revenue
1	Speedy Express	1996-07	15238.4
1	Speedy Express	1996-08	8489.44
1	Speedy Express	1996-09	3878
1	Speedy Express	1996-10	15258.35
1	Speedy Express	1996-11	11573.7
1	Speedy Express	1996-12	13357.06
1	Speedy Express	1997-01	26724.53
1	Speedy Express	1997-02	1165.2
2	United Package	1996-07	10725.25
2	United Package	1996-08	11727.7