

## Database Schema Documentation

### Core Tables

#### Products

Central table storing product information

- ProductID (Primary Key)
- Name - Product name
- Price - Product price
- StockQuantity - Available inventory
- CategoryID (Foreign Key) - References Categories table

#### Categories

Product categorization

- CategoryID (Primary Key)
- CategoryName - Name of product category

#### Customers

Customer information storage

- CustomerID (Primary Key)
- Name - Customer name
- Email - Customer email address

### Transaction Tables

#### Orders

Main order records

- OrderID (Primary Key)
- CustomerID (Foreign Key) - References Customers table
- OrderDate - Date of order placement

#### OrderDetails

Itemized order information

- DetailID (Primary Key)
- OrderID (Foreign Key) - References Orders table
- ProductID (Foreign Key) - References Products table
- Quantity - Number of items ordered

### Supporting Tables

#### Reviews

Product review system

- ReviewID (Primary Key)
- ProductID (Foreign Key) - References Products table
- CustomerID (Foreign Key) - References Customers table
- Rating - Numerical rating
- Comment - Text review

#### Shipping

Delivery tracking information

- ShippingID (Primary Key)
- OrderID (Foreign Key) - References Orders table
- ShipDate - Date order shipped
- DeliveryDate - Expected/actual delivery date

#### Discounts

Product-specific discounts

- DiscountID (Primary Key)
- ProductID (Foreign Key) - References Products table
- DiscountAmount - Discount value

# E-Commerce Analysis

## 1. Total number of products in each category:

```
748 -- 1. Total number of products in each category
749 SELECT c.CategoryName, COUNT(p.ProductID) AS TotalProducts
750 FROM Categories c
751 LEFT JOIN Products p ON c.CategoryID = p.CategoryID
752 GROUP BY c.CategoryName;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
CategoryName	TotalProducts		
▶ Electronics	4		
Clothing	3		
Home & Kitchen	3		
Books	4		
Beauty & Personal Care	3		
Sports & Outdoors	3		

## 2. Total sales amount for each product:

```
754 -- 2. Total sales amount for each product
755 SELECT p.Name, SUM(od.Quantity * p.Price) AS TotalSales
756 FROM OrderDetails od
757 JOIN Products p ON od.ProductID = p.ProductID
758 GROUP BY p.Name;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Name	TotalSales		
▶ Smartphone	520000.00		
Laptop	900000.00		
Wireless Headphones	78000.00		
Men's T-Shirt	7787.00		
Women's Dress	28578.00		
Jeans	38974.00		
Cooking Pan	16182.00		
Blender	67473.00		
Dish Soap	1188.00		
Fiction Novel	9576.00		
Cookbook	12475.00		
Self-Help Book	5681.00		
Face Wash	6279.00		
Shampoo	11980.00		
Moisturizer	7689.00		
Tennis Racket	27489.00		
Yoga Mat	19980.00		
Dumbbells	47976.00		

### 3. Average rating of each product:

```
760 -- 3. Average rating of each product
761 • SELECT p.Name, AVG(r.Rating) AS AverageRating
762 FROM Products p
763 LEFT JOIN Reviews r ON p.ProductID = r.ProductID
764 GROUP BY p.Name;
765
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Name	AverageRating		
Smartphone	3.2857		
Laptop	3.1429		
Wireless Headphones	3.8571		
Men's T-Shirt	4.1429		
Women's Dress	2.4286		
Jeans	4.0000		
Cooking Pan	3.5714		
Blender	3.7143		
Dish Soap	2.8571		
Fiction Novel	4.1429		
Cookbook	3.7143		
Self-Help Book	2.7143		
Face Wash	4.0000		
Shampoo	2.3333		
Moisturizer	4.6667		
Tennis Racket	2.1667		
Yoga Mat	3.6667		
Dumbbells	3.8333		

### 4. Top 5 customers by total spending

```
766 -- 4.Top 5 customers by total spending
767 • SELECT c.Name, SUM(od.Quantity * p.Price) AS TotalSpent
768 FROM Customers c
769 JOIN Orders o ON c.CustomerID = o.CustomerID
770 JOIN OrderDetails od ON o.OrderID = od.OrderID
771 JOIN Products p ON od.ProductID = p.ProductID
772 GROUP BY c.Name
773 ORDER BY TotalSpent DESC
774 LIMIT 5;
775
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
Name	TotalSpent			
Joseph Martin	200299.00			
Adam Smith	190598.00			
Joshua Baker	150000.00			
Daniel Martinez	101596.00			
Emily Garcia	100099.00			

## 5. the total revenue generated from all orders

```
776 -- 5. the total revenue generated from all orders
777 • SELECT SUM(od.Quantity * p.Price) AS TotalRevenue
778 FROM OrderDetails od
779 JOIN Products p ON od.ProductID = p.ProductID;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	TotalRevenue			
▶	1807307.00			

## 6. Total number of orders placed by top 5 Customers

```
781 -- 6. Total number of orders placed by top 5 Customers
782 • SELECT c.Name, COUNT(o.OrderID) AS TotalOrders
783 FROM Customers c
784 LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
785 GROUP BY c.Name
786 ORDER BY TotalOrders DESC
787 LIMIT 5;
788
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	Name	TotalOrders			
▶	Adam Smith	3			
	Eve Johnson	2			
	John Doe	2			
	Jane Brown	2			
	Michael Davis	2			

## 7. Products with discounts greater than 200

```
789      -- 7. Products with discounts greater than 200
790 •    SELECT p.Name, d.DiscountAmount
791      FROM Products p
792      JOIN Discounts d ON p.ProductID = d.ProductID
793      WHERE d.DiscountAmount > 200;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Name	DiscountAmount		
▶ Smartphone	1500.00		
Laptop	5000.00		
Blender	250.00		
Tennis Racket	300.00		

## 8. Total quantity of products sold for each category

```
794
795      -- 8. Total quantity of products sold for each category
796 •    SELECT c.CategoryName, SUM(od.Quantity) AS TotalQuantitySold
797      FROM Categories c
798      JOIN Products p ON c.CategoryID = p.CategoryID
799      JOIN OrderDetails od ON p.ProductID = od.ProductID
800      GROUP BY c.CategoryName
801      ORDER BY TotalQuantitySold DESC;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
CategoryName	TotalQuantitySold		
▶ Electronics	70		
Books	68		
Clothing	61		
Home & Kitchen	57		
Sports & Outdoors	55		
Beauty & Personal Care	52		

## 9. Customers who have purchased more than 1 different products

```
803 -- 9. Customers who have purchased more than 1 different products
804 • SELECT c.Name FROM Customers c
805 JOIN Orders o ON c.CustomerID = o.CustomerID
806 JOIN OrderDetails od ON o.OrderID = od.OrderID
807 GROUP BY c.CustomerID
808 HAVING COUNT(DISTINCT od.ProductID) > 1;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Name			
Adam Smith			
Eve Johnson			
John Doe			
Jane Brown			
Michael Davis			
Sarah Wilson			
David Miller			
Emily Garcia			
Daniel Martinez			
Laura Rodriguez			
James Anderson			
Linda Thomas			
Robert Taylor			
Patricia Hernan...			
Charles Moore			
Jessica Jackson			
William White			
Barbara Harris			
Joseph Martin			

## 10. Average price of products in each category

```
810 -- 10. Average price of products in each category
811 • SELECT c.CategoryName, AVG(p.Price) AS AveragePrice
812 FROM Categories c
813 JOIN Products p ON c.CategoryID = p.CategoryID
814 GROUP BY c.CategoryName;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
CategoryName	AveragePrice		
Electronics	18750.000000		
Clothing	1132.333333		
Home & Kitchen	1165.666667		
Books	349.000000		
Beauty & Personal Care	532.333333		
Sports & Outdoors	1832.333333		

```

816 -- 11. Total number of reviews for each product
817 • SELECT p.Name, COUNT(r.ReviewID) AS TotalReviews
818 FROM Products p
819 LEFT JOIN Reviews r ON p.ProductID = r.ProductID
820 GROUP BY p.Name;
821

```



Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	Name	TotalReviews
▶	Smartphone	7
	Laptop	7
	Wireless Headphones	7
	Men's T-Shirt	7
	Women's Dress	7
	Jeans	7
	Cooking Pan	7
	Blender	7
	Dish Soap	7
	Fiction Novel	7
	Cookbook	7
	Self-Help Book	7
	Face Wash	6
	Shampoo	6
	Moisturizer	6
	Tennis Racket	6
	Yoga Mat	6
	Dumbbells	6

```

822 -- 12. Top 5 products by total sales
823 • SELECT p.Name, SUM(od.Quantity) AS TotalSold
824 FROM Products p
825 JOIN OrderDetails od ON p.ProductID = od.ProductID
826 GROUP BY p.Name
827 ORDER BY TotalSold DESC
828 LIMIT 5;
829

```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content:  | Fetch rows: 

	Name	TotalSold
▶	Blender	27
	Smartphone	26
	Wireless Headphones	26
	Jeans	26
	Cookbook	25

```

830  -- 13.Total discounts applied to each product
831  • SELECT p.Name, SUM(d.DiscountAmount) AS TotalDiscount
832  FROM Products p
833  LEFT JOIN Discounts d ON p.ProductID = d.ProductID
834  GROUP BY p.Name;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Name	TotalDiscount
Smartphone	1500.00
Laptop	5000.00
Wireless Headphones	NULL
Men's T-Shirt	100.00
Women's Dress	200.00
Jeans	NULL
Cooking Pan	50.00
Blender	250.00
Dish Soap	NULL
Fiction Novel	NULL
Cookbook	30.00
Self-Help Book	NULL
Face Wash	NULL
Shampoo	50.00
Moisturizer	70.00
Tennis Racket	300.00
Yoga Mat	100.00
Dumbbells	200.00
Children's Book	NULL
Speaker	NULL

```

836  -- 14. How many orders were placed in the last month
837  • SELECT COUNT(*) AS OrdersLastMonth
838  FROM Orders
839  WHERE OrderDate >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH);
840

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

OrdersLastMonth
5



```

841 -- 15. Products with the highest average rating
842 • SELECT p.Name, AVG(r.Rating) AS AverageRating
843 FROM Products p
844 JOIN Reviews r ON p.ProductID = r.ProductID
845 GROUP BY p.Name
846 ORDER BY AverageRating DESC
847 LIMIT 5;

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	Name	AverageRating				
▶	Moisturizer	4.6667				
	Fiction Novel	4.1429				
	Men's T-Shirt	4.1429				
	Face Wash	4.0000				
	Jeans	4.0000				

```

849 -- 16. Total Revenue Generated in the Last Month
850 • SELECT SUM(od.Quantity * p.Price) AS TotalRevenueLastMonth
851 FROM Orders o
852 JOIN OrderDetails od ON o.OrderID = od.OrderID
853 JOIN Products p ON od.ProductID = p.ProductID
854 WHERE o.OrderDate >= NOW() - INTERVAL 1 MONTH;
855





```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	TotalRevenueLastMonth				
▶	11986.00				

```

856 -- 17.Total Revenue by Year
857 • SELECT YEAR(o.OrderDate) AS OrderYear, SUM(od.Quantity * p.Price) AS TotalRevenue
858 FROM OrderDetails od
859 JOIN Products p ON od.ProductID = p.ProductID
860 JOIN Orders o ON od.OrderID = o.OrderID
861 GROUP BY YEAR(o.OrderDate)
862 ORDER BY OrderYear ASC;
863

```

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 

	OrderYear	TotalRevenue
▶	2023	1034848.00
	2024	772459.00

```

864 -- 18. Total Revenue for Every Month of the Year 2024
865 • SELECT MONTH(o.OrderDate) AS OrderMonth, SUM(od.Quantity * p.Price) AS TotalRevenue
866 FROM OrderDetails od
867 JOIN Products p ON od.ProductID = p.ProductID
868 JOIN Orders o ON od.OrderID = o.OrderID
869 WHERE YEAR(o.OrderDate) = 2024
870 GROUP BY MONTH(o.OrderDate)
871 ORDER BY TotalRevenue desc;
872

```

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 

	OrderMonth	TotalRevenue
▶	6	219993.00
	10	214795.00
	3	122290.00
	2	115691.00
	7	22585.00
	11	21279.00
	9	19086.00
	4	13485.00
	5	9588.00
	1	7982.00
	8	5685.00

```

873 -- 19. Top 3 categories by total sales
874 • SELECT c.CategoryName, SUM(od.Quantity * p.Price) AS TotalSales
875 FROM Categories c
876 JOIN Products p ON c.CategoryID = p.CategoryID
877 JOIN OrderDetails od ON p.ProductID = od.ProductID
878 JOIN Orders o ON od.OrderID = o.OrderID
879 GROUP BY c.CategoryName
880 ORDER BY TotalSales DESC
881 LIMIT 3;

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
CategoryName	TotalSales			
▶ Electronics	1498000.00			
Sports & Outdoors	95445.00			
Home & Kitchen	84843.00			

```

883 -- 20. Total number of products in stock for each category
884 • SELECT c.CategoryName, SUM(p.StockQuantity) AS TotalStock
885 FROM Categories c
886 JOIN Products p ON c.CategoryID = p.CategoryID
887 GROUP BY c.CategoryName;



```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
CategoryName	TotalStock		
▶ Electronics	107		
Clothing	220		
Home & Kitchen	210		
Books	300		
Beauty & Personal Care	250		
Sports & Outdoors	120		

```

889 -- 21. Revenue by Quarter of previous year 2023
890 • SELECT YEAR(o.OrderDate) AS Year,
891        QUARTER(o.OrderDate) AS Quarter,
892        SUM(od.Quantity * p.Price) AS Revenue
893 FROM Orders o
894 JOIN OrderDetails od ON o.OrderID = od.OrderID
895 JOIN Products p ON od.ProductID = p.ProductID
896 WHERE YEAR(o.OrderDate) = 2023
897 GROUP BY YEAR(o.OrderDate), QUARTER(o.OrderDate);
898

```





Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	Year	Quarter	Revenue
▶	2023	1	244962.00
	2023	2	292762.00
	2023	3	244559.00
	2023	4	252565.00

```

899 -- 22. Highest Earning Month of 2023
900 • SELECT YEAR(o.OrderDate), MONTH(o.OrderDate) AS Month, SUM(od.Quantity * p.Price) AS Revenue
901 FROM Orders o
902 JOIN OrderDetails od ON o.OrderID = od.OrderID
903 JOIN Products p ON od.ProductID = p.ProductID
904 WHERE YEAR(o.OrderDate) = 2023
905 GROUP BY YEAR(o.OrderDate), MONTH(o.OrderDate)
906 ORDER BY Revenue DESC
907 LIMIT 1;

```





Result Grid   Filter Rows:  Export:  Wrap Cell Content:  Fetch rows: 

	YEAR(o.OrderDate)	Month	Revenue
▶	2023	4	265992.00

```

909      -- 23.Total sales on weekends vs weekdays IN YEAR 2023
910  • SELECT CASE
911          WHEN DAYOFWEEK(o.OrderDate) IN (1, 7) THEN 'Weekend'
912          ELSE 'Weekday'
913      END AS DayType,
914          SUM(od.Quantity * p.Price) AS TotalSales
915  FROM Orders o
916  JOIN OrderDetails od ON o.OrderID = od.OrderID
917  JOIN Products p ON od.ProductID = p.ProductID
918  WHERE YEAR(o.OrderDate) = 2023
919  GROUP BY DayType;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	DayType	TotalSales
▶	Weekend	354938.00
	Weekday	679910.00

340

```

921      -- 24. Obtain All Negative Reviews (Rating 1 or 2) Given by Customers to Products
922  • SELECT r.ReviewID, r.Comment, r.Rating, p.Name AS product_name, c.Name AS Customer_name
923  FROM Reviews r
924  JOIN Products p ON r.ProductID = p.ProductID
925  JOIN Customers c ON r.CustomerID = c.CustomerID
926  WHERE r.Rating <= 2
927  limit 10;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content:  Fetch rows: 

	ReviewID	Comment	Rating	product_name	Customer_name
▶	4	Not what I expected, disappointed.	2	Men's T-Shirt	John Doe
	5	Terrible experience, will not buy again.	1	Women's Dress	Jane Brown
	12	Not worth the price.	2	Self-Help Book	James Anderson
	16	Horrible quality, very disappointed.	1	Tennis Racket	Charles Moore
	19	Not great, could use improvements.	2	Smartphone	Barbara Harris
	23	Did not like it at all.	1	Women's Dress	Margaret Martinez
	27	Not satisfied, expected better.	2	Dish Soap	Nancy Lewis
	32	Not worth the money.	2	Shampoo	Steven Allen
	34	Terrible product, do not buy.	1	Tennis Racket	Paul Wright
	38	Not what I expected.	2	Laptop	Joshua Baker

```

929  -- 25. What is the total discount amount applied across all orders
930  • SELECT SUM(d.DiscountAmount) AS TotalDiscount
931  FROM Discounts d
932  JOIN Products p ON d.ProductID = p.ProductID
933  JOIN OrderDetails od ON p.ProductID = od.ProductID;

```

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 

	TotalDiscount
▶	54230.00

```

935  -- 26. Quarterly sales performance for 2024
936  • SELECT YEAR(o.OrderDate) AS Year,
937         QUARTER(o.OrderDate) AS Quarter,
938         SUM(od.Quantity * p.Price) AS Revenue
939  FROM Orders o
940  JOIN OrderDetails od ON o.OrderID = od.OrderID
941  JOIN Products p ON od.ProductID = p.ProductID
942  WHERE YEAR(o.OrderDate) = 2024
943  GROUP BY YEAR(o.OrderDate), QUARTER(o.OrderDate);

```

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 






	Year	Quarter	Revenue
▶	2024	1	245963.00
	2024	2	243066.00
	2024	4	236074.00
	2024	3	47356.00



```

945 -- 27. Highest Earning Month of 2024
946 • SELECT YEAR(o.OrderDate), MONTH(o.OrderDate) AS Month, SUM(od.Quantity * p.Price) AS Revenue
947 FROM Orders o
948 JOIN OrderDetails od ON o.OrderID = od.OrderID
949 JOIN Products p ON od.ProductID = p.ProductID
950 WHERE YEAR(o.OrderDate) = 2024
951 GROUP BY YEAR(o.OrderDate), MONTH(o.OrderDate)
952 ORDER BY Revenue DESC
953 LIMIT 1;

```






Result Grid   Filter Rows:  | Export:  | Wrap Cell Content:  | Fetch rows: 

	YEAR(o.OrderDate)	Month	Revenue
▶	2024	6	219993.00

```

955 -- 28. Identify the Best-Selling Product
956 • SELECT p.Name AS ProductName, SUM(od.Quantity) AS TotalSold
957 FROM Products p
958 JOIN OrderDetails od ON p.ProductID = od.ProductID
959 GROUP BY p.ProductID
960 ORDER BY TotalSold DESC
961 LIMIT 1;
962

```

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content:  | Fetch rows: 

	ProductName	TotalSold
▶	Blender	27

```

963      -- 29 Total sales on weekends vs weekdays IN YEAR 2024
964 • SELECT CASE
965         WHEN DAYOFWEEK(o.OrderDate) IN (1, 7) THEN 'Weekend'
966         ELSE 'Weekday'
967     END AS DayType,
968         SUM(od.Quantity * p.Price) AS TotalSales
969 FROM Orders o
970 JOIN OrderDetails od ON o.OrderID = od.OrderID
971 JOIN Products p ON od.ProductID = p.ProductID
972 WHERE YEAR(o.OrderDate) = 2024
973 GROUP BY DayType;

```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	DayType	TotalSales
▶	Weekday	615999.00
	Weekend	156460.00

```

975      -- 30. Identify the Least-Selling Product
976 • SELECT p.Name AS ProductName, SUM(od.Quantity) AS TotalSold
977 FROM Products p
978 JOIN OrderDetails od ON p.ProductID = od.ProductID
979 GROUP BY p.ProductID
980 ORDER BY TotalSold ASC
981 LIMIT 1;
982

```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content:  | Fetch rows: 



	ProductName	TotalSold
▶	Moisturizer	11



```

983      -- 31. Products without any discounts
984  •   SELECT p.Name
985      FROM Products p
986      LEFT JOIN Discounts d ON p.ProductID = d.ProductID
987      WHERE d.DiscountAmount IS NULL;
988

```


Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	Name
▶	Wireless Headphones
	Jeans
	Dish Soap
	Fiction Novel
	Self-Help Book
	Face Wash
	Children's Book
	Speaker

```

989      -- 32. Obtain All Positive Reviews (Rating 4 or 5) Given by Customers to Products
990  •   SELECT r.ReviewID, r.Comment, r.Rating, p.Name AS product_name, c.Name AS Customer_name
991      FROM Reviews r
992      JOIN Products p ON r.ProductID = p.ProductID
993      JOIN Customers c ON r.CustomerID = c.CustomerID
994      WHERE r.Rating >= 4
995      limit 10;
996

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content:  Fetch rows: 

	ReviewID	Comment	Rating	product_name	Customer_name
▶	1	Excellent product! Highly recommend.	5	Smartphone	Adam Smith
	2	Very good quality, satisfied with my purchase.	4	Laptop	Adam Smith
	6	Absolutely loved it! Will buy again.	5	Jeans	Michael Davis
	7	Good value for money.	4	Cooking Pan	Sarah Wilson
	8	Amazing! Exceeded my expectations.	5	Blender	David Miller
	10	Pretty good, would recommend.	4	Fiction Novel	Daniel Martinez
	11	Fantastic! Will tell my friends.	5	Cookbook	Laura Rodriguez
	13	Very nice product, happy with my purchase.	4	Face Wash	Linda Thomas
	15	Loved it! Will buy more.	5	Moisturizer	Patricia Hernandez
	17	Good product, met my expectations.	4	Yoga Mat	Jessica Jackson

```

997 -- 33. Get the Number of Reviews for Each Product
998 • SELECT p.ProductID, p.Name, COUNT(r.ReviewID) AS ReviewCount
999 FROM Products p
1000 LEFT JOIN Reviews r ON p.ProductID = r.ProductID
1001 GROUP BY p.ProductID, p.Name;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	ProductID	Name	ReviewCount
▶	101	Smartphone	7
	102	Laptop	7
	103	Wireless Headphones	7
	104	Men's T-Shirt	7
	105	Women's Dress	7
	106	Jeans	7
	107	Cooking Pan	7
	108	Blender	7
	109	Dish Soap	7
	110	Fiction Novel	7
	111	Cookbook	7
	112	Self-Help Book	7
	113	Face Wash	6
	114	Shampoo	6
	115	Moisturizer	6
	116	Tennis Racket	6
	117	Yoga Mat	6
	118	Dumbbells	6
	119	Children's Book	0
	120	Speaker	0

```

1003 -- 34. the total number of orders delivered in the first half of the year 2024
1004
1005 • SELECT COUNT(o.OrderID) AS TotalOrdersDelivered
1006 FROM Orders o
1007 JOIN Shipping s ON o.OrderID = s.OrderID
1008 WHERE s.DeliveryDate >= '2024-01-01' AND s.DeliveryDate <= '2024-06-30';

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	TotalOrdersDelivered
▶	30

```

1010 -- 35. Get Products with No Reviews
1011 • SELECT p.ProductID, p.Name
1012 FROM Products p
1013 LEFT JOIN Reviews r ON p.ProductID = r.ProductID
1014 WHERE r.ReviewID IS NULL;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	ProductID	Name
▶	119	Children's Book
	120	Speaker

```

1016 -- 36. The sales of each category where total sales exceeds 80000
1017 • SELECT c.CategoryName, SUM(od.Quantity * p.Price) AS TotalSales
1018 FROM Categories c
1019 JOIN Products p ON c.CategoryID = p.CategoryID
1020 JOIN OrderDetails od ON p.ProductID = od.ProductID
1021 JOIN Orders o ON od.OrderID = o.OrderID
1022 GROUP BY c.CategoryName
1023 having TotalSales > 80000;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	CategoryName	TotalSales
▶	Electronics	1498000.00
	Home & Kitchen	84843.00
	Sports & Outdoors	95445.00

```

1025      -- 37. Get Products with an Average Rating Below 3
1026 •    SELECT p.ProductID, p.Name, AVG(r.Rating) AS AverageRating
1027      FROM Products p
1028      JOIN Reviews r ON p.ProductID = r.ProductID
1029      GROUP BY p.ProductID, p.Name
1030      HAVING AVG(r.Rating) < 3;

```

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 

	ProductID	Name	AverageRating
▶	105	Women's Dress	2.4286
	109	Dish Soap	2.8571
	112	Self-Help Book	2.7143
	114	Shampoo	2.3333
	116	Tennis Racket	2.1667

```

1032      -- 38. Show Product Wise Min, Max, Avg, Count, Sum Orders
1033 •    SELECT p.ProductID, p.Name,
1034           MIN(od.Quantity) AS MinOrderQuantity,
1035           MAX(od.Quantity) AS MaxOrderQuantity,
1036           AVG(od.Quantity) AS AvgOrderQuantity,
1037           COUNT(od.DetailID) AS OrderCount,
1038           SUM(od.Quantity) AS TotalOrdered
1039      FROM Products p
1040      LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID
1041      GROUP BY p.ProductID, p.Name;

```

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 





	ProductID	Name	MinOrderQuantity	MaxOrderQuantity	AvgOrderQuantity	OrderCount	TotalOrdered
▶	101	Smartphone	2	5	3.7143	7	26
	102	Laptop	2	4	2.5714	7	18
	103	Wireless Headphones	3	5	3.7143	7	26
	104	Men's T-Shirt	1	5	1.8571	7	13
	105	Women's Dress	1	4	3.1429	7	22
	106	Jeans	2	5	3.7143	7	26
	107	Cooking Pan	2	4	2.5714	7	18
	108	Blender	3	6	3.8571	7	27
	109	Dish Soap	1	5	1.7143	7	12
	110	Fiction Novel	1	4	3.4286	7	24
	111	Cookbook	1	5	3.5714	7	25
	112	Self-Help Book	1	5	2.7143	7	19
	113	Face Wash	3	5	3.5000	6	21
	114	Shampoo	1	5	3.3333	6	20
	115	Moisturizer	1	4	1.8333	6	11



```

1043 -- 39. Find Average Delivery Time from OrderDate to DeliveryDate
1044 • SELECT AVG(DATEDIFF(s.DeliveryDate, o.OrderDate)) AS AvgDeliveryTime
1045 FROM Orders o
1046 JOIN Shipping s ON o.OrderID = s.OrderID;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	AvgDeliveryTime
▶	1.5500

```

1048 -- 40. Get Total Quantity Sold for Each Product
1049 • SELECT p.ProductID, p.Name, SUM(od.Quantity) AS TotalSold
1050 FROM Products p
1051 JOIN OrderDetails od ON p.ProductID = od.ProductID
1052 GROUP BY p.ProductID, p.Name;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	ProductID	Name	TotalSold
▶	101	Smartphone	26
	102	Laptop	18
	103	Wireless Headphones	26
	104	Men's T-Shirt	13
	105	Women's Dress	22
	106	Jeans	26
	107	Cooking Pan	18
	108	Blender	27
	109	Dish Soap	12
	110	Fiction Novel	24
	111	Cookbook	25
	112	Self-Help Book	19
	113	Face Wash	21
	114	Shampoo	20
	115	Moisturizer	11
	116	Tennis Racket	11
	117	Yoga Mat	20
	118	Dumbbells	24

```

1054      -- 41. Get Customers Who Have Not Placed Any Orders
1055  •    SELECT c.CustomerID, c.Name
1056      FROM Customers c
1057      LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
1058      WHERE o.OrderID IS NULL;
-----

```






Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	CustomerID	Name
▶	101	john wick
	102	Jane Doe
	103	David Lee
	104	Sarah Jones
	105	Michael Brown
	106	Emily Davis
	107	Daniel White
	108	Olivia Miller
	109	James Wilson

```

1060      -- 42. Get All Orders with Shipping Information
1061  •    SELECT o.OrderID, o.OrderDate, s.ShipDate, s.DeliveryDate
1062      FROM Orders o
1063      JOIN Shipping s ON o.OrderID = s.OrderID
1064      limit 10;
-----

```



Result Grid   Filter Rows:  Export:  Wrap Cell Content:  Fetch rows: 

	OrderID	OrderDate	ShipDate	DeliveryDate
▶	1001	2023-01-01 01:00:00	2023-01-02 02:00:50	2023-01-03 01:00:00
	1002	2023-01-06 12:34:56	2023-01-07 13:30:57	2023-01-08 12:34:56
	1003	2023-01-11 08:15:30	2023-01-12 08:15:30	2023-01-13 08:15:30
	1004	2023-01-17 20:45:12	2023-01-18 21:45:12	2023-01-19 20:45:12
	1005	2023-01-21 19:30:00	2023-01-22 20:30:00	2023-01-23 19:30:00
	1006	2023-01-27 14:22:33	2023-01-28 16:22:33	2023-01-29 14:22:33
	1007	2023-02-02 03:45:00	2023-02-03 03:45:00	2023-02-04 03:45:00
	1008	2023-02-03 11:11:11	2023-02-04 11:11:11	2023-02-05 11:11:11
	1009	2023-02-16 22:22:22	2023-02-17 22:22:22	2023-02-18 22:22:22
	1010	2023-02-19 17:17:17	2023-02-20 17:17:17	2023-02-21 17:17:17

```

1066 -- 43. Get the Lowest Rated Product
1067 • SELECT p.ProductID, p.Name, AVG(r.Rating) AS AverageRating
1068 FROM Products p
1069 JOIN Reviews r ON p.ProductID = r.ProductID
1070 GROUP BY p.ProductID, p.Name
1071 ORDER BY AverageRating ASC
1072 LIMIT 1;
1073

```


Result Grid   Filter Rows:  | Export:  | Wrap Cell Content:  | Fetch rows: 

	ProductID	Name	AverageRating
▶	116	Tennis Racket	2.1667

```

1074 -- 44. Get All Products with Stock quantity below 10
1075 • SELECT p.ProductID, p.Name, p.StockQuantity
1076 FROM Products p
1077 WHERE p.StockQuantity < 10;

```

Result Grid   Filter Rows:  | Edit:    | Export/Import:   | W

	ProductID	Name	StockQuantity
▶	120	Speaker	7
•	NULL	NULL	NULL

```

1079 -- 45. Get All Orders with More Than 5 Items
1080 • SELECT o.OrderID, SUM(od.Quantity) AS TotalItems
1081 FROM Orders o
1082 JOIN OrderDetails od ON o.OrderID = od.OrderID
1083 GROUP BY o.OrderID
1084 HAVING TotalItems > 5;
1085

```


Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 

	OrderID	TotalItems
▶	1008	6

```

1086      -- 46. Get All Products That Have Never Been Ordered
1087  •    SELECT p.ProductID, p.Name
1088      FROM Products p
1089      LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID
1090      WHERE od.DetailID IS NULL;
1091

```





Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	ProductID	Name
▶	119	Children's Book
	120	Speaker

```

1092      -- 47. Get All Products with Their Categories and Discounts
1093  •    SELECT p.ProductID, p.Name, c.CategoryName, d.DiscountAmount
1094      FROM Products p
1095      JOIN Categories c ON p.CategoryID = c.CategoryID
1096      LEFT JOIN Discounts d ON p.ProductID = d.ProductID
1097      order by d.DiscountAmount DESC
1098      LIMIT 12;
1099

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content:  Fetch rows: 


	ProductID	Name	CategoryName	DiscountAmount
▶	102	Laptop	Electronics	5000.00
	101	Smartphone	Electronics	1500.00
	116	Tennis Racket	Sports & Outdoors	300.00
	108	Blender	Home & Kitchen	250.00
	105	Women's Dress	Clothing	200.00
	118	Dumbbells	Sports & Outdoors	200.00
	104	Men's T-Shirt	Clothing	100.00
	117	Yoga Mat	Sports & Outdoors	100.00
	115	Moisturizer	Beauty & Personal Care	70.00
	107	Cooking Pan	Home & Kitchen	50.00
	114	Shampoo	Beauty & Personal Care	50.00
	111	Cookbook	Books	30.00



```

1100 -- 48. Total Revenue for the 4th Quarter of 2024
1101 • SELECT SUM(od.Quantity * p.Price) AS 4th_qtr_Revenue
1102 FROM Orders o
1103 JOIN OrderDetails od ON o.OrderID = od.OrderID
1104 JOIN Products p ON od.ProductID = p.ProductID
1105 WHERE QUARTER(o.OrderDate) = 4 AND YEAR(o.OrderDate) = 2024;

```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	4th_qtr_Revenue
▶	236074.00

```

1107 -- 49. Get the Most Popular Product Based on Sales Volume
1108 • SELECT p.ProductID, p.Name,
1109      (SELECT SUM(od.Quantity)
1110       FROM OrderDetails od
1111       WHERE od.ProductID = p.ProductID) AS TotalSalesVolume
1112 FROM Products p
1113 ORDER BY TotalSalesVolume DESC
1114 LIMIT 3;


```


Result Grid   Filter Rows:  Export:  Wrap Cell Content:  Fetch rows: 


	ProductID	Name	TotalSalesVolume
▶	108	Blender	27
	101	Smartphone	26
	103	Wireless Headphones	26


```
1116 -- 50. Deliveries between 6:00 AM and 12:00 PM (Morning Deliveries)
1117 • SELECT ShippingId, OrderId, DATE(deliverydate) AS Delivery_date, TIME(deliverydate) AS Delivery_Time
1118 FROM shipping
1119 WHERE HOUR(deliverydate) >= 6 AND HOUR(deliverydate) < 12
1120 order by Delivery_date
1121 limit 10;
```

Result Grid

 Filter Rows:

Export: 

Wrap Cell Content: 

Fetch rows: 

	ShippingId	OrderId	Delivery_date	Delivery_Time
▶	3	1003	2023-01-13	08:15:30
	8	1008	2023-02-05	11:11:11
	11	1011	2023-02-28	09:09:09
	12	1012	2023-03-05	06:06:06
	18	1018	2023-04-08	10:10:10
	23	1023	2023-05-09	11:11:11
	25	1025	2023-05-19	07:07:07
	33	1033	2023-07-11	09:09:09
	38	1038	2023-08-09	10:10:10
	43	1043	2023-09-11	11:00:11