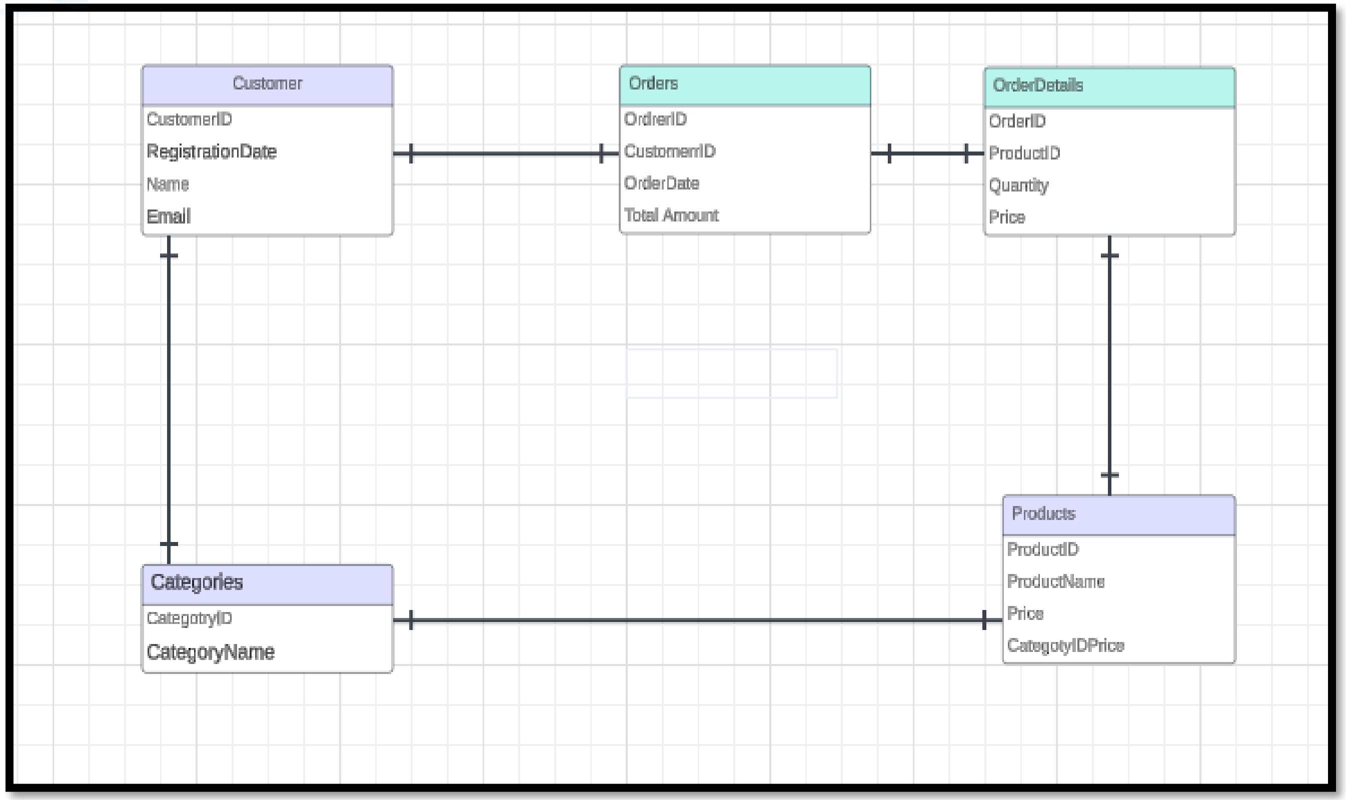
Retail Analytics Database

Presented by- Amulya Singh

Database Schema



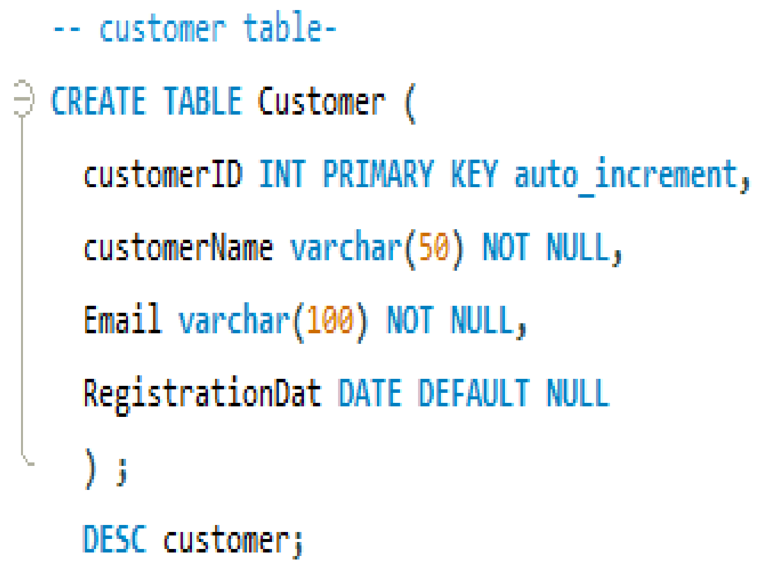
Objective

- The objective of this project is to design and manage a relational database system for an online retail store, encapsulating essential components such as Customers, Products, Categories, Orders, and Order Details. The database aims to streamline and analyze transaction data to support efficient operations, enhance business insights, and improve decision-making. By implementing this schema, the project seeks to:

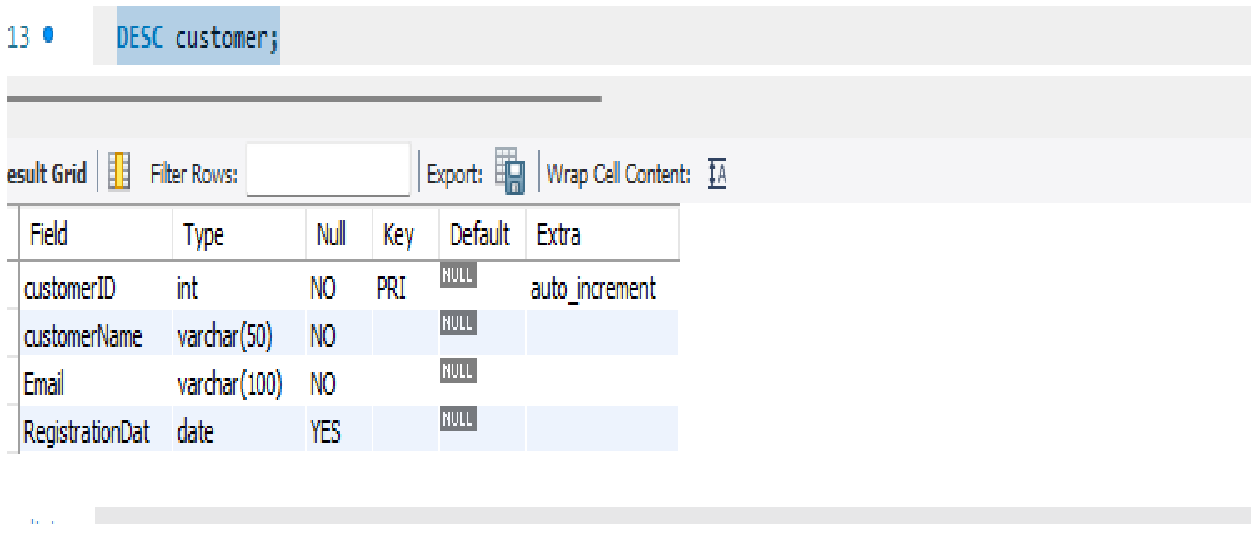
- Facilitate effective management of customer information, product inventories, and order processing.
- Enable detailed analysis of sales patterns, customer behavior, and product performance.
- Support advanced queries to extract valuable insights and optimize retail operations.

- Create all the tables with the specified columns and foreign key references.

Customer -



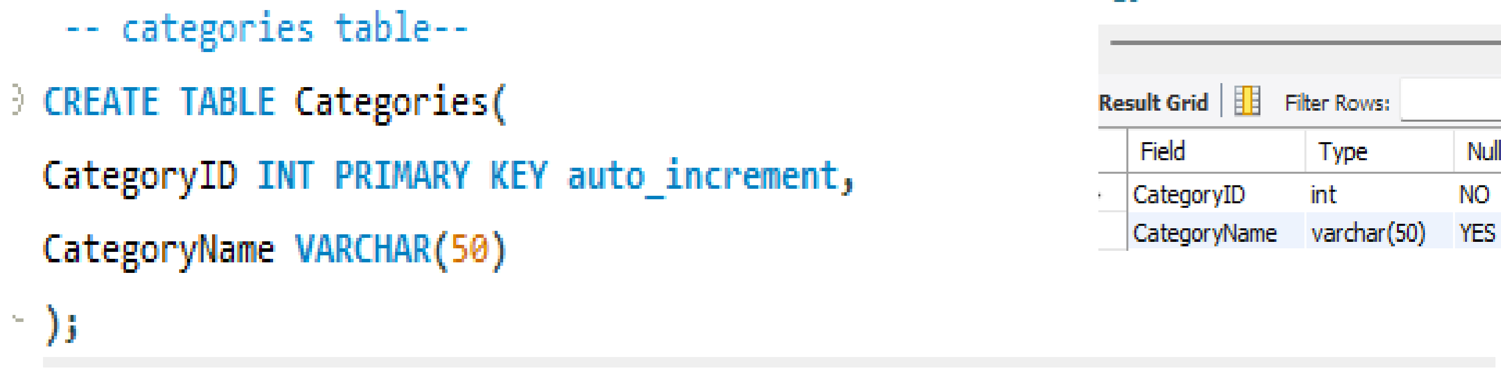
```
-- customer table-
CREATE TABLE Customer (
  customerID INT PRIMARY KEY auto_increment,
  customerName varchar(50) NOT NULL,
  Email varchar(100) NOT NULL,
  RegistrationDat DATE DEFAULT NULL
  ) ;
  DESC customer;
```



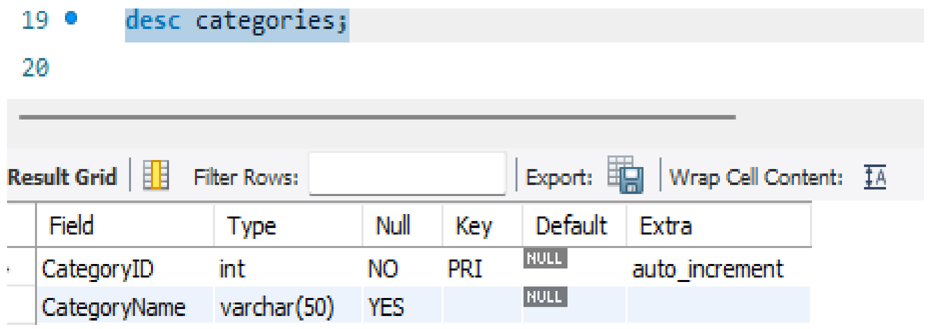
```
DESC customer;
```

Field	Type	Null	Key	Default	Extra
customerID	int	NO	PRI	NULL	auto_increment
customerName	varchar(50)	NO		NULL	
Email	varchar(100)	NO		NULL	
RegistrationDat	date	YES		NULL	

Categories-



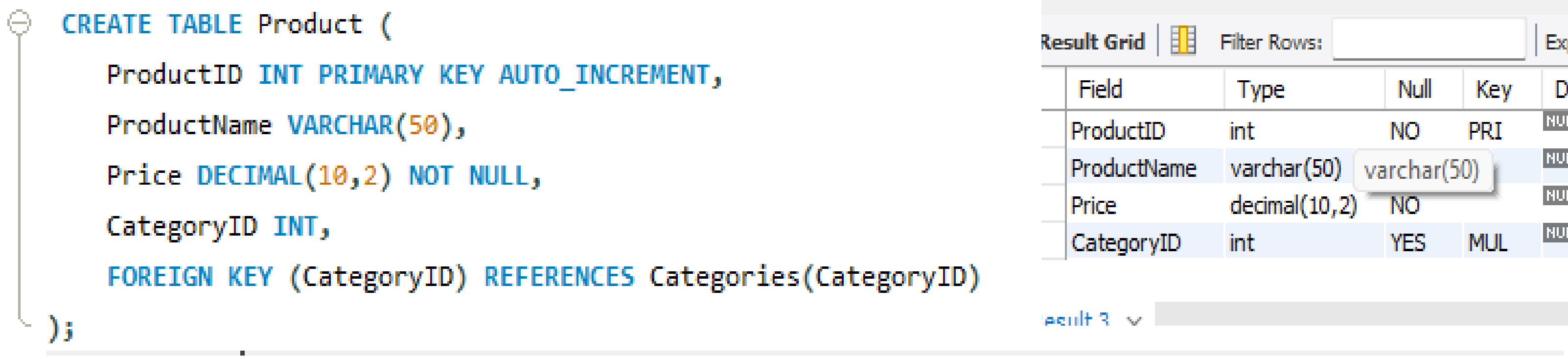
```
 -- categories table--
CREATE TABLE Categories(
CategoryID INT PRIMARY KEY auto_increment,
CategoryName VARCHAR(50)
);
```



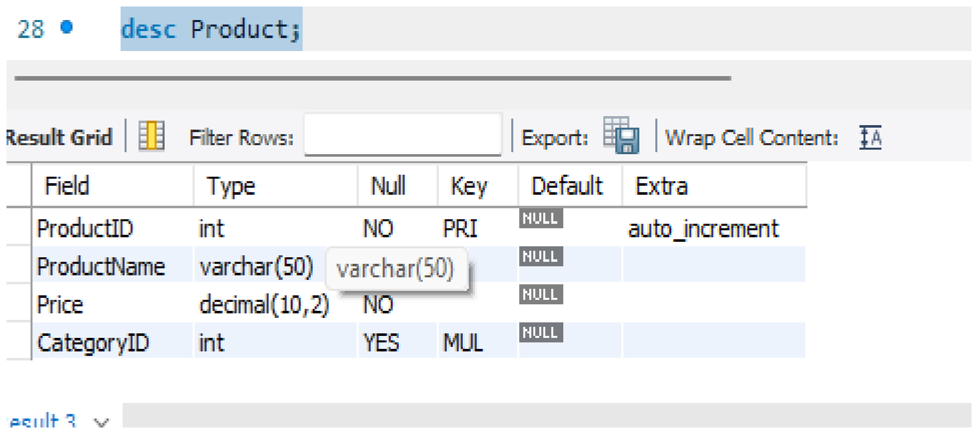
```
desc categories;
```

Field	Type	Null	Key	Default	Extra
CategoryID	int	NO	PRI	NULL	auto_increment
CategoryName	varchar(50)	YES		NULL	

Product-



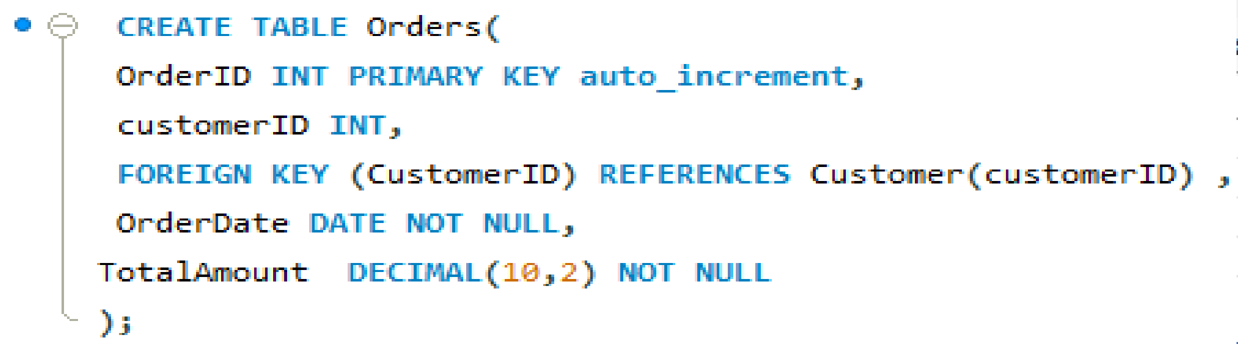
```
CREATE TABLE Product (
    ProductID INT PRIMARY KEY AUTO_INCREMENT,
    ProductName VARCHAR(50),
    Price DECIMAL(10,2) NOT NULL,
    CategoryID INT,
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);
```



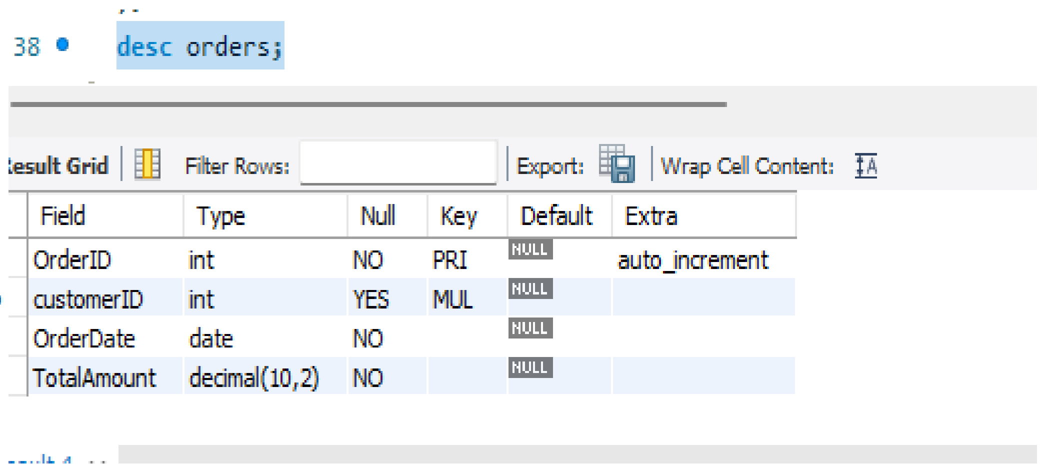
```
desc Product;
```

Field	Type	Null	Key	Default	Extra
ProductID	int	NO	PRI	NULL	auto_increment
ProductName	varchar(50)			NULL	
Price	decimal(10,2)	NO		NULL	
CategoryID	int	YES	MUL	NULL	

Orders-



```
CREATE TABLE Orders(
OrderID INT PRIMARY KEY auto_increment,
customerID INT,
FOREIGN KEY (CustomerID) REFERENCES Customer(customerID) ,
OrderDate DATE NOT NULL,
TotalAmount  DECIMAL(10,2) NOT NULL
);
```

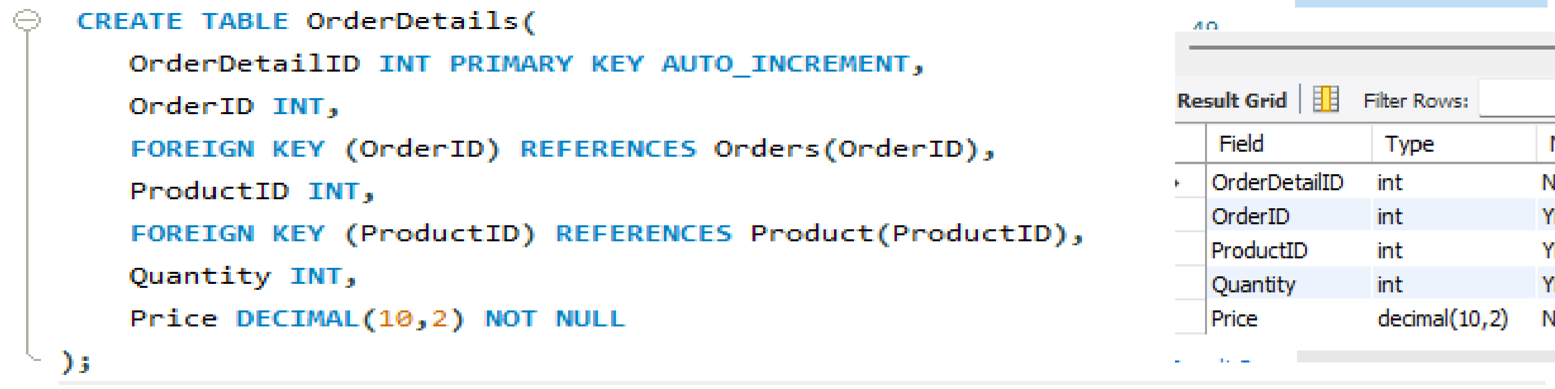


```
desc orders;
```

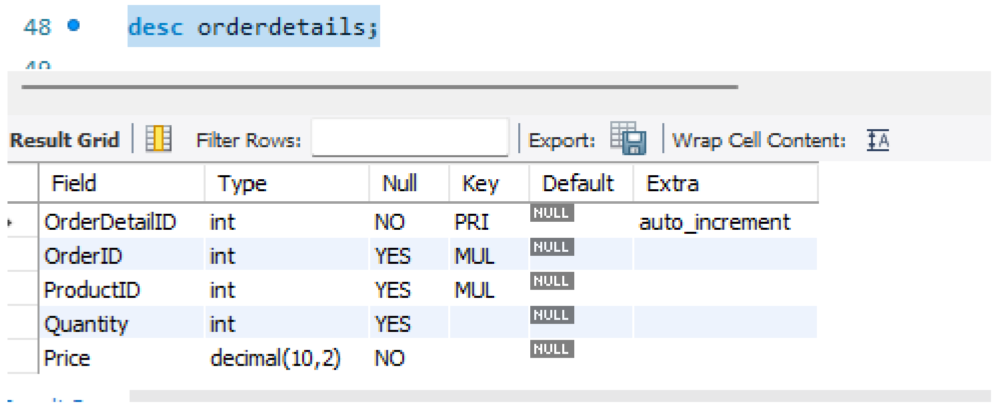
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
OrderID	int	NO	PRI	NULL	auto_increment
customerID	int	YES	MUL	NULL	
OrderDate	date	NO		NULL	
TotalAmount	decimal(10,2)	NO		NULL	

Order Details-



```
CREATE TABLE OrderDetails(
    OrderDetailID INT PRIMARY KEY AUTO_INCREMENT,
    OrderID INT,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    ProductID INT,
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID),
    Quantity INT,
    Price DECIMAL(10,2) NOT NULL
);
```

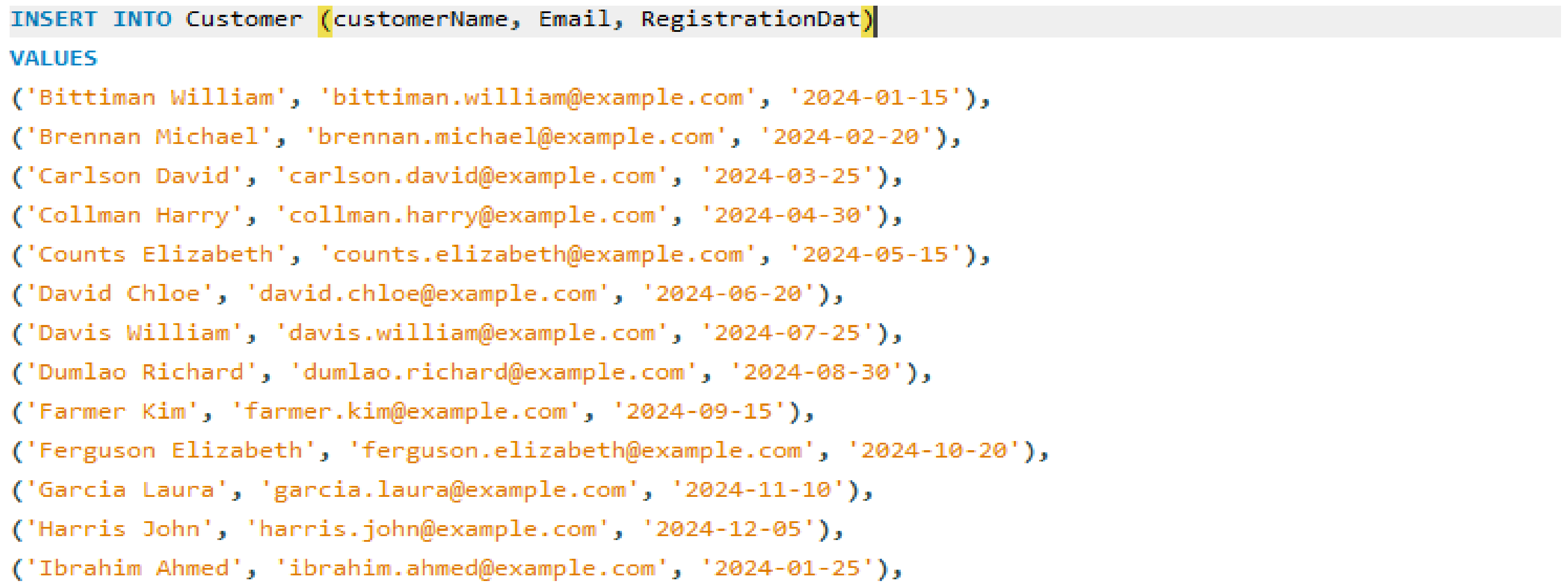


```
desc orderdetails;
```

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

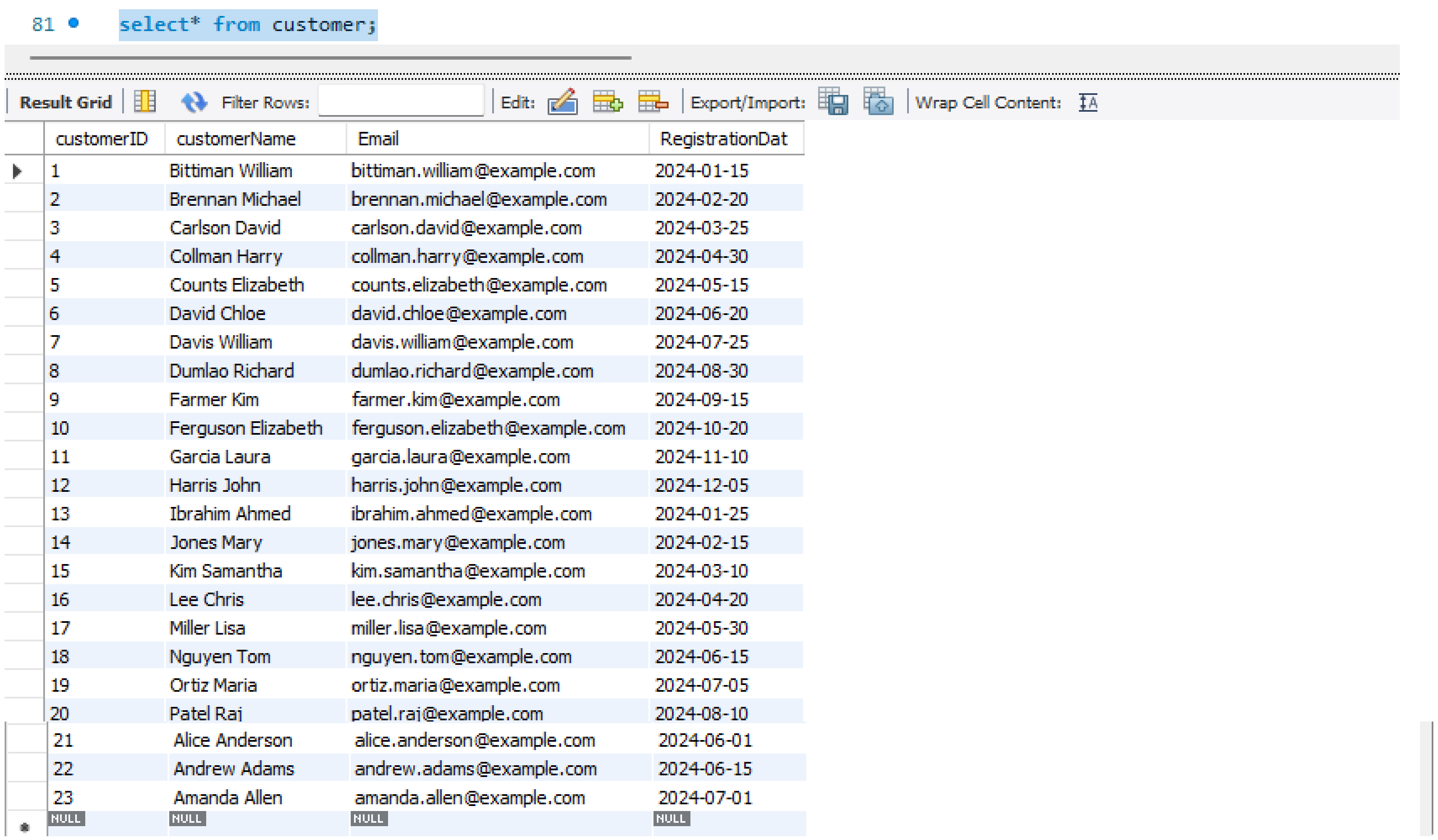
Field	Type	Null	Key	Default	Extra
OrderDetailID	int	NO	PRI	NULL	auto_increment
OrderID	int	YES	MUL	NULL	
ProductID	int	YES	MUL	NULL	
Quantity	int	YES		NULL	
Price	decimal(10,2)	NO		NULL	

- Insert records into each table to populate the database with sample data.



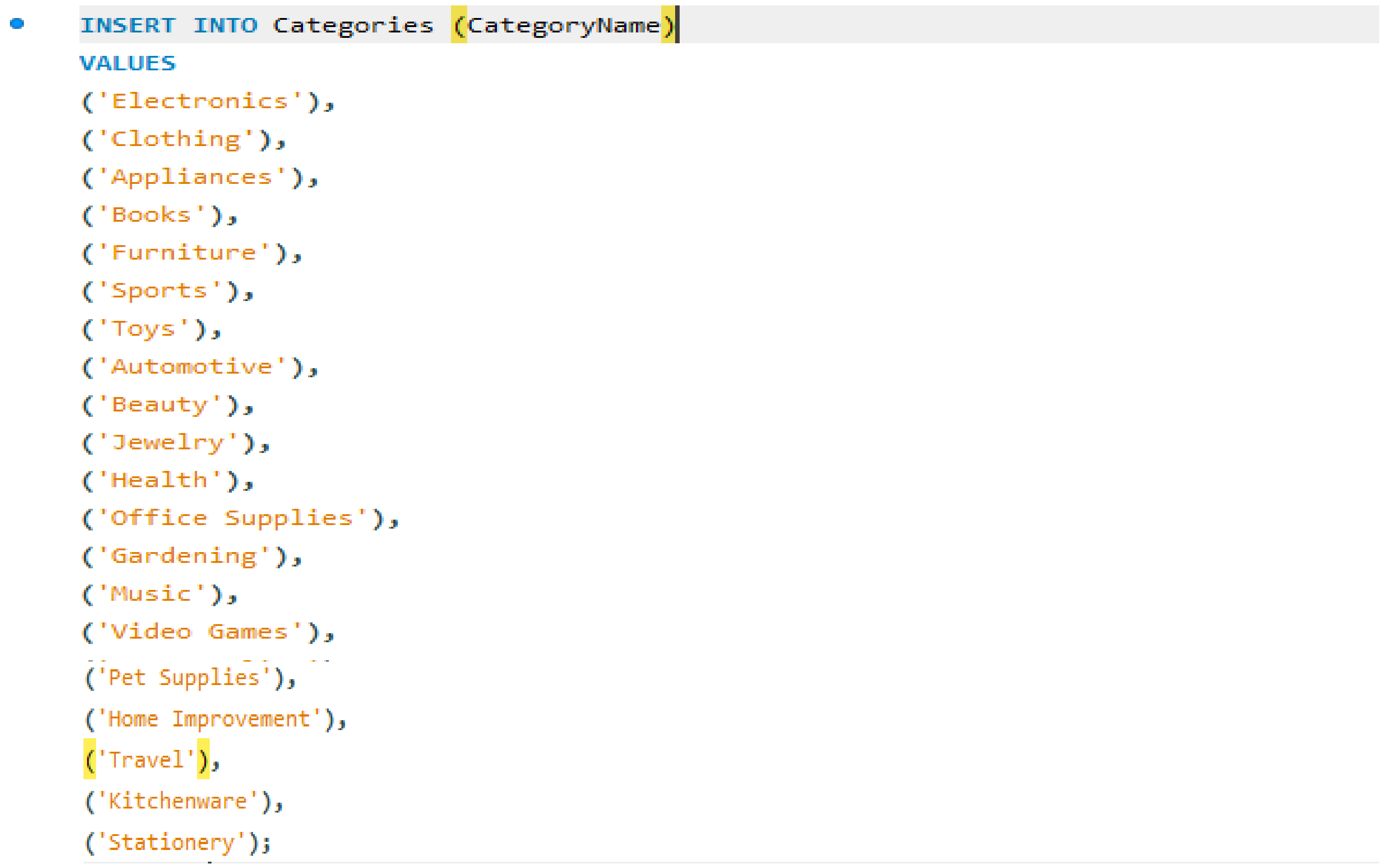
```sql
INSERT INTO Customer (customerName, Email, RegistrationDat)
VALUES
('Bittiman William', 'bittiman.william@example.com', '2024-01-15'),
('Brennan Michael', 'brennan.michael@example.com', '2024-02-20'),
('Carlson David', 'carlson.david@example.com', '2024-03-25'),
('Collman Harry', 'collman.harry@example.com', '2024-04-30'),
('Counts Elizabeth', 'counts.elizabeth@example.com', '2024-05-15'),
('David Chloe', 'david.chloe@example.com', '2024-06-20'),
('Davis William', 'davis.william@example.com', '2024-07-25'),
('Dumlao Richard', 'dumlao.richard@example.com', '2024-08-30'),
('Farmer Kim', 'farmer.kim@example.com', '2024-09-15'),
('Ferguson Elizabeth', 'ferguson.elizabeth@example.com', '2024-10-20'),
('Garcia Laura', 'garcia.laura@example.com', '2024-11-10'),
('Harris John', 'harris.john@example.com', '2024-12-05'),
('Ibrahim Ahmed', 'ibrahim.ahmed@example.com', '2024-01-25'),
('Jones Mary', 'jones.mary@example.com', '2024-02-15'),
('Kim Samantha', 'kim.samantha@example.com', '2024-03-10'),
('Lee Chris', 'lee.chris@example.com', '2024-04-20'),
('Miller Lisa', 'miller.lisa@example.com', '2024-05-30'),
('Nguyen Tom', 'nguyen.tom@example.com', '2024-06-15'),
('Ortiz Maria', 'ortiz.maria@example.com', '2024-07-05'),
('Patel Raj', 'patel.raj@example.com', '2024-08-10'),
('Alice Anderson', 'alice.anderson@example.com', '2024-06-01'),
('Andrew Adams', 'andrew.adams@example.com', '2024-06-15'),
('Amanda Allen', 'amanda.allen@example.com', '2024-07-01');
```

-

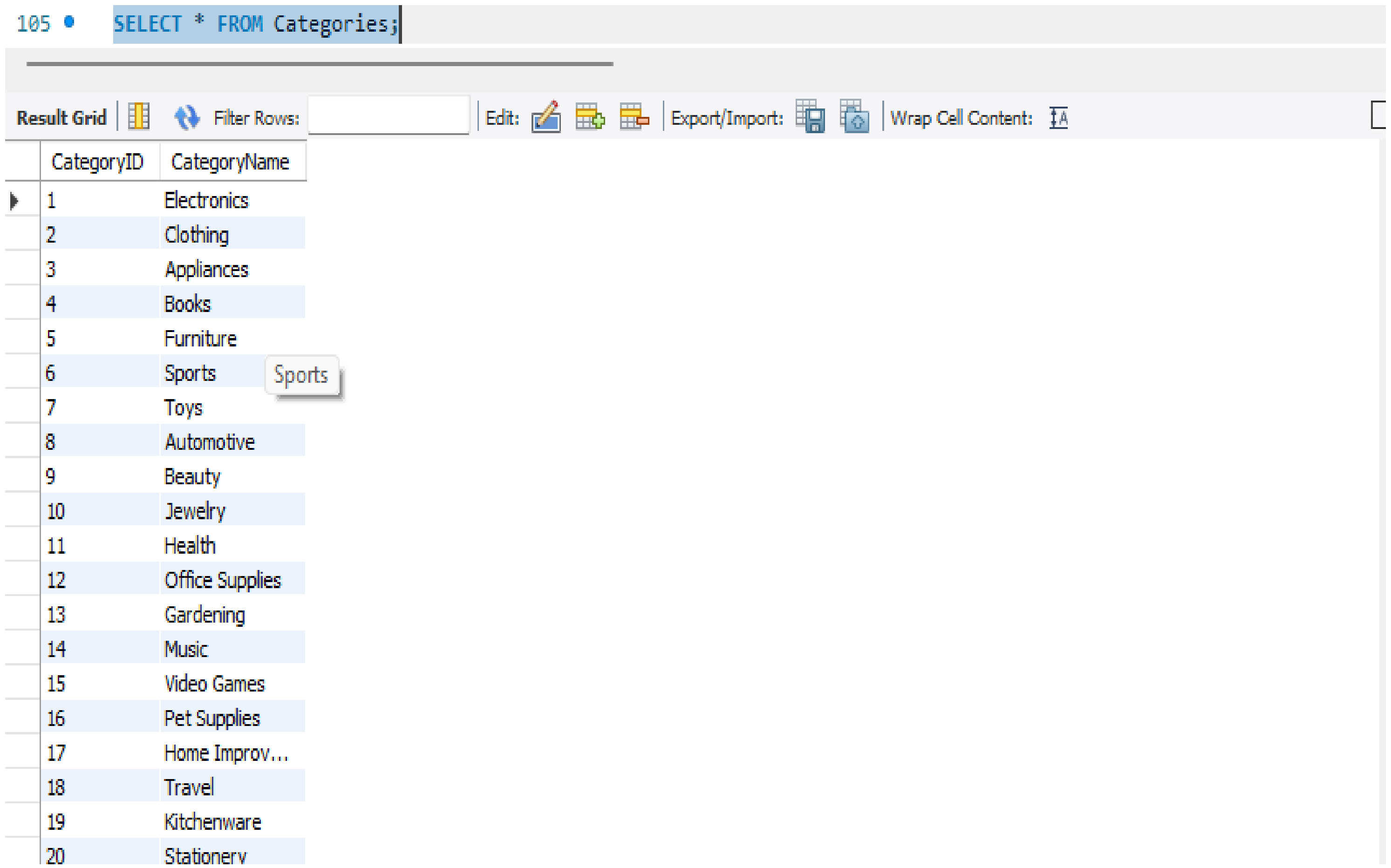


```sql
select* from customer;
```

customerID	customerName	Email	RegistrationDat
1	Bittiman William	bittiman.william@example.com	2024-01-15
2	Brennan Michael	brennan.michael@example.com	2024-02-20
3	Carlson David	carlson.david@example.com	2024-03-25
4	Collman Harry	collman.harry@example.com	2024-04-30
5	Counts Elizabeth	counts.elizabeth@example.com	2024-05-15
6	David Chloe	david.chloe@example.com	2024-06-20
7	Davis William	davis.william@example.com	2024-07-25
8	Dumlao Richard	dumlao.richard@example.com	2024-08-30
9	Farmer Kim	farmer.kim@example.com	2024-09-15
10	Ferguson Elizabeth	ferguson.elizabeth@example.com	2024-10-20
11	Garcia Laura	garcia.laura@example.com	2024-11-10
12	Harris John	harris.john@example.com	2024-12-05
13	Ibrahim Ahmed	ibrahim.ahmed@example.com	2024-01-25
14	Jones Mary	jones.mary@example.com	2024-02-15
15	Kim Samantha	kim.samantha@example.com	2024-03-10
16	Lee Chris	lee.chris@example.com	2024-04-20
17	Miller Lisa	miller.lisa@example.com	2024-05-30
18	Nguyen Tom	nguyen.tom@example.com	2024-06-15
19	Ortiz Maria	ortiz.maria@example.com	2024-07-05
20	Patel Raj	patel.raj@example.com	2024-08-10
21	Alice Anderson	alice.anderson@example.com	2024-06-01
22	Andrew Adams	andrew.adams@example.com	2024-06-15
23	Amanda Allen	amanda.allen@example.com	2024-07-01
NULL	NULL	NULL	NULL



```sql
INSERT INTO Categories (CategoryName)
VALUES
('Electronics'),
('Clothing'),
('Appliances'),
('Books'),
('Furniture'),
('Sports'),
('Toys'),
('Automotive'),
('Beauty'),
('Jewelry'),
('Health'),
('Office Supplies'),
('Gardening'),
('Music'),
('Video Games'),
('Pet Supplies'),
('Home Improvement'),
('Travel'),
('Kitchenware'),
('Stationery');
```

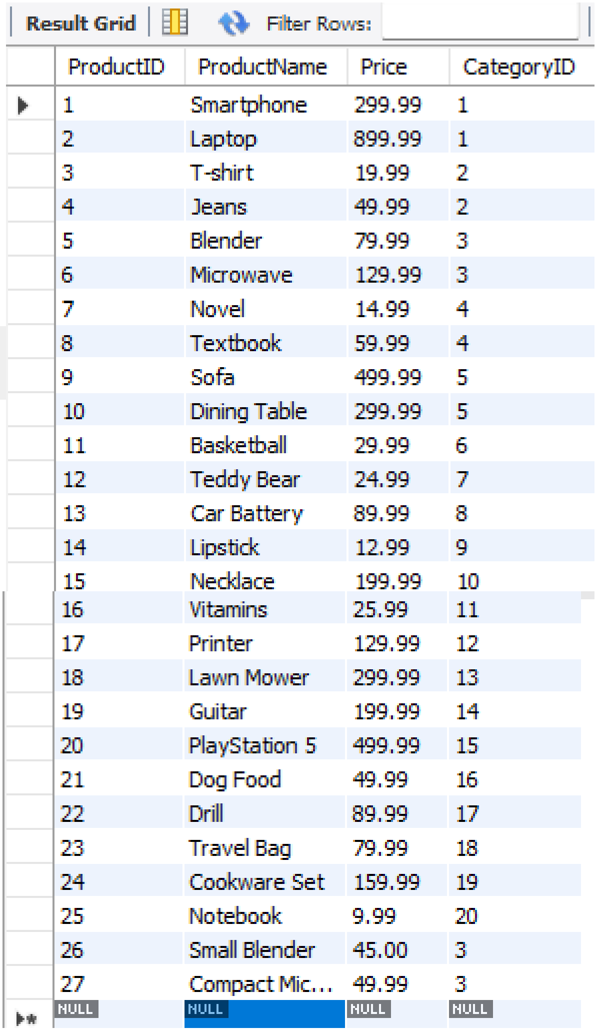


```
SELECT * FROM Categories;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

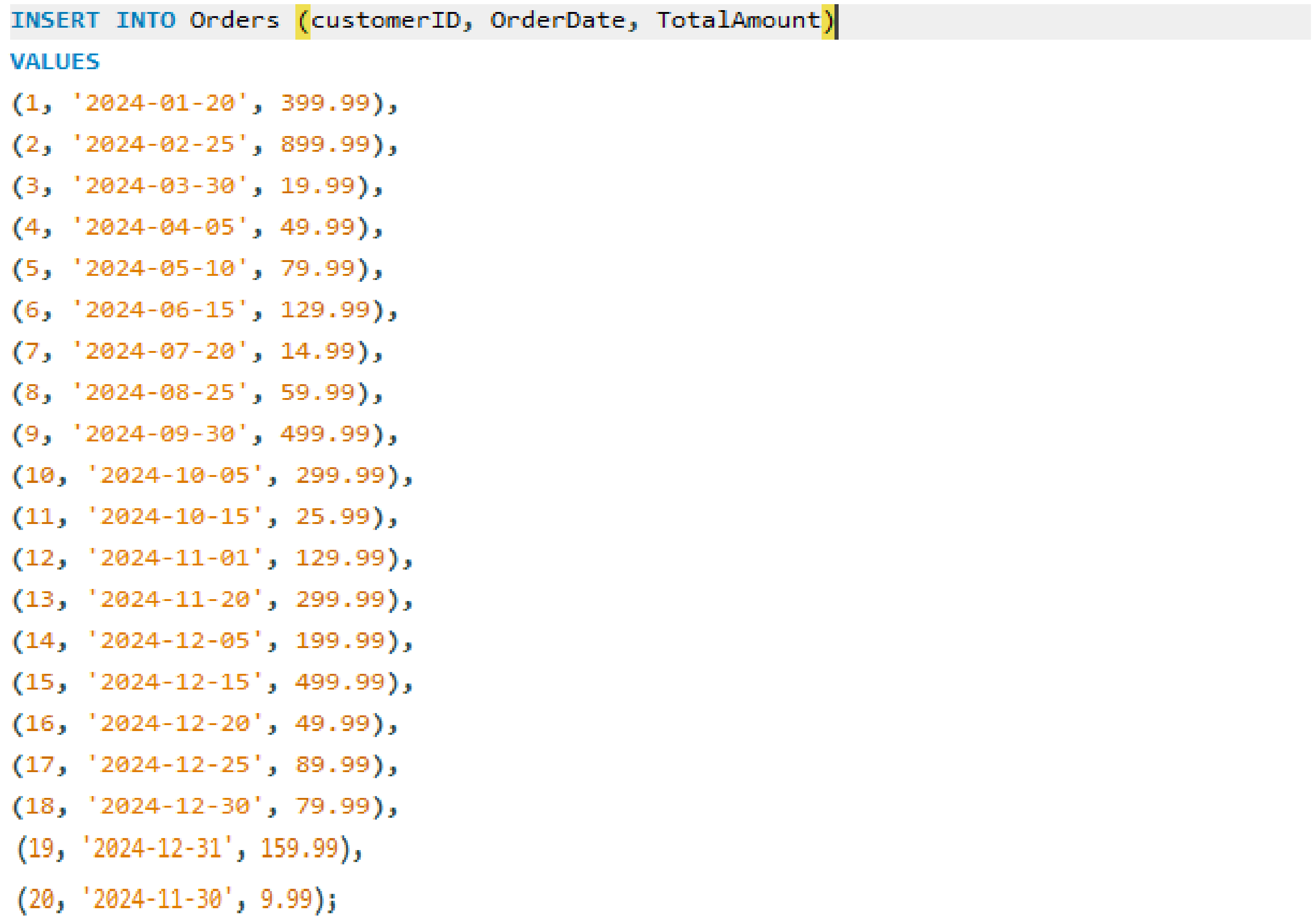
CategoryID	CategoryName
1	Electronics
2	Clothing
3	Appliances
4	Books
5	Furniture
6	Sports
7	Toys
8	Automotive
9	Beauty
10	Jewelry
11	Health
12	Office Supplies
13	Gardening
14	Music
15	Video Games
16	Pet Supplies
17	Home Improv...
18	Travel
19	Kitchenware
20	Stationery

```
INSERT INTO Product (ProductName, Price, CategoryID)
VALUES
('Smartphone', 299.99, 1),
('Laptop', 899.99, 1),
('T-shirt', 19.99, 2),
('Jeans', 49.99, 2),
('Blender', 79.99, 3),
('Microwave', 129.99, 3),
('Novel', 14.99, 4),
('Textbook', 59.99, 4),
('Sofa', 499.99, 5),
('Dining Table', 299.99, 5),
('Basketball', 29.99, 6),
('Teddy Bear', 24.99, 7),
('Car Battery', 89.99, 8),
('Lipstick', 12.99, 9),
('Necklace', 199.99, 10),
('Vitamins', 25.99, 11),
('Printer', 129.99, 12),
('Lawn Mower', 299.99, 13),
('Guitar', 199.99, 14),
('PlayStation 5', 499.99, 15),
('Dog Food', 49.99, 16),
('Drill', 89.99, 17),
('Travel Bag', 79.99, 18),
('Cookware Set', 159.99, 19),
('Notebook', 9.99, 20),
('Small Blender', 45.00, 3),
('Compact Microwave', 49.99, 3);
```

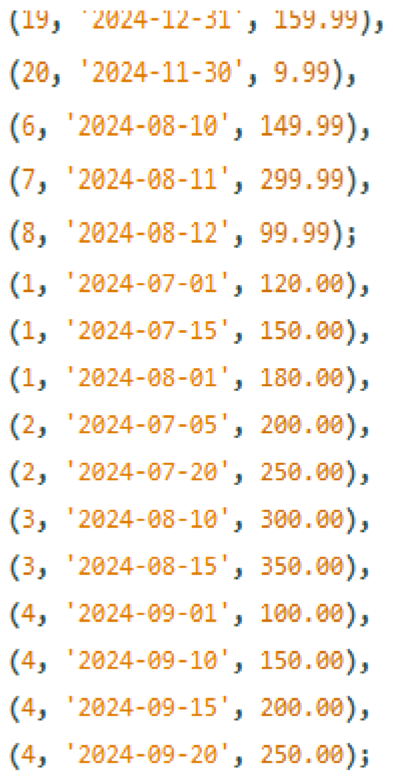


Result Grid | Filter Rows:

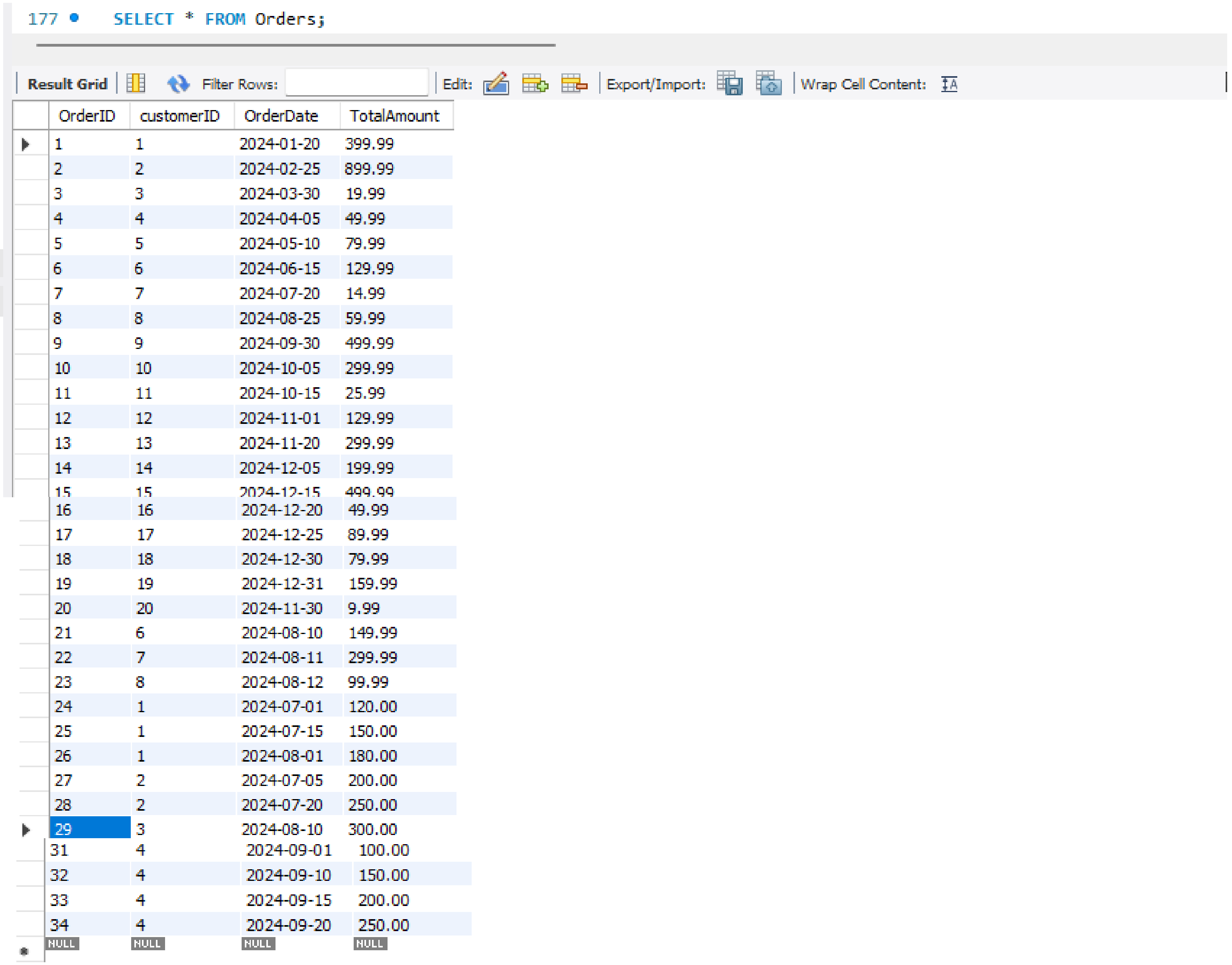
ProductID	ProductName	Price	CategoryID
1	Smartphone	299.99	1
2	Laptop	899.99	1
3	T-shirt	19.99	2
4	Jeans	49.99	2
5	Blender	79.99	3
6	Microwave	129.99	3
7	Novel	14.99	4
8	Textbook	59.99	4
9	Sofa	499.99	5
10	Dining Table	299.99	5
11	Basketball	29.99	6
12	Teddy Bear	24.99	7
13	Car Battery	89.99	8
14	Lipstick	12.99	9
15	Necklace	199.99	10
16	Vitamins	25.99	11
17	Printer	129.99	12
18	Lawn Mower	299.99	13
19	Guitar	199.99	14
20	PlayStation 5	499.99	15
21	Dog Food	49.99	16
22	Drill	89.99	17
23	Travel Bag	79.99	18
24	Cookware Set	159.99	19
25	Notebook	9.99	20
26	Small Blender	45.00	3
27	Compact Mic...	49.99	3
NULL	NULL	NULL	NULL



```sql
INSERT INTO Orders (customerID, OrderDate, TotalAmount)
VALUES
(1, '2024-01-20', 399.99),
(2, '2024-02-25', 899.99),
(3, '2024-03-30', 19.99),
(4, '2024-04-05', 49.99),
(5, '2024-05-10', 79.99),
(6, '2024-06-15', 129.99),
(7, '2024-07-20', 14.99),
(8, '2024-08-25', 59.99),
(9, '2024-09-30', 499.99),
(10, '2024-10-05', 299.99),
(11, '2024-10-15', 25.99),
(12, '2024-11-01', 129.99),
(13, '2024-11-20', 299.99),
(14, '2024-12-05', 199.99),
(15, '2024-12-15', 499.99),
(16, '2024-12-20', 49.99),
(17, '2024-12-25', 89.99),
(18, '2024-12-30', 79.99),
(19, '2024-12-31', 159.99),
(20, '2024-11-30', 9.99);
```



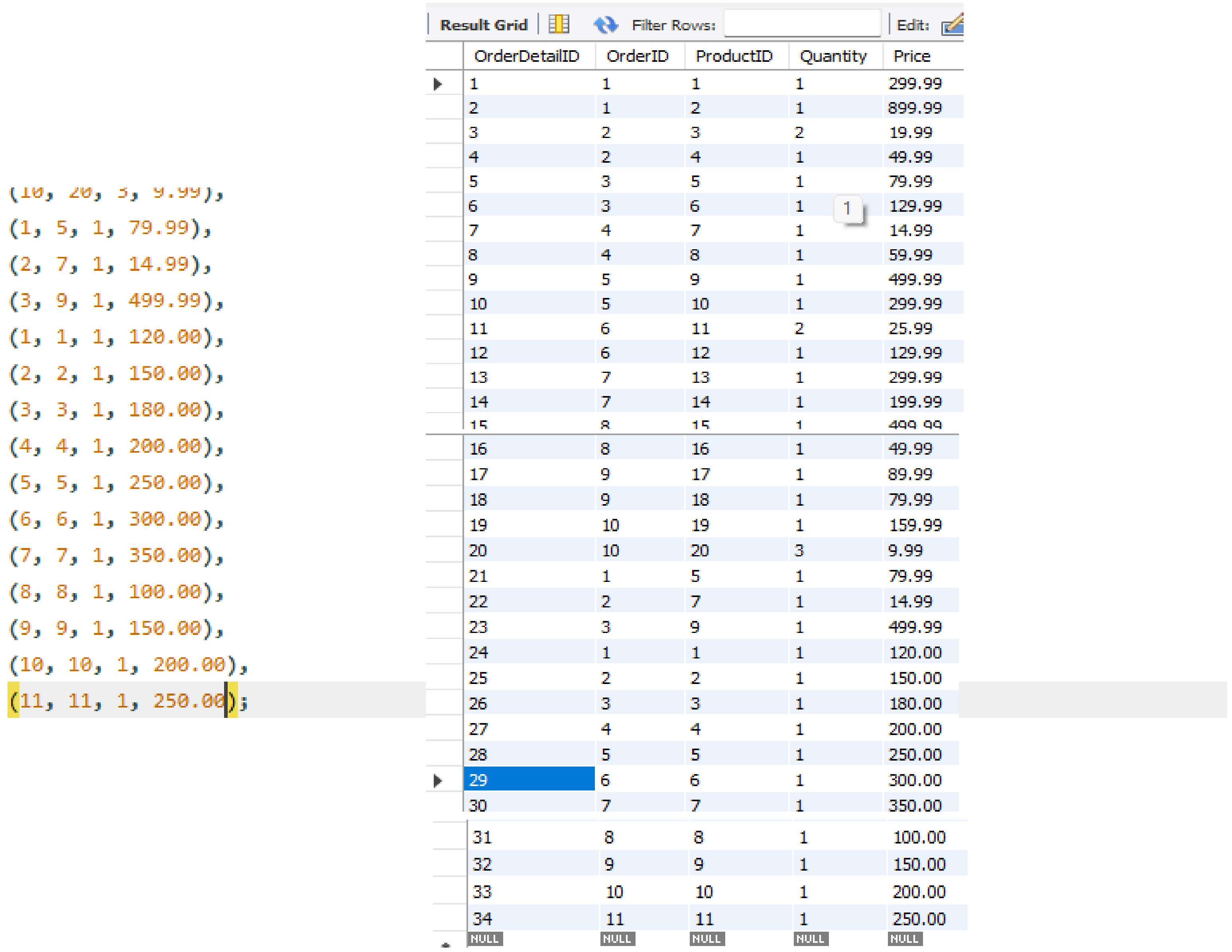
```
(19, '2024-12-31', 159.99),
(20, '2024-11-30', 9.99),
(6, '2024-08-10', 149.99),
(7, '2024-08-11', 299.99),
(8, '2024-08-12', 99.99);
(1, '2024-07-01', 120.00),
(1, '2024-07-15', 150.00),
(1, '2024-08-01', 180.00),
(2, '2024-07-05', 200.00),
(2, '2024-07-20', 250.00),
(3, '2024-08-10', 300.00),
(3, '2024-08-15', 350.00),
(4, '2024-09-01', 100.00),
(4, '2024-09-10', 150.00),
(4, '2024-09-15', 200.00),
(4, '2024-09-20', 250.00);
```



```
SELECT * FROM Orders;
```

OrderID	customerID	OrderDate	TotalAmount
1	1	2024-01-20	399.99
2	2	2024-02-25	899.99
3	3	2024-03-30	19.99
4	4	2024-04-05	49.99
5	5	2024-05-10	79.99
6	6	2024-06-15	129.99
7	7	2024-07-20	14.99
8	8	2024-08-25	59.99
9	9	2024-09-30	499.99
10	10	2024-10-05	299.99
11	11	2024-10-15	25.99
12	12	2024-11-01	129.99
13	13	2024-11-20	299.99
14	14	2024-12-05	199.99
15	15	2024-12-15	499.99
16	16	2024-12-20	49.99
17	17	2024-12-25	89.99
18	18	2024-12-30	79.99
19	19	2024-12-31	159.99
20	20	2024-11-30	9.99
21	6	2024-08-10	149.99
22	7	2024-08-11	299.99
23	8	2024-08-12	99.99
24	1	2024-07-01	120.00
25	1	2024-07-15	150.00
26	1	2024-08-01	180.00
27	2	2024-07-05	200.00
28	2	2024-07-20	250.00
29	3	2024-08-10	300.00
31	4	2024-09-01	100.00
32	4	2024-09-10	150.00
33	4	2024-09-15	200.00
34	4	2024-09-20	250.00
NULL	NULL	NULL	NULL

-
  ```
  INSERT INTO OrderDetails (OrderID, ProductID, Quantity, Price)
  VALUES
  (1, 1, 1, 299.99),
  (1, 2, 1, 899.99),
  (2, 3, 2, 19.99),
  (2, 4, 1, 49.99),
  (3, 5, 1, 79.99),
  (3, 6, 1, 129.99),
  (4, 7, 1, 14.99),
  (4, 8, 1, 59.99),
  (5, 9, 1, 499.99),
  (5, 10, 1, 299.99),
  (6, 11, 2, 25.99),
  (6, 12, 1, 129.99),
  (7, 13, 1, 299.99),
  (7, 14, 1, 199.99),
  (8, 15, 1, 499.99),
  (8, 16, 1, 49.99),
  (9, 17, 1, 89.99),
  (9, 18, 1, 79.99),
  (10, 19, 1, 159.99),
  ```

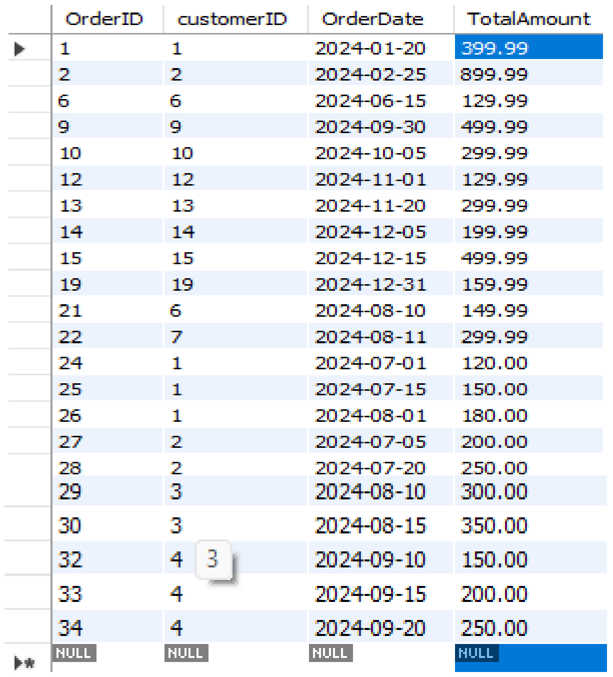


```
(10, 20, 3, 9.99),
(1, 5, 1, 79.99),
(2, 7, 1, 14.99),
(3, 9, 1, 499.99),
(1, 1, 1, 120.00),
(2, 2, 1, 150.00),
(3, 3, 1, 180.00),
(4, 4, 1, 200.00),
(5, 5, 1, 250.00),
(6, 6, 1, 300.00),
(7, 7, 1, 350.00),
(8, 8, 1, 100.00),
(9, 9, 1, 150.00),
(10, 10, 1, 200.00),
(11, 11, 1, 250.00);
```

OrderDetailID	OrderID	ProductID	Quantity	Price
1	1	1	1	299.99
2	1	2	1	899.99
3	2	3	2	19.99
4	2	4	1	49.99
5	3	5	1	79.99
6	3	6	1	129.99
7	4	7	1	14.99
8	4	8	1	59.99
9	5	9	1	499.99
10	5	10	1	299.99
11	6	11	2	25.99
12	6	12	1	129.99
13	7	13	1	299.99
14	7	14	1	199.99
15	8	15	1	499.99
16	8	16	1	49.99
17	9	17	1	89.99
18	9	18	1	79.99
19	10	19	1	159.99
20	10	20	3	9.99
21	1	5	1	79.99
22	2	7	1	14.99
23	3	9	1	499.99
24	1	1	1	120.00
25	2	2	1	150.00
26	3	3	1	180.00
27	4	4	1	200.00
28	5	5	1	250.00
29	6	6	1	300.00
30	7	7	1	350.00
31	8	8	1	100.00
32	9	9	1	150.00
33	10	10	1	200.00
34	11	11	1	250.00
NULL	NULL	NULL	NULL	NULL

- How many high-value orders are there?

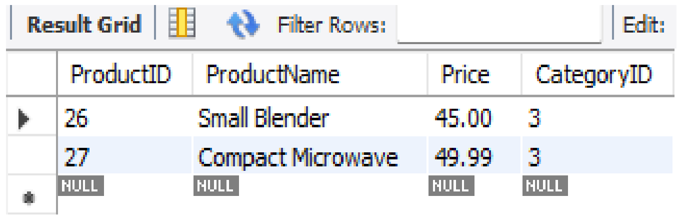
```sql
-- all details from the Orders table where the TotalAmount is greater than $100--
SELECT *
FROM Orders
WHERE TotalAmount > 100;
```



OrderID	customerID	OrderDate	TotalAmount
1	1	2024-01-20	399.99
2	2	2024-02-25	899.99
6	6	2024-06-15	129.99
9	9	2024-09-30	499.99
10	10	2024-10-05	299.99
12	12	2024-11-01	129.99
13	13	2024-11-20	299.99
14	14	2024-12-05	199.99
15	15	2024-12-15	499.99
19	19	2024-12-31	159.99
21	6	2024-08-10	149.99
22	7	2024-08-11	299.99
24	1	2024-07-01	120.00
25	1	2024-07-15	150.00
26	1	2024-08-01	180.00
27	2	2024-07-05	200.00
28	2	2024-07-20	250.00
29	3	2024-08-10	300.00
30	3	2024-08-15	350.00
32	4	2024-09-10	150.00
33	4	2024-09-15	200.00
34	4	2024-09-20	250.00
NULL	NULL	NULL	NULL

- How many products in the 'Appliances' category are priced between $20 and $50?

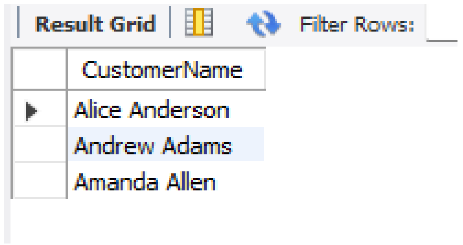
```
-- all products from the Products table where the Price is between $20 and $50 and the CategoryID is 3--
SELECT *
FROM Product
WHERE Price BETWEEN 20 AND 50
 AND CategoryID = 3;
```


Result Grid | Filter Rows: | Edit:

ProductID	ProductName	Price	CategoryID
26	Small Blender	45.00	3
27	Compact Microwave	49.99	3
NULL	NULL	NULL	NULL

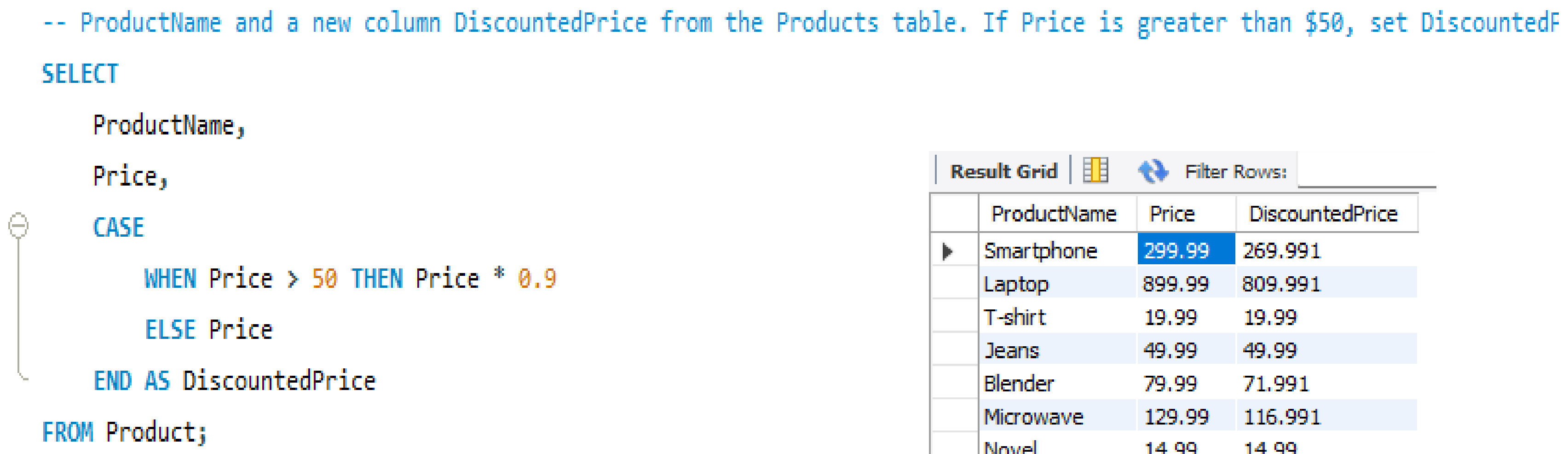
- How many customers have names that start with 'A'?

```
SELECT CustomerName
from Customer
where CustomerName like 'A%';
```

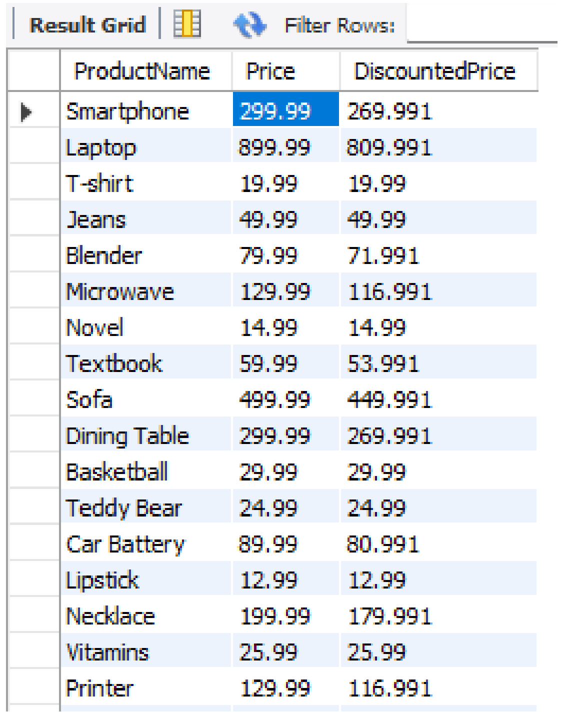

Result Grid | Filter Rows:

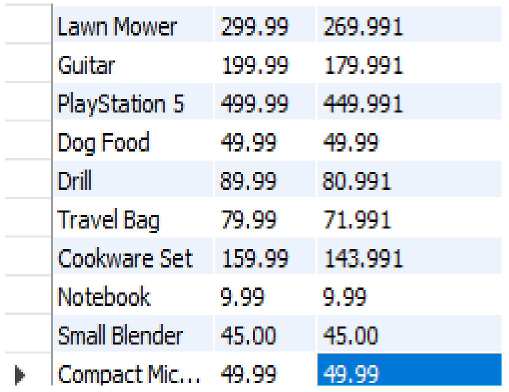
CustomerName
Alice Anderson
Andrew Adams
Amanda Allen

- Which products would receive a discount and what would their discounted prices be?



```
-- ProductName and a new column DiscountedPrice from the Products table. If Price is greater than $50, set DiscountedF
SELECT
    ProductName,
    Price,
    CASE
        WHEN Price > 50 THEN Price * 0.9
        ELSE Price
    END AS DiscountedPrice
FROM Product;
```

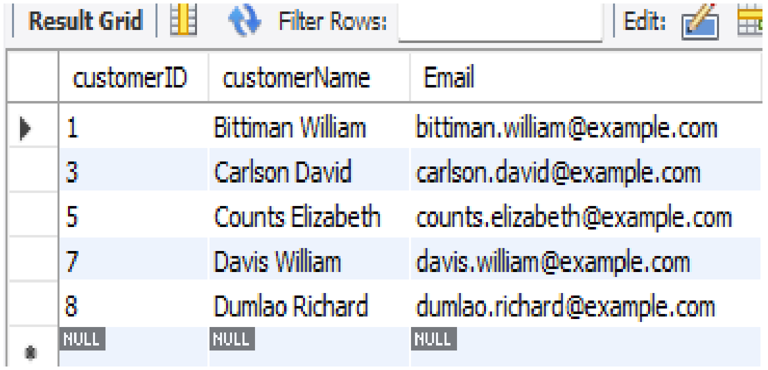




ProductName	Price	DiscountedPrice
Smartphone	299.99	269.991
Laptop	899.99	809.991
T-shirt	19.99	19.99
Jeans	49.99	49.99
Blender	79.99	71.991
Microwave	129.99	116.991
Novel	14.99	14.99
Textbook	59.99	53.991
Sofa	499.99	449.991
Dining Table	299.99	269.991
Basketball	29.99	29.99
Teddy Bear	24.99	24.99
Car Battery	89.99	80.991
Lipstick	12.99	12.99
Necklace	199.99	179.991
Vitamins	25.99	25.99
Printer	129.99	116.991
Lawn Mower	299.99	269.991
Guitar	199.99	179.991
PlayStation 5	499.99	449.991
Dog Food	49.99	49.99
Drill	89.99	80.991
Travel Bag	79.99	71.991
Cookware Set	159.99	143.991
Notebook	9.99	9.99
Small Blender	45.00	45.00
Compact Mic...	49.99	49.99

- Who are the most expending customers

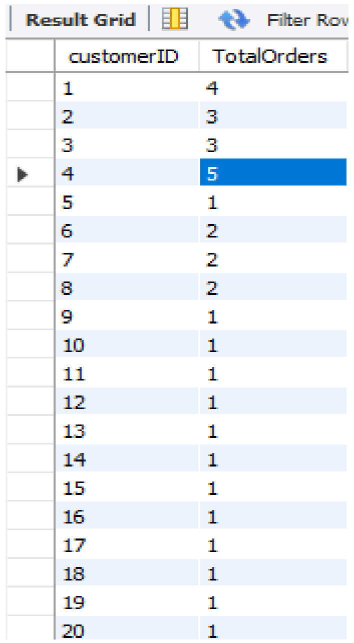
```
-- all customers who have placed orders totaling more than $500.
SELECT DISTINCT c.customerID, c.customerName, c.Email
FROM Customer c
WHERE c.customerID IN (
    SELECT o.customerID
    FROM Orders o
    JOIN OrderDetails od ON o.OrderID = od.OrderID
    GROUP BY o.customerID
    HAVING SUM(od.Price * od.Quantity) > 500
);
```



customerID	customerName	Email
1	Bittiman William	bittiman.william@example.com
3	Carlson David	carlson.david@example.com
5	Counts Elizabeth	counts.elizabeth@example.com
7	Davis William	davis.william@example.com
8	Dumlao Richard	dumlao.richard@example.com
NULL	NULL	NULL

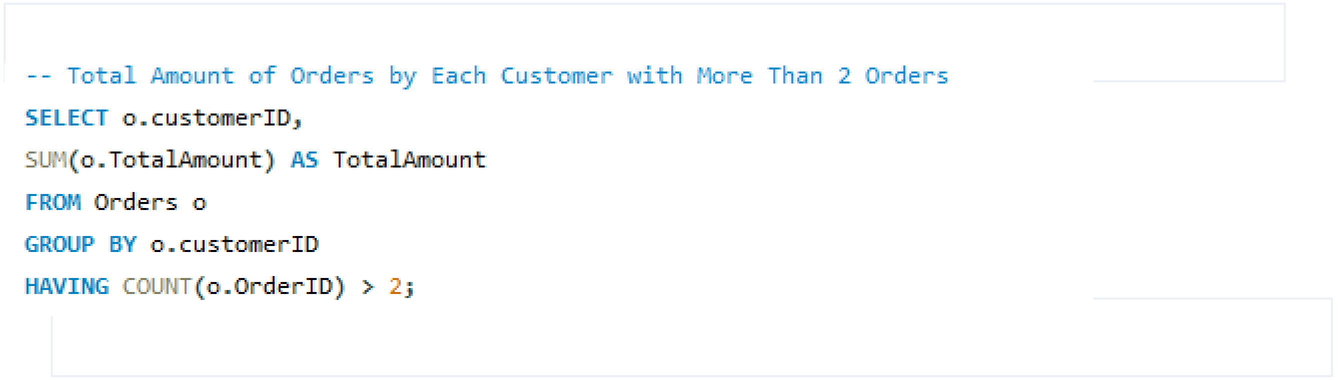
- Which customer has placed the most orders?

```
-- the total number of orders placed by each customer
SELECT
    o.customerID,
    COUNT(o.OrderID) AS TotalOrders
FROM Orders o
GROUP BY o.customerID;
```

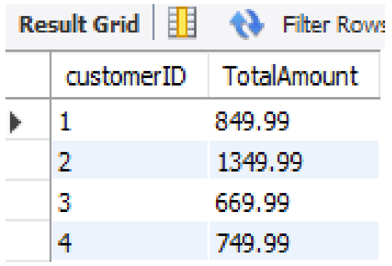


customerID	TotalOrders
1	4
2	3
3	3
4	5
5	1
6	2
7	2
8	2
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1

- What is the total amount spent by customers with more than 2 orders?

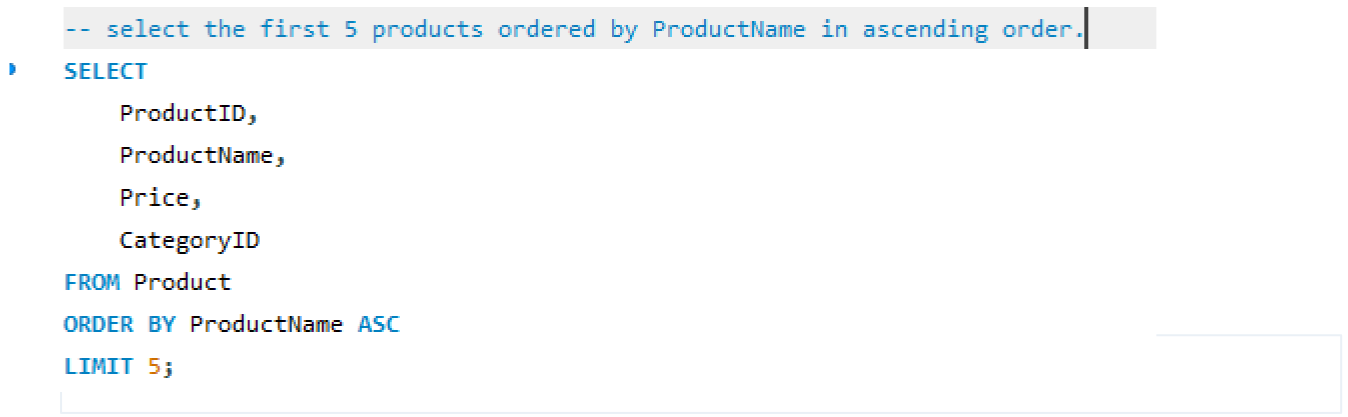


```sql
-- Total Amount of Orders by Each Customer with More Than 2 Orders
SELECT o.customerID,
SUM(o.TotalAmount) AS TotalAmount
FROM Orders o
GROUP BY o.customerID
HAVING COUNT(o.OrderID) > 2;
```

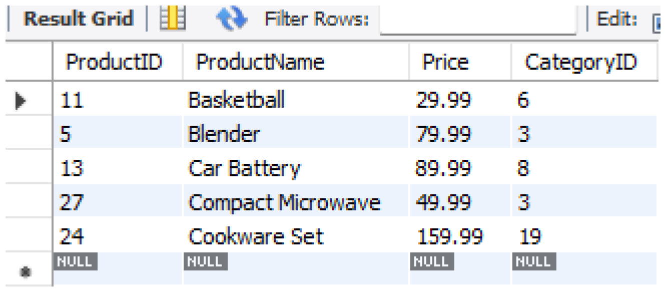


customerID	TotalAmount
1	849.99
2	1349.99
3	669.99
4	749.99

- What are the first 5 products ordered alphabetically?

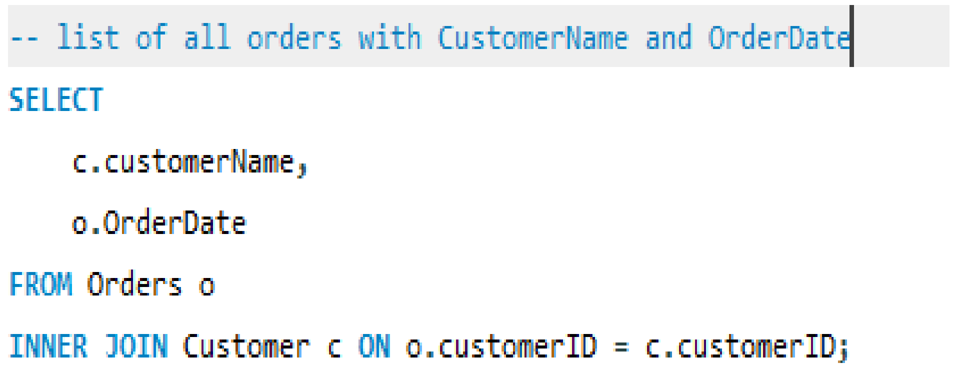


```sql
-- select the first 5 products ordered by ProductName in ascending order.
SELECT
    ProductID,
    ProductName,
    Price,
    CategoryID
FROM Product
ORDER BY ProductName ASC
LIMIT 5;
```

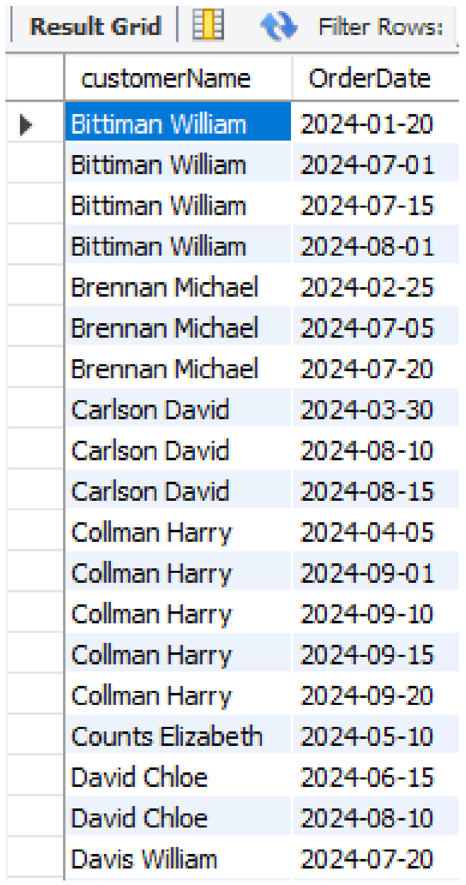


ProductID	ProductName	Price	CategoryID
11	Basketball	29.99	6
5	Blender	79.99	3
13	Car Battery	89.99	8
27	Compact Microwave	49.99	3
24	Cookware Set	159.99	19
NULL	NULL	NULL	NULL

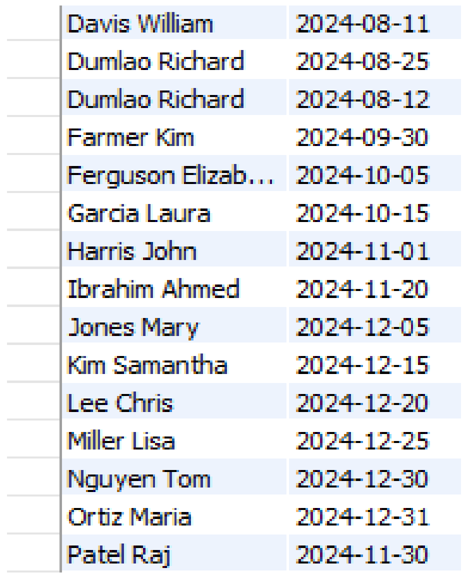
- Which customers placed orders on '2024-07-01'?



```
-- list of all orders with CustomerName and OrderDate
SELECT
    c.customerName,
    o.OrderDate
FROM Orders o
INNER JOIN Customer c ON o.customerID = c.customerID;
```



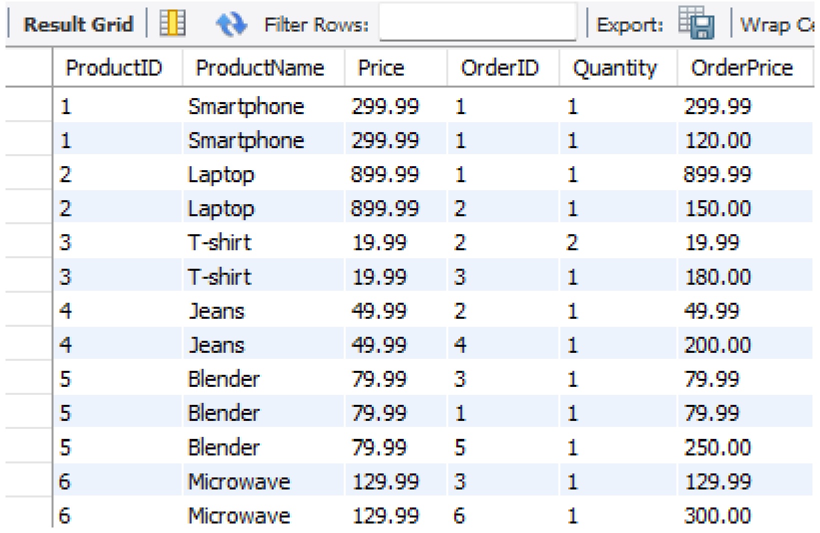
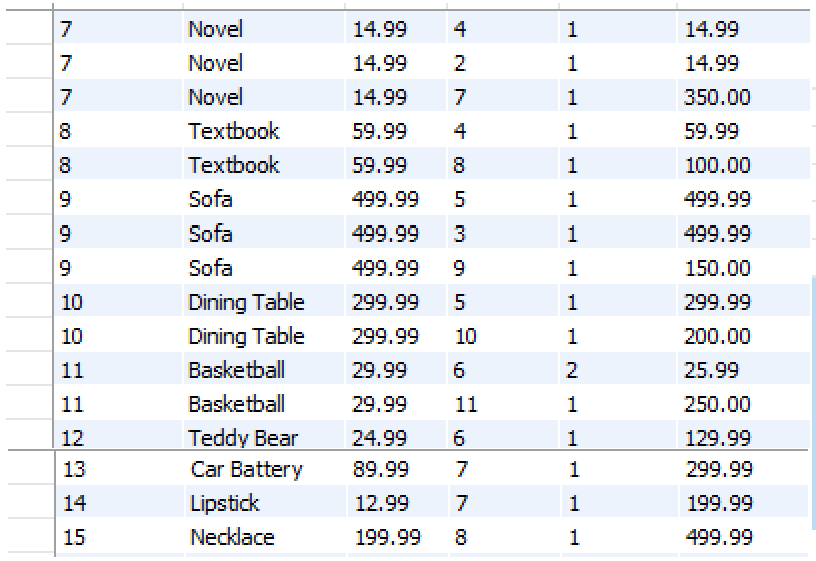
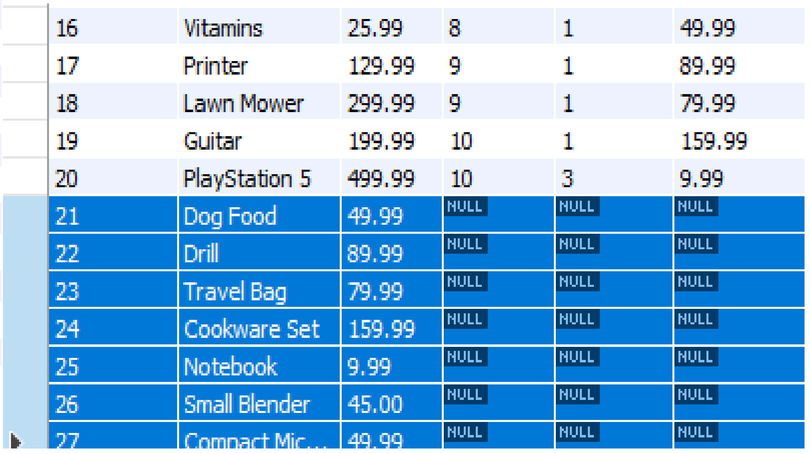
customerName	OrderDate
Bittiman William	2024-01-20
Bittiman William	2024-07-01
Bittiman William	2024-07-15
Bittiman William	2024-08-01
Brennan Michael	2024-02-25
Brennan Michael	2024-07-05
Brennan Michael	2024-07-20
Carlson David	2024-03-30
Carlson David	2024-08-10
Carlson David	2024-08-15
Collman Harry	2024-04-05
Collman Harry	2024-09-01
Collman Harry	2024-09-10
Collman Harry	2024-09-15
Collman Harry	2024-09-20
Counts Elizabeth	2024-05-10
David Chloe	2024-06-15
David Chloe	2024-08-10
Davis William	2024-07-20



Davis William	2024-08-11
Dumlao Richard	2024-08-25
Dumlao Richard	2024-08-12
Farmer Kim	2024-09-30
Ferguson Elizab...	2024-10-05
Garcia Laura	2024-10-15
Harris John	2024-11-01
Ibrahim Ahmed	2024-11-20
Jones Mary	2024-12-05
Kim Samantha	2024-12-15
Lee Chris	2024-12-20
Miller Lisa	2024-12-25
Nguyen Tom	2024-12-30
Ortiz Maria	2024-12-31
Patel Raj	2024-11-30

- How many products have no associated orders?

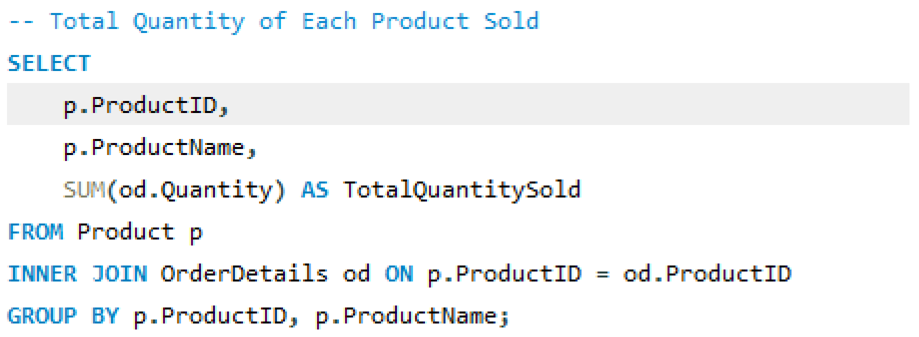
```
--  All Products and Associated Order Detail. Include products that might not have been ordered.
SELECT
    p.ProductID,
    p.ProductName,
    p.Price,
    od.OrderID,
    od.Quantity,
    od.Price AS OrderPrice
FROM Product p
LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID;
```

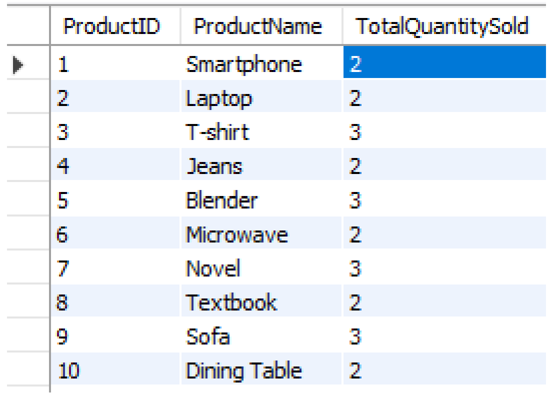
Result Grid | Filter Rows: | Export: | Wrap C

ProductID	ProductName	Price	OrderID	Quantity	OrderPrice
1	Smartphone	299.99	1	1	299.99
1	Smartphone	299.99	1	1	120.00
2	Laptop	899.99	1	1	899.99
2	Laptop	899.99	2	1	150.00
3	T-shirt	19.99	2	2	19.99
3	T-shirt	19.99	3	1	180.00
4	Jeans	49.99	2	1	49.99
4	Jeans	49.99	4	1	200.00
5	Blender	79.99	3	1	79.99
5	Blender	79.99	1	1	79.99
5	Blender	79.99	5	1	250.00
6	Microwave	129.99	3	1	129.99
6	Microwave	129.99	6	1	300.00
7	Novel	14.99	4	1	14.99
7	Novel	14.99	2	1	14.99
7	Novel	14.99	7	1	350.00
8	Textbook	59.99	4	1	59.99
8	Textbook	59.99	8	1	100.00
9	Sofa	499.99	5	1	499.99
9	Sofa	499.99	3	1	499.99
9	Sofa	499.99	9	1	150.00
10	Dining Table	299.99	5	1	299.99
10	Dining Table	299.99	10	1	200.00
11	Basketball	29.99	6	2	25.99
11	Basketball	29.99	11	1	250.00
12	Teddy Bear	24.99	6	1	129.99
13	Car Battery	89.99	7	1	299.99
14	Lipstick	12.99	7	1	199.99
15	Necklace	199.99	8	1	499.99
16	Vitamins	25.99	8	1	49.99
17	Printer	129.99	9	1	89.99
18	Lawn Mower	299.99	9	1	79.99
19	Guitar	199.99	10	1	159.99
20	PlayStation 5	499.99	10	3	9.99
21	Dog Food	49.99	NULL	NULL	NULL
22	Drill	89.99	NULL	NULL	NULL
23	Travel Bag	79.99	NULL	NULL	NULL
24	Cookware Set	159.99	NULL	NULL	NULL
25	Notebook	9.99	NULL	NULL	NULL
26	Small Blender	45.00	NULL	NULL	NULL
27	Compact Mic...	49.99	NULL	NULL	NULL

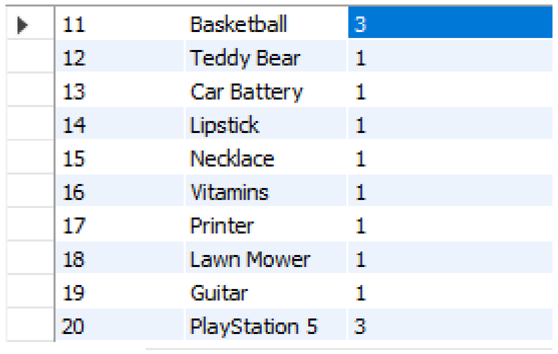
- Which product has the highest total quantity sold?



```
-- Total Quantity of Each Product Sold
SELECT
    p.ProductID,
    p.ProductName,
    SUM(od.Quantity) AS TotalQuantitySold
FROM Product p
INNER JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductID, p.ProductName;
```

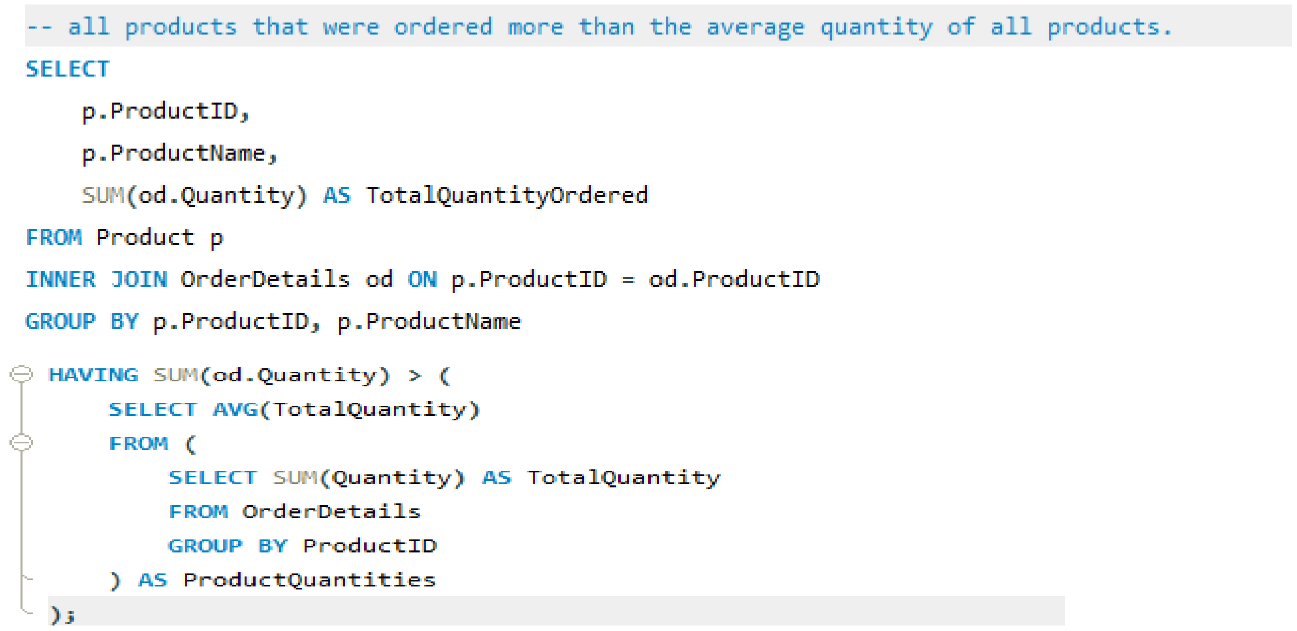


ProductID	ProductName	TotalQuantitySold
1	Smartphone	2
2	Laptop	2
3	T-shirt	3
4	Jeans	2
5	Blender	3
6	Microwave	2
7	Novel	3
8	Textbook	2
9	Sofa	3
10	Dining Table	2



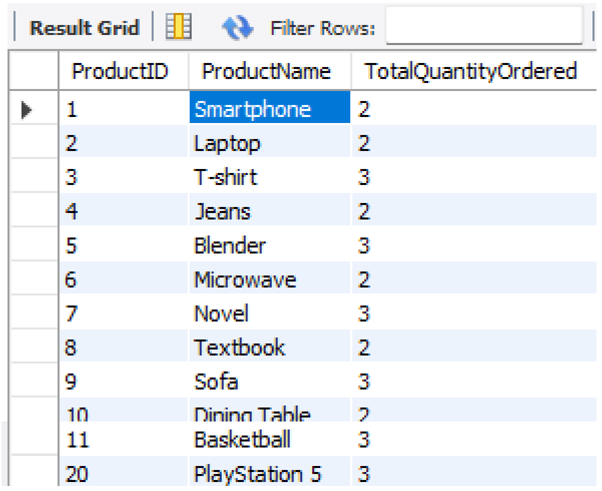
11	Basketball	3
12	Teddy Bear	1
13	Car Battery	1
14	Lipstick	1
15	Necklace	1
16	Vitamins	1
17	Printer	1
18	Lawn Mower	1
19	Guitar	1
20	PlayStation 5	3

- How many products are ordered more than the average quantity



```
-- all products that were ordered more than the average quantity of all products.
SELECT
    p.ProductID,
    p.ProductName,
    SUM(od.Quantity) AS TotalQuantityOrdered
FROM Product p
INNER JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductID, p.ProductName

HAVING SUM(od.Quantity) > (
    SELECT AVG(TotalQuantity)
    FROM (
        SELECT SUM(Quantity) AS TotalQuantity
        FROM OrderDetails
        GROUP BY ProductID
    ) AS ProductQuantities
);
```

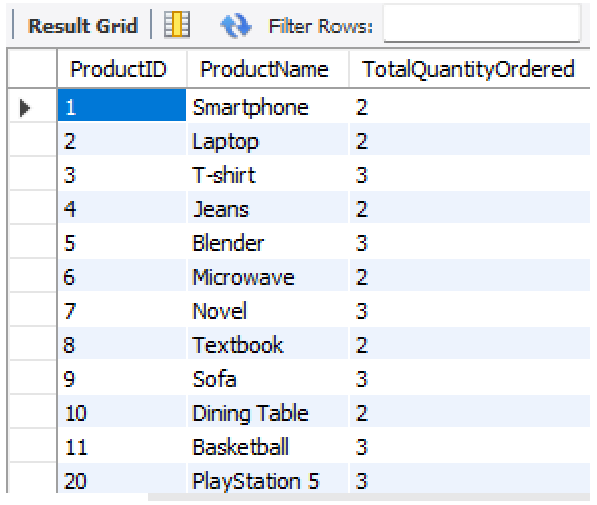


Result Grid | Filter Rows:

ProductID	ProductName	TotalQuantityOrdered
1	Smartphone	2
2	Laptop	2
3	T-shirt	3
4	Jeans	2
5	Blender	3
6	Microwave	2
7	Novel	3
8	Textbook	2
9	Sofa	3
10	Dining Table	2
11	Basketball	3
20	PlayStation 5	3

- Which products have been ordered in quantities greater than the average order quantity?

```
-- list of CustomerName, OrderDate, and ProductName for all orders.
SELECT
    c.customerName,
    o.OrderDate,
    p.ProductName
FROM Orders o
INNER JOIN Customer c ON o.customerID = c.customerID
INNER JOIN OrderDetails od ON o.OrderID = od.OrderID
INNER JOIN Product p ON od.ProductID = p.ProductID;
```



ProductID	ProductName	TotalQuantityOrdered
1	Smartphone	2
2	Laptop	2
3	T-shirt	3
4	Jeans	2
5	Blender	3
6	Microwave	2
7	Novel	3
8	Textbook	2
9	Sofa	3
10	Dining Table	2
11	Basketball	3
20	PlayStation 5	3