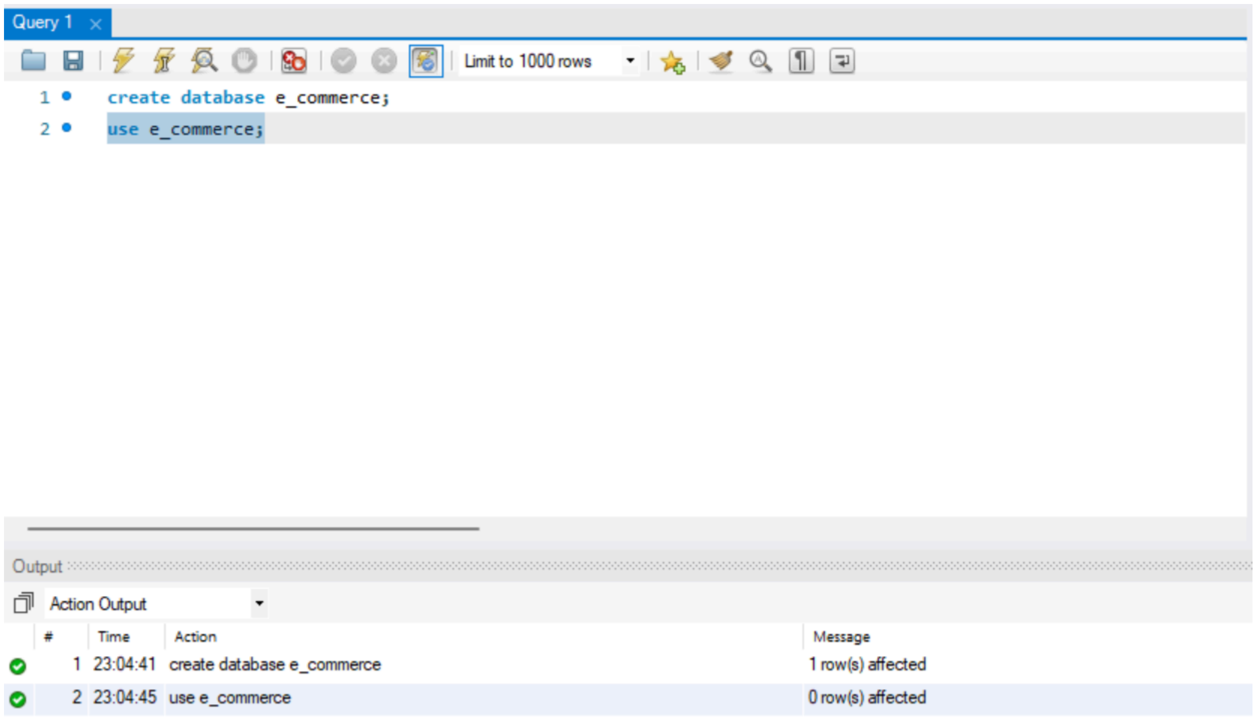
SQL Assignment - 1

1. Create Database e_commerce



```
create database e_commerce;
use e_commerce;
```

```
#   Time      Action                      Message
1   23:04:41  create database e_commerce  1 row(s) affected
2   23:04:45  use e_commerce              0 row(s) affected
```

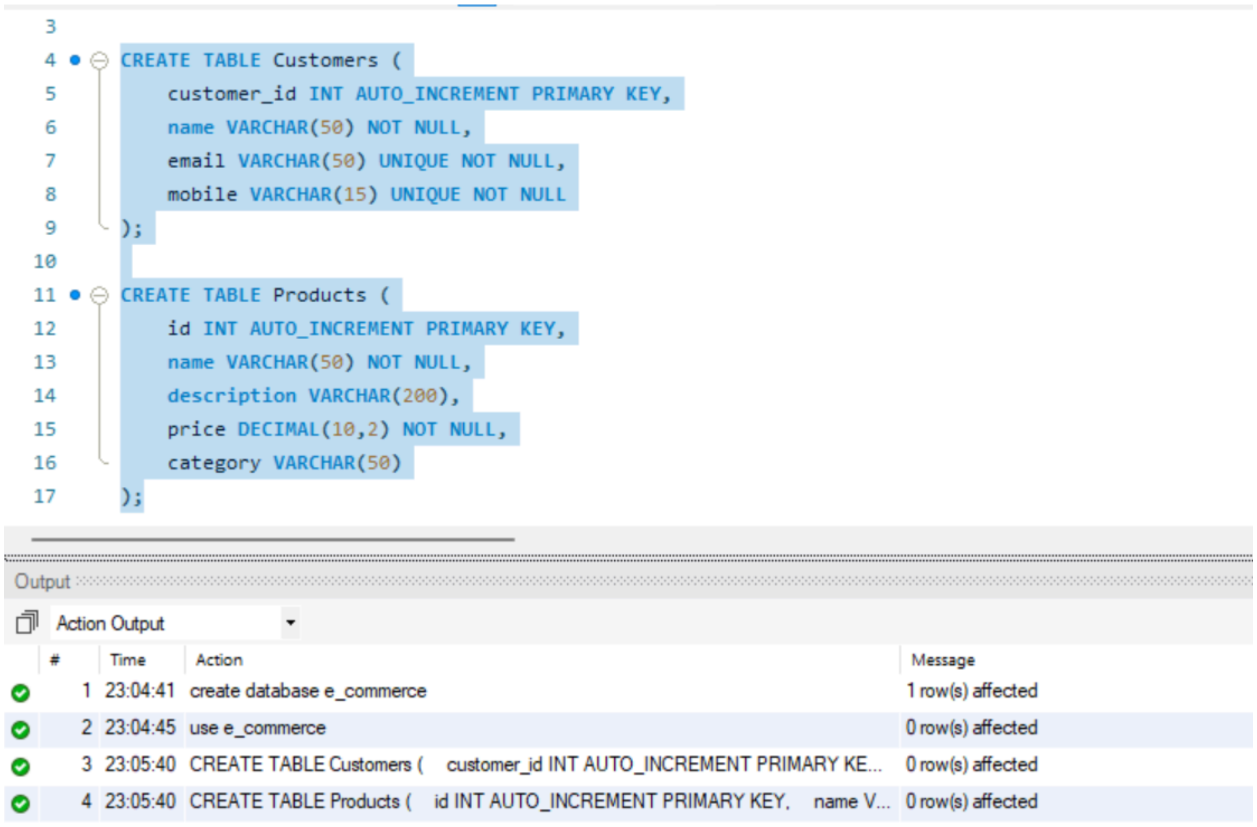
2. Create following Tables:

Customers:

a. customer_id - int auto-increment primary key
b. name - varchar(50)
c. email - varchar(50)
d. mobile - varchar(15)

Products:

a. id - int
b. name - varchar(50) not null
c. description - varchar(200)
d. price - decimal(10, 2) not null
e. category - varchar(50)



```sql
CREATE TABLE Customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    email VARCHAR(50) UNIQUE NOT NULL,
    mobile VARCHAR(15) UNIQUE NOT NULL
);

CREATE TABLE Products (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(200),
    price DECIMAL(10,2) NOT NULL,
    category VARCHAR(50)
);
```

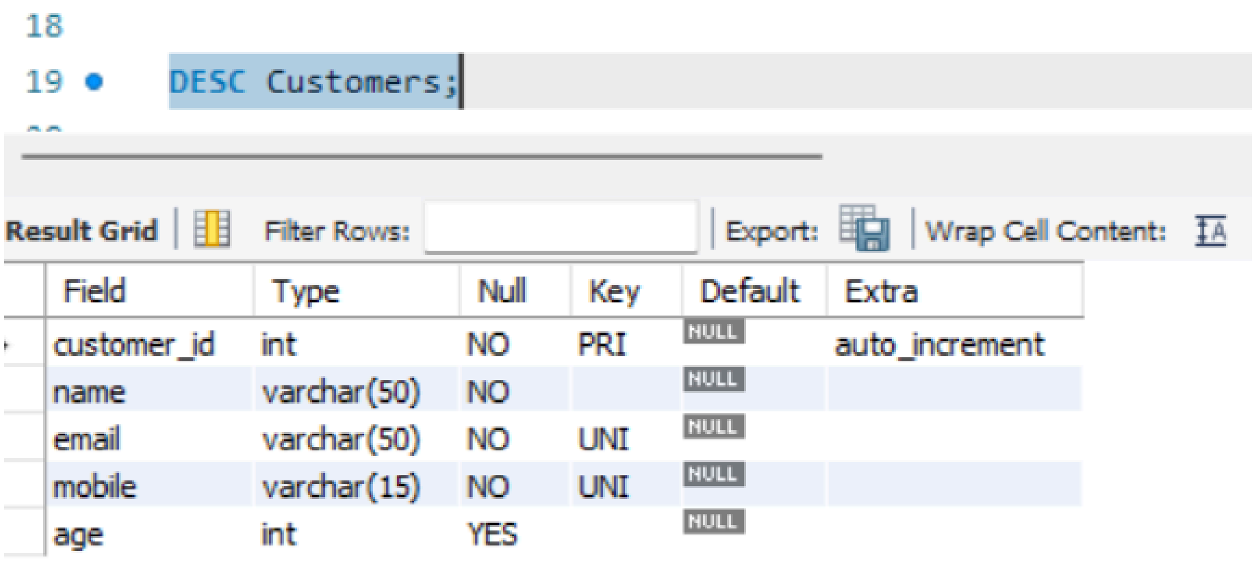
```
Output
Action Output
#  Time      Action                                                                        Message
1  23:04:41  create database e_commerce                                                    1 row(s) affected
2  23:04:45  use e_commerce                                                                0 row(s) affected
3  23:05:40  CREATE TABLE Customers (    customer_id INT AUTO_INCREMENT PRIMARY KE...      0 row(s) affected
4  23:05:40  CREATE TABLE Products (    id INT AUTO_INCREMENT PRIMARY KEY,    name V...    0 row(s) affected
```

3. Modify Tables(using Alter keyword):

a. Add not null on name and email in the Customers table
b. Add unique key on email in the Customers table
c. Add column age in the Customers table
d. Change column name from id to product_id in the Products table;
e. Add primary key and auto increment on product_id in the Products table
f. Change datatype of description from varchar to text in the Products table

ANS -

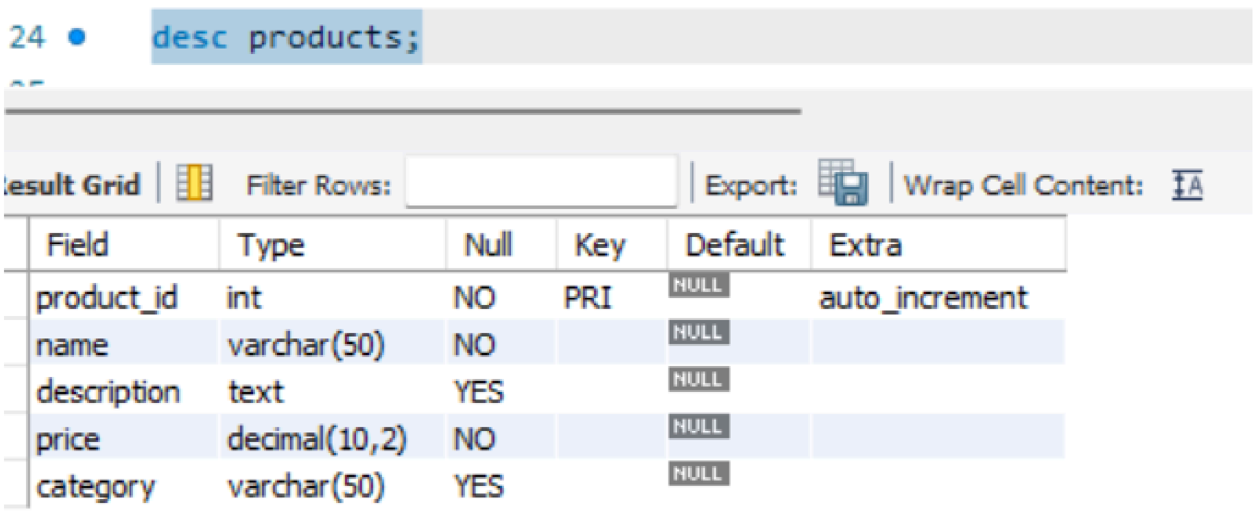
- ALTER TABLE Customers MODIFY name VARCHAR(50) NOT NULL, MODIFY email VARCHAR(50) NOT NULL UNIQUE, ADD COLUMN age INT;
- ALTER TABLE Products
- CHANGE COLUMN id product_id INT AUTO_INCREMENT PRIMARY KEY,
- MODIFY description TEXT;



```
DESC Customers;
```

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

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	auto_increment
name	varchar(50)	NO		NULL	
email	varchar(50)	NO	UNI	NULL	
mobile	varchar(15)	NO	UNI	NULL	
age	int	YES		NULL	

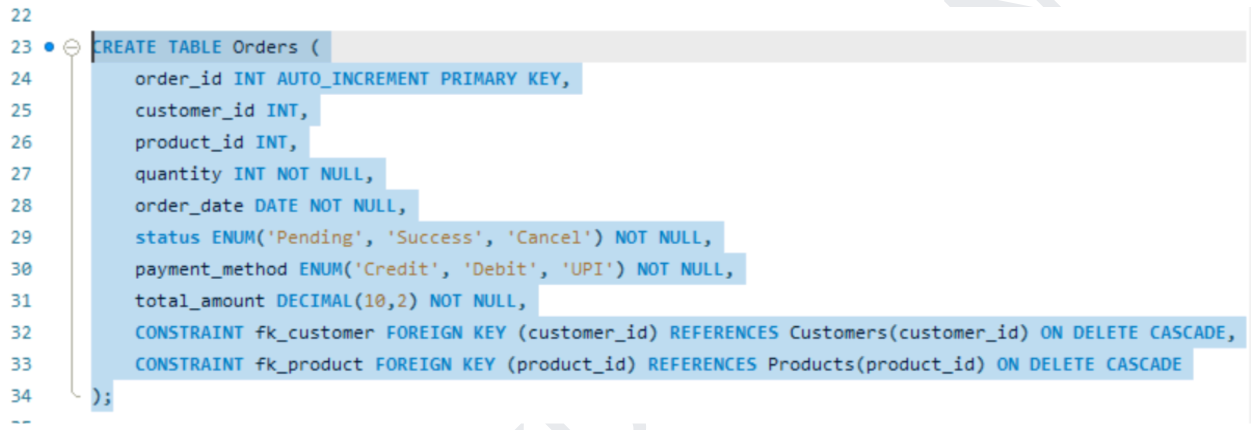


```
desc products;
```

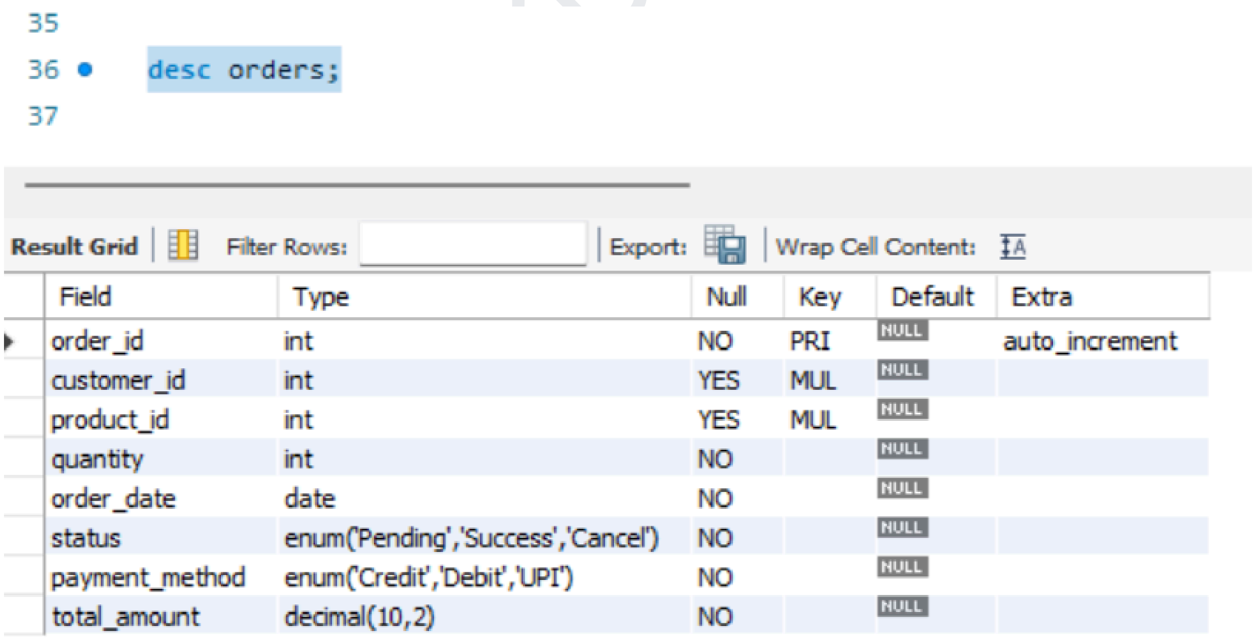
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
product_id	int	NO	PRI	NULL	auto_increment
name	varchar(50)	NO		NULL	
description	text	YES		NULL	
price	decimal(10,2)	NO		NULL	
category	varchar(50)	YES		NULL	

4. **Create table Order:**
    a. order_id - int auto-increment primary key
    b. customer_id - int -foreign key
    c. product_id - int
    d. quantity - int not null,
    e. order_date - date not null,
    f. status - enum(Pending, Success, Cancel),
    g. payment_method - enum(Credit, Debit, UPI),
    h. total_amount - decimal(10, 2) not null



```sql
CREATE TABLE Orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    quantity INT NOT NULL,
    order_date DATE NOT NULL,
    status ENUM('Pending', 'Success', 'Cancel') NOT NULL,
    payment_method ENUM('Credit', 'Debit', 'UPI') NOT NULL,
    total_amount DECIMAL(10,2) NOT NULL,
    CONSTRAINT fk_customer FOREIGN KEY (customer_id) REFERENCES Customers(customer_id) ON DELETE CASCADE,
    CONSTRAINT fk_product FOREIGN KEY (product_id) REFERENCES Products(product_id) ON DELETE CASCADE
);
```



```sql
desc orders;
```

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

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	auto_increment
customer_id	int	YES	MUL	NULL	
product_id	int	YES	MUL	NULL	
quantity	int	NO		NULL	
order_date	date	NO		NULL	
status	enum('Pending','Success','Cancel')	NO		NULL	
payment_method	enum('Credit','Debit','UPI')	NO		NULL	
total_amount	decimal(10,2)	NO		NULL	

5. Modify Orders Table(using Alter keyword):

a. Change table name Order -> Orders

```
ALTER TABLE `Order` RENAME TO Orders;
```

b. Set default value pending in status.

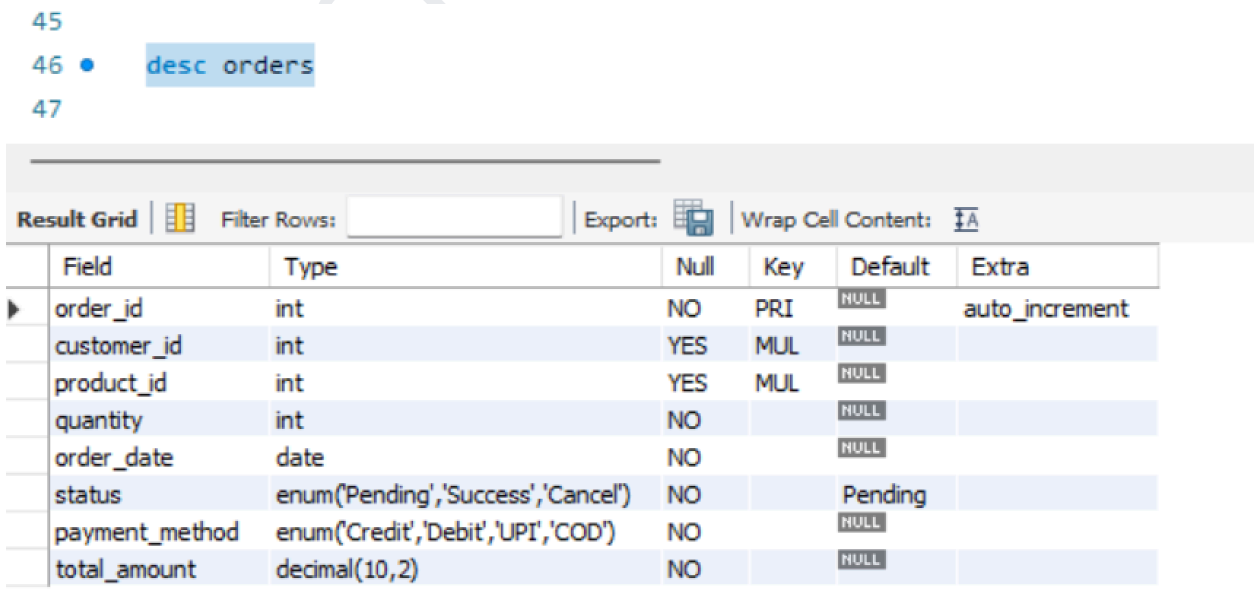
```
ALTER TABLE Orders
MODIFY status ENUM('Pending', 'Success', 'Cancel') NOT NULL
DEFAULT 'Pending';
```

c. Modify payment_method ENUM to add one more value: 'COD'

```
ALTER TABLE Orders
MODIFY payment_method ENUM('Credit', 'Debit', 'UPI', 'COD') NOT
NULL;
```

d. Make product id as foreign key

```
ALTER TABLE Orders
ADD CONSTRAINT fk_product FOREIGN KEY (product_id)
REFERENCES Products(product_id) ON DELETE CASCADE;
```

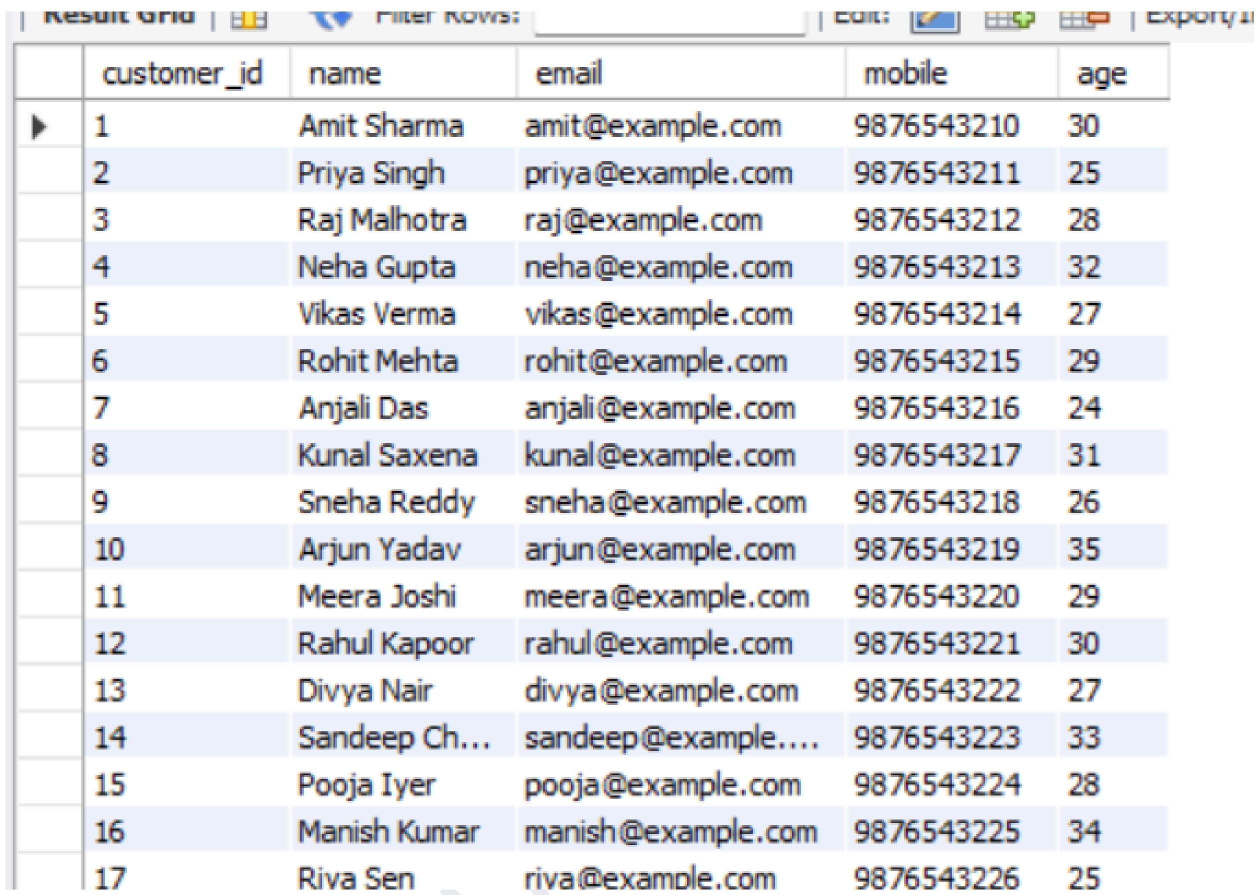


```
desc orders
```

Field	Type	Null	Key	Default	Extra
order_id	int	NO	PRI	NULL	auto_increment
customer_id	int	YES	MUL	NULL	
product_id	int	YES	MUL	NULL	
quantity	int	NO		NULL	
order_date	date	NO		NULL	
status	enum('Pending','Success','Cancel')	NO		Pending	
payment_method	enum('Credit','Debit','UPI','COD')	NO		NULL	
total_amount	decimal(10,2)	NO		NULL	

6. **Insert 20 sample records in all the tables.**

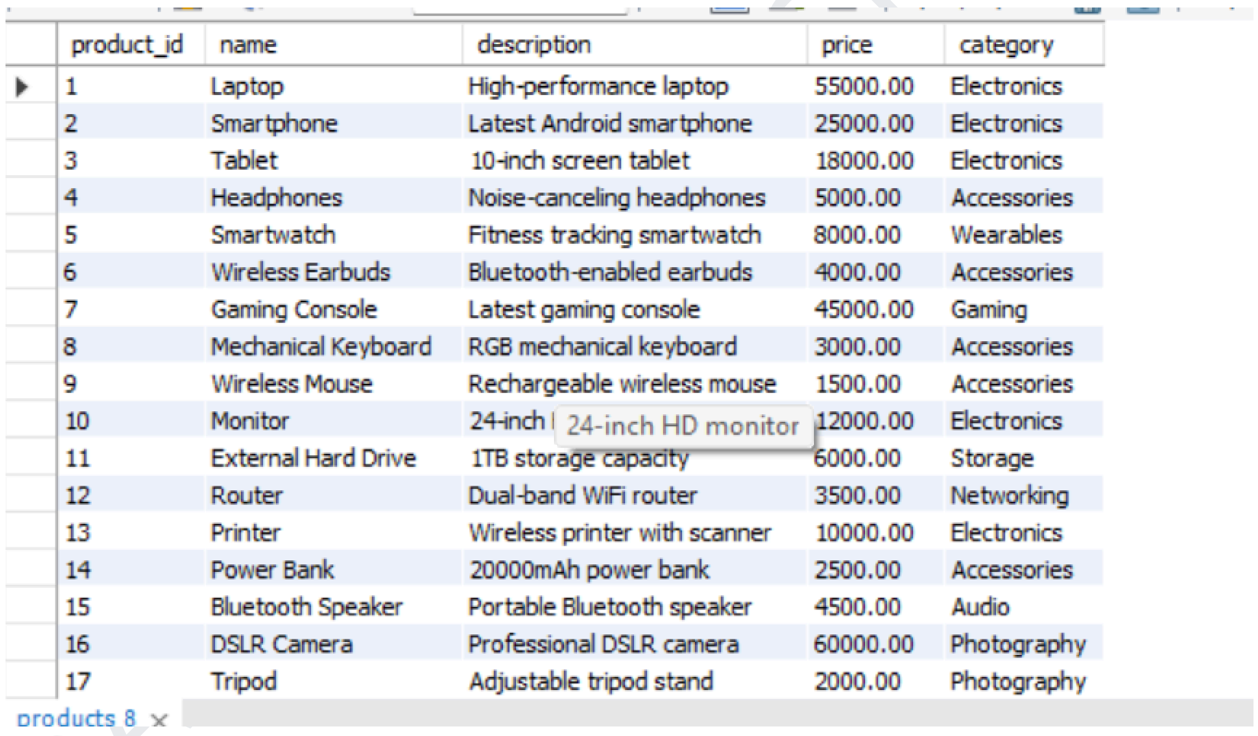
```
INSERT INTO Customers (name, email, mobile, age) VALUES
('Amit Sharma', 'amit@example.com', '9876543210', 30),
('Priya Singh', 'priya@example.com', '9876543211', 25),
('Raj Malhotra', 'raj@example.com', '9876543212', 28),
('Neha Gupta', 'neha@example.com', '9876543213', 32),
('Vikas Verma', 'vikas@example.com', '9876543214', 27),
('Rohit Mehta', 'rohit@example.com', '9876543215', 29),
('Anjali Das', 'anjali@example.com', '9876543216', 24),
('Kunal Saxena', 'kunal@example.com', '9876543217', 31),
('Sneha Reddy', 'sneha@example.com', '9876543218', 26),
('Arjun Yadav', 'arjun@example.com', '9876543219', 35),
('Meera Joshi', 'meera@example.com', '9876543220', 29),
('Rahul Kapoor', 'rahul@example.com', '9876543221', 30),
('Divya Nair', 'divya@example.com', '9876543222', 27),
('Sandeep Chauhan', 'sandeep@example.com', '9876543223', 33),
('Pooja Iyer', 'pooja@example.com', '9876543224', 28),
('Manish Kumar', 'manish@example.com', '9876543225', 34),
('Riya Sen', 'riya@example.com', '9876543226', 25),
('Aditya Rao', 'aditya@example.com', '9876543227', 29),
('Sonal Bajaj', 'sonal@example.com', '9876543228', 26),
('Harsh Aggarwal', 'harsh@example.com', '9876543229', 31);
```



	customer_id	name	email	mobile	age
▶	1	Amit Sharma	amit@example.com	9876543210	30
	2	Priya Singh	priya@example.com	9876543211	25
	3	Raj Malhotra	raj@example.com	9876543212	28
	4	Neha Gupta	neha@example.com	9876543213	32
	5	Vikas Verma	vikas@example.com	9876543214	27
	6	Rohit Mehta	rohit@example.com	9876543215	29
	7	Anjali Das	anjali@example.com	9876543216	24
	8	Kunal Saxena	kunal@example.com	9876543217	31
	9	Sneha Reddy	sneha@example.com	9876543218	26
	10	Arjun Yadav	arjun@example.com	9876543219	35
	11	Meera Joshi	meera@example.com	9876543220	29
	12	Rahul Kapoor	rahul@example.com	9876543221	30
	13	Divya Nair	divya@example.com	9876543222	27
	14	Sandeep Ch...	sandeep@example....	9876543223	33
	15	Pooja Iyer	pooja@example.com	9876543224	28
	16	Manish Kumar	manish@example.com	9876543225	34
	17	Riya Sen	riya@example.com	9876543226	25

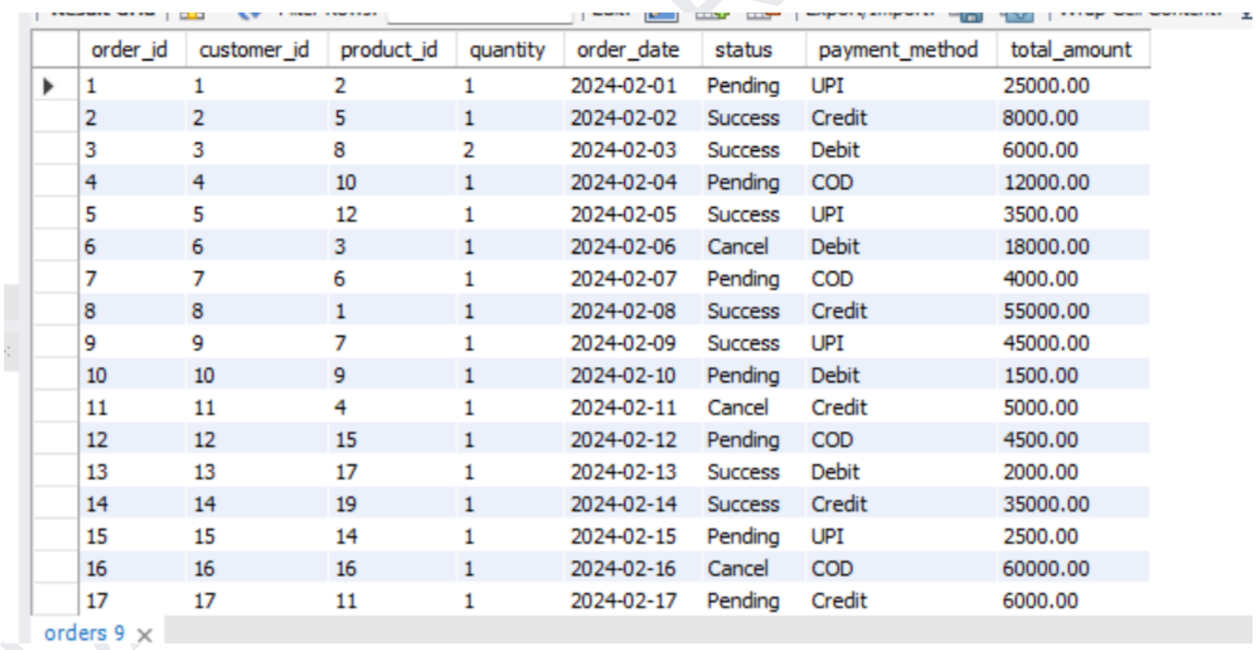
INSERT INTO Products (name, description, price, category) VALUES
('Laptop', 'High-performance laptop', 55000.00, 'Electronics'),
('Smartphone', 'Latest Android smartphone', 25000.00, 'Electronics'),
('Tablet', '10-inch screen tablet', 18000.00, 'Electronics'),
('Headphones', 'Noise-canceling headphones', 5000.00,
'Accessories'),
('Smartwatch', 'Fitness tracking smartwatch', 8000.00, 'Wearables'),
('Wireless Earbuds', 'Bluetooth-enabled earbuds', 4000.00,
'Accessories'),
('Gaming Console', 'Latest gaming console', 45000.00, 'Gaming'),
('Mechanical Keyboard', 'RGB mechanical keyboard', 3000.00,
'Accessories'),
('Wireless Mouse', 'Rechargeable wireless mouse', 1500.00,
'Accessories'),
('Monitor', '24-inch HD monitor', 12000.00, 'Electronics'),
('External Hard Drive', '1TB storage capacity', 6000.00, 'Storage'),
('Router', 'Dual-band WiFi router', 3500.00, 'Networking'),

```
('Printer', 'Wireless printer with scanner', 10000.00, 'Electronics'),
('Power Bank', '20000mAh power bank', 2500.00, 'Accessories'),
('Bluetooth Speaker', 'Portable Bluetooth speaker', 4500.00, 'Audio'),
('DSLR Camera', 'Professional DSLR camera', 60000.00,
'Photography'),
('Tripod', 'Adjustable tripod stand', 2000.00, 'Photography'),
('Microwave Oven', '30L convection oven', 15000.00, 'Home
Appliances'),
('Air Conditioner', '1.5 Ton Inverter AC', 35000.00, 'Home Appliances'),
('Refrigerator', '300L double-door fridge', 28000.00, 'Home
Appliances');
```



product_id	name	description	price	category
1	Laptop	High-performance laptop	55000.00	Electronics
2	Smartphone	Latest Android smartphone	25000.00	Electronics
3	Tablet	10-inch screen tablet	18000.00	Electronics
4	Headphones	Noise-canceling headphones	5000.00	Accessories
5	Smartwatch	Fitness tracking smartwatch	8000.00	Wearables
6	Wireless Earbuds	Bluetooth-enabled earbuds	4000.00	Accessories
7	Gaming Console	Latest gaming console	45000.00	Gaming
8	Mechanical Keyboard	RGB mechanical keyboard	3000.00	Accessories
9	Wireless Mouse	Rechargeable wireless mouse	1500.00	Accessories
10	Monitor	24-inch HD monitor	12000.00	Electronics
11	External Hard Drive	1TB storage capacity	6000.00	Storage
12	Router	Dual-band WiFi router	3500.00	Networking
13	Printer	Wireless printer with scanner	10000.00	Electronics
14	Power Bank	20000mAh power bank	2500.00	Accessories
15	Bluetooth Speaker	Portable Bluetooth speaker	4500.00	Audio
16	DSLR Camera	Professional DSLR camera	60000.00	Photography
17	Tripod	Adjustable tripod stand	2000.00	Photography

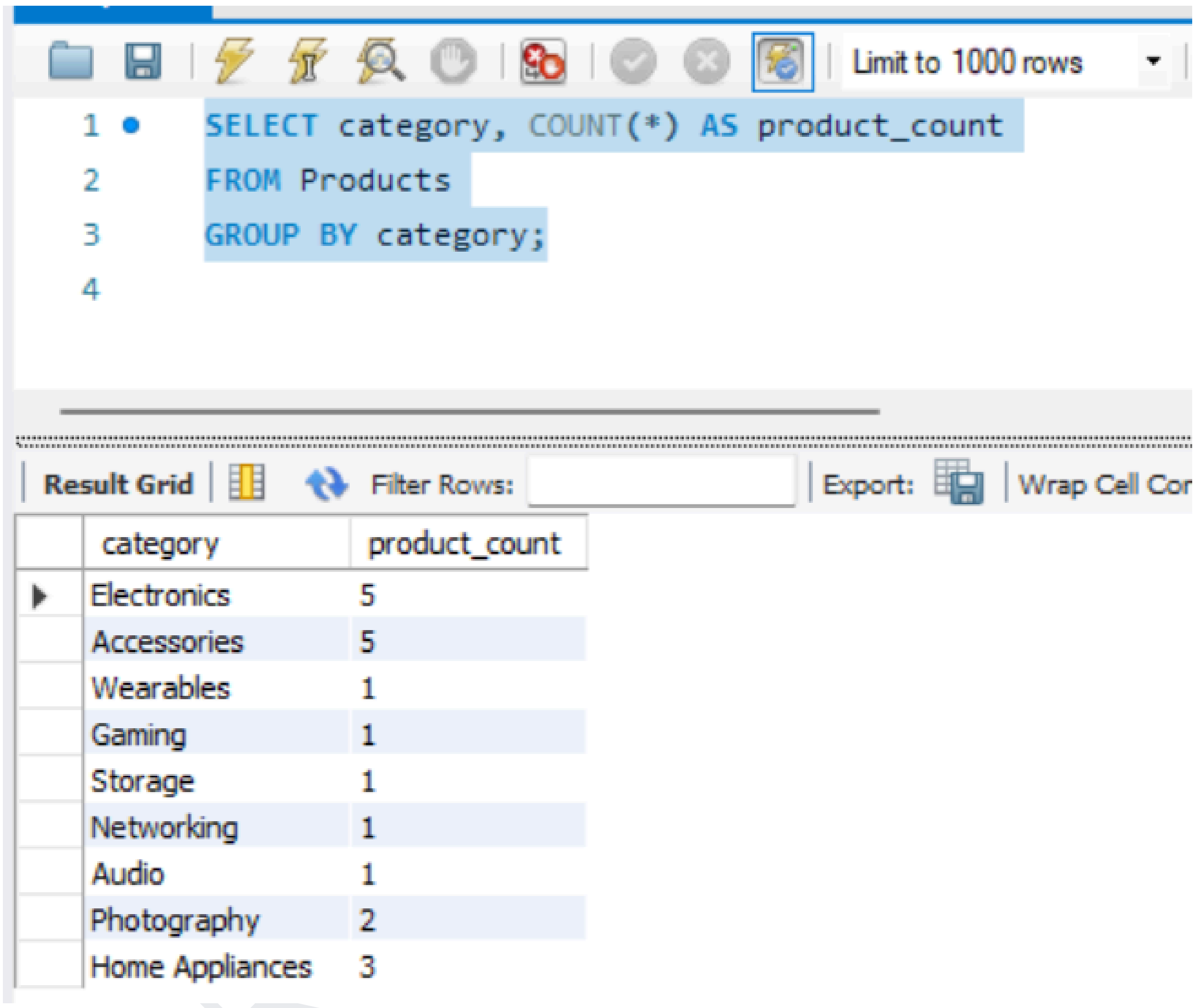
```
INSERT INTO Orders (customer_id, product_id, quantity, order_date,
status, payment_method, total_amount) VALUES
(1, 2, 1, '2024-02-01', 'Pending', 'UPI', 25000.00),
(2, 5, 1, '2024-02-02', 'Success', 'Credit', 8000.00),
(3, 8, 2, '2024-02-03', 'Success', 'Debit', 6000.00),
(4, 10, 1, '2024-02-04', 'Pending', 'COD', 12000.00),
(5, 12, 1, '2024-02-05', 'Success', 'UPI', 3500.00),
(6, 3, 1, '2024-02-06', 'Cancel', 'Debit', 18000.00),
(7, 6, 1, '2024-02-07', 'Pending', 'COD', 4000.00),
(8, 1, 1, '2024-02-08', 'Success', 'Credit', 55000.00),
(9, 7, 1, '2024-02-09', 'Success', 'UPI', 45000.00),
(10, 9, 1, '2024-02-10', 'Pending', 'Debit', 1500.00),
(11, 4, 1, '2024-02-11', 'Cancel', 'Credit', 5000.00),
(12, 15, 1, '2024-02-12', 'Pending', 'COD', 4500.00),
(13, 17, 1, '2024-02-13', 'Success', 'Debit', 2000.00),
(14, 19, 1, '2024-02-14', 'Success', 'Credit', 35000.00),
(15, 14, 1, '2024-02-15', 'Pending', 'UPI', 2500.00),
(16, 16, 1, '2024-02-16', 'Cancel', 'COD', 60000.00),
(17, 11, 1, '2024-02-17', 'Pending', 'Credit', 6000.00),
(18, 13, 1, '2024-02-18', 'Success', 'UPI', 10000.00),
(19, 20, 1, '2024-02-19', 'Pending', 'COD', 28000.00),
(20, 18, 1, '2024-02-20', 'Success', 'Debit', 15000.00);
```



order_id	customer_id	product_id	quantity	order_date	status	payment_method	total_amount
1	1	2	1	2024-02-01	Pending	UPI	25000.00
2	2	5	1	2024-02-02	Success	Credit	8000.00
3	3	8	2	2024-02-03	Success	Debit	6000.00
4	4	10	1	2024-02-04	Pending	COD	12000.00
5	5	12	1	2024-02-05	Success	UPI	3500.00
6	6	3	1	2024-02-06	Cancel	Debit	18000.00
7	7	6	1	2024-02-07	Pending	COD	4000.00
8	8	1	1	2024-02-08	Success	Credit	55000.00
9	9	7	1	2024-02-09	Success	UPI	45000.00
10	10	9	1	2024-02-10	Pending	Debit	1500.00
11	11	4	1	2024-02-11	Cancel	Credit	5000.00
12	12	15	1	2024-02-12	Pending	COD	4500.00
13	13	17	1	2024-02-13	Success	Debit	2000.00
14	14	19	1	2024-02-14	Success	Credit	35000.00
15	15	14	1	2024-02-15	Pending	UPI	2500.00
16	16	16	1	2024-02-16	Cancel	COD	60000.00
17	17	11	1	2024-02-17	Pending	Credit	6000.00

7. **Perform following queries:**

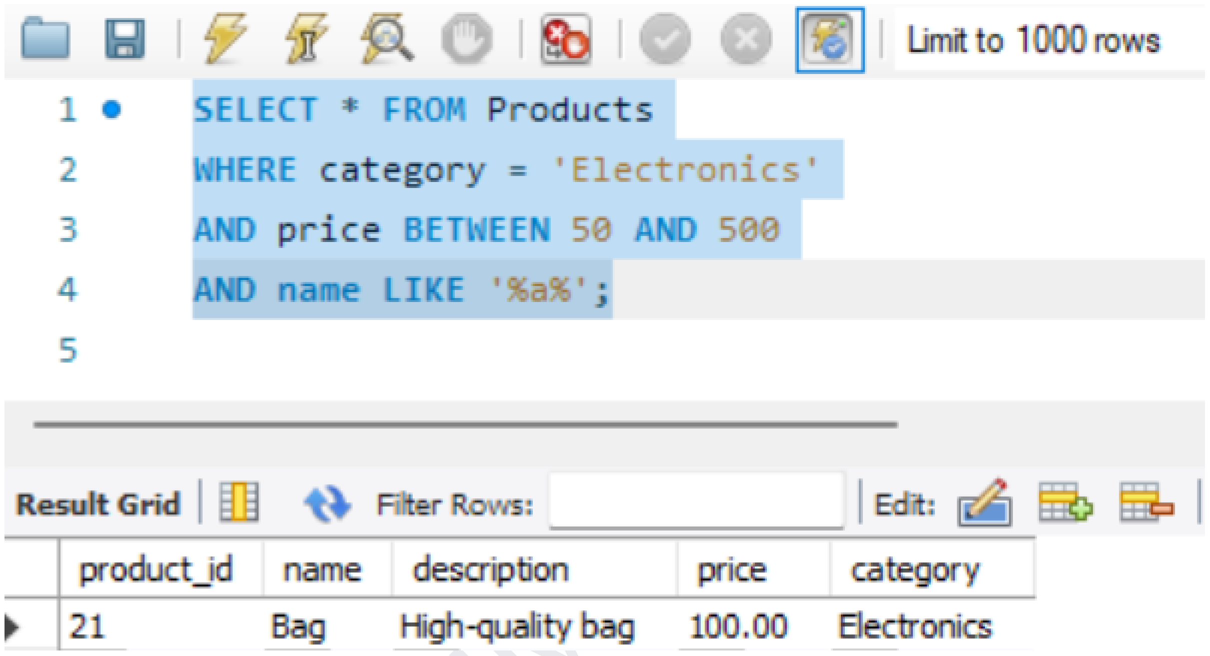
   a. Count the number of products as product_count in each category.



```sql
SELECT category, COUNT(*) AS product_count
FROM Products
GROUP BY category;
```

category	product_count
Electronics	5
Accessories	5
Wearables	1
Gaming	1
Storage	1
Networking	1
Audio	1
Photography	2
Home Appliances	3

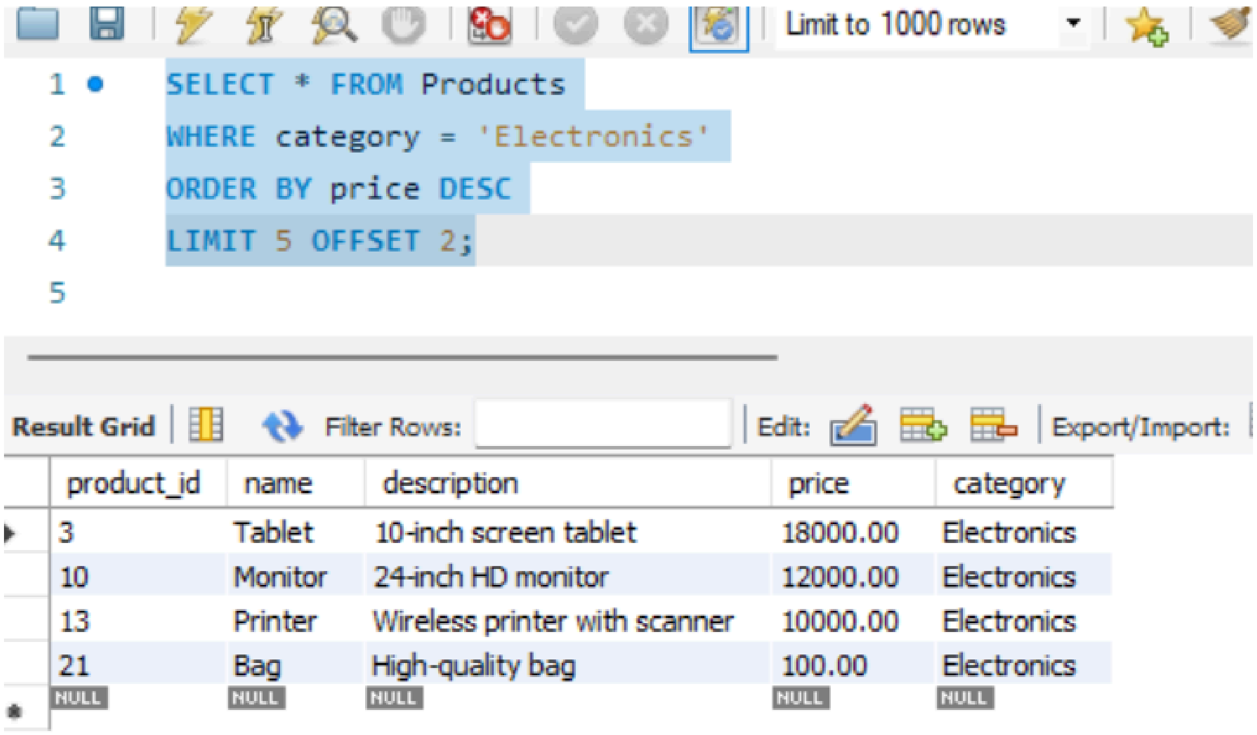
b. Retrieve all products that belong to the 'Electronics' category, have a price between $50 and $500, and whose name contains the letter 'a'.



```sql
SELECT * FROM Products
WHERE category = 'Electronics'
AND price BETWEEN 50 AND 500
AND name LIKE '%a%';
```

product_id	name	description	price	category
21	Bag	High-quality bag	100.00	Electronics

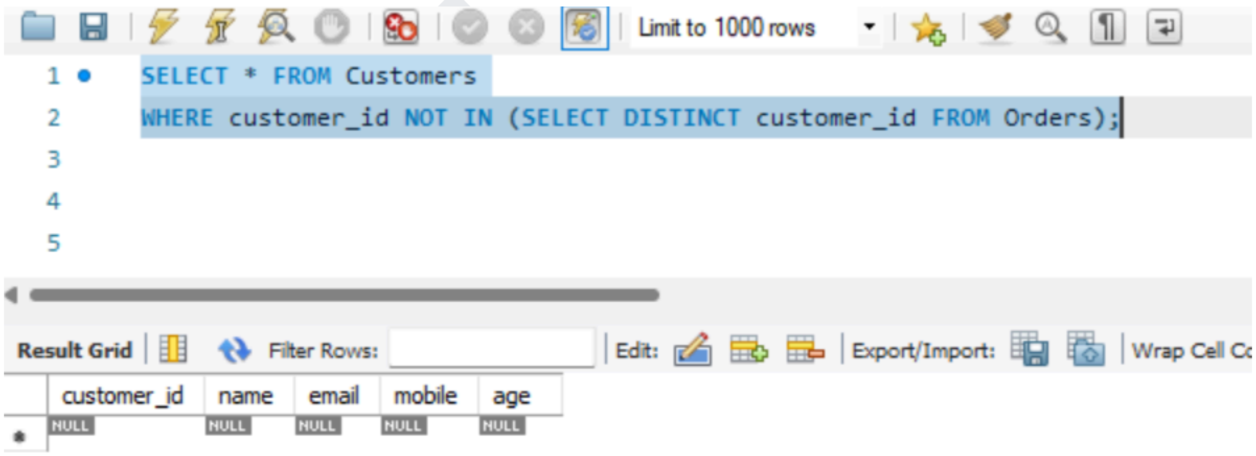
c. Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2.



```
SELECT * FROM Products
WHERE category = 'Electronics'
ORDER BY price DESC
LIMIT 5 OFFSET 2;
```

product_id	name	description	price	category
3	Tablet	10-inch screen tablet	18000.00	Electronics
10	Monitor	24-inch HD monitor	12000.00	Electronics
13	Printer	Wireless printer with scanner	10000.00	Electronics
21	Bag	High-quality bag	100.00	Electronics
NULL	NULL	NULL	NULL	NULL

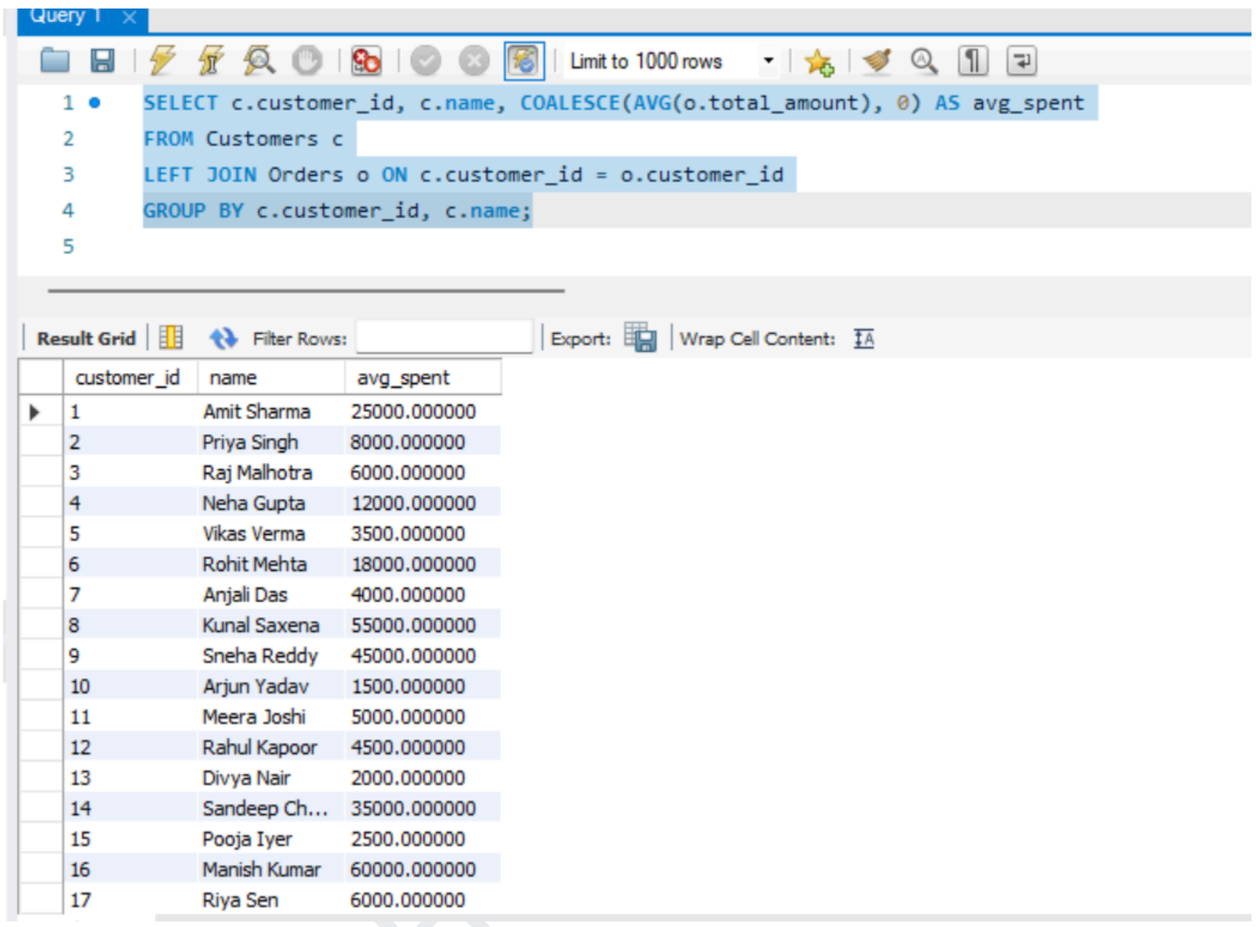
d. Retrieve customers who have not placed any orders.



```
SELECT * FROM Customers
WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM Orders);
```

customer_id	name	email	mobile	age
NULL	NULL	NULL	NULL	NULL

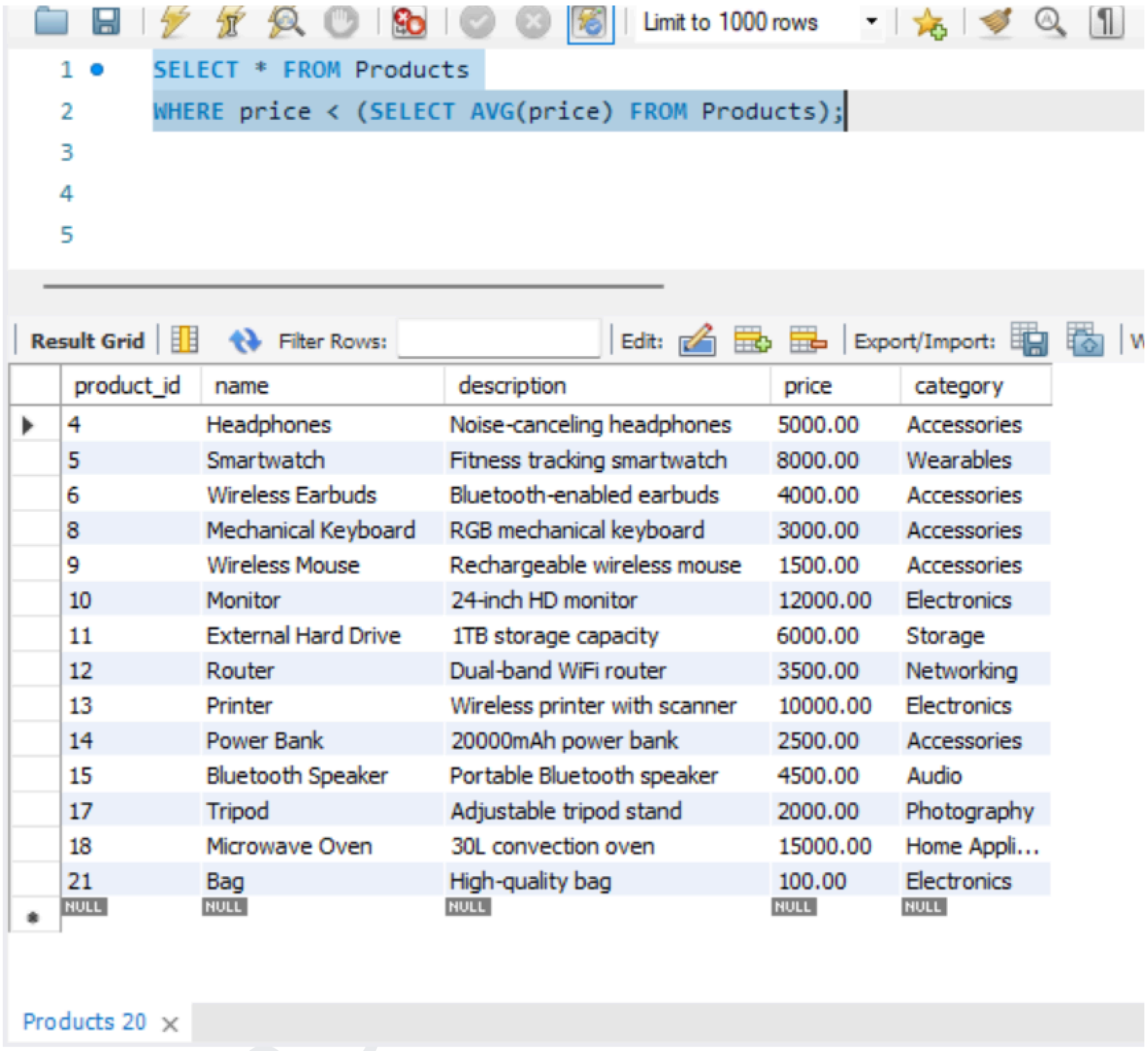
e. Find the average total amount spent by each customer.



```sql
SELECT c.customer_id, c.name, COALESCE(AVG(o.total_amount), 0) AS avg_spent
FROM Customers c
LEFT JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name;
```

customer_id	name	avg_spent
1	Amit Sharma	25000.000000
2	Priya Singh	8000.000000
3	Raj Malhotra	6000.000000
4	Neha Gupta	12000.000000
5	Vikas Verma	3500.000000
6	Rohit Mehta	18000.000000
7	Anjali Das	4000.000000
8	Kunal Saxena	55000.000000
9	Sneha Reddy	45000.000000
10	Arjun Yadav	1500.000000
11	Meera Joshi	5000.000000
12	Rahul Kapoor	4500.000000
13	Divya Nair	2000.000000
14	Sandeep Ch...	35000.000000
15	Pooja Iyer	2500.000000
16	Manish Kumar	60000.000000
17	Riya Sen	6000.000000

f. Get the products that have a price less than the average price of all products.



Limit to 1000 rows

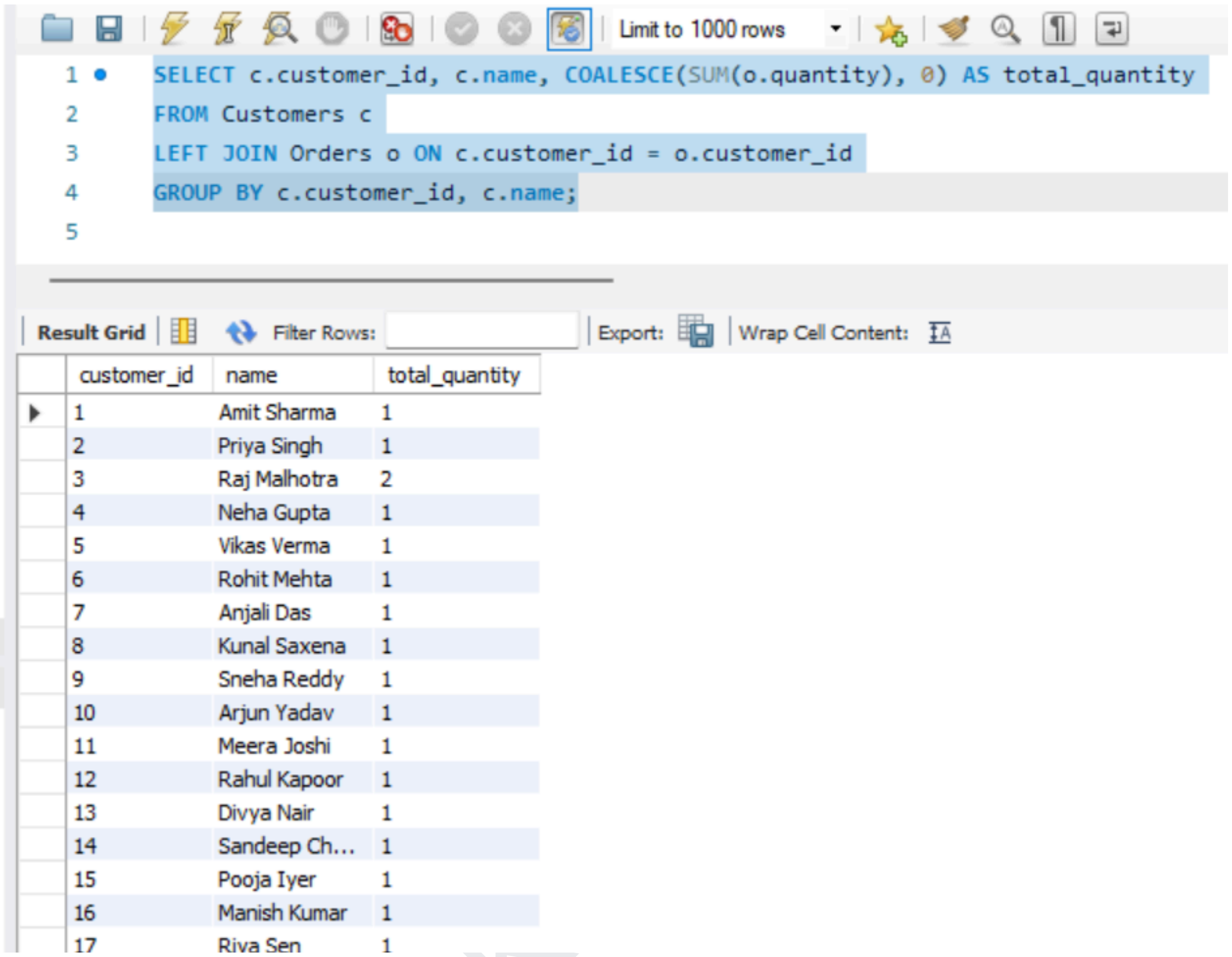
```
SELECT * FROM Products
WHERE price < (SELECT AVG(price) FROM Products);
```

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

product_id	name	description	price	category
4	Headphones	Noise-canceling headphones	5000.00	Accessories
5	Smartwatch	Fitness tracking smartwatch	8000.00	Wearables
6	Wireless Earbuds	Bluetooth-enabled earbuds	4000.00	Accessories
8	Mechanical Keyboard	RGB mechanical keyboard	3000.00	Accessories
9	Wireless Mouse	Rechargeable wireless mouse	1500.00	Accessories
10	Monitor	24-inch HD monitor	12000.00	Electronics
11	External Hard Drive	1TB storage capacity	6000.00	Storage
12	Router	Dual-band WiFi router	3500.00	Networking
13	Printer	Wireless printer with scanner	10000.00	Electronics
14	Power Bank	20000mAh power bank	2500.00	Accessories
15	Bluetooth Speaker	Portable Bluetooth speaker	4500.00	Audio
17	Tripod	Adjustable tripod stand	2000.00	Photography
18	Microwave Oven	30L convection oven	15000.00	Home Appli...
21	Bag	High-quality bag	100.00	Electronics
NULL	NULL	NULL	NULL	NULL

Products 20

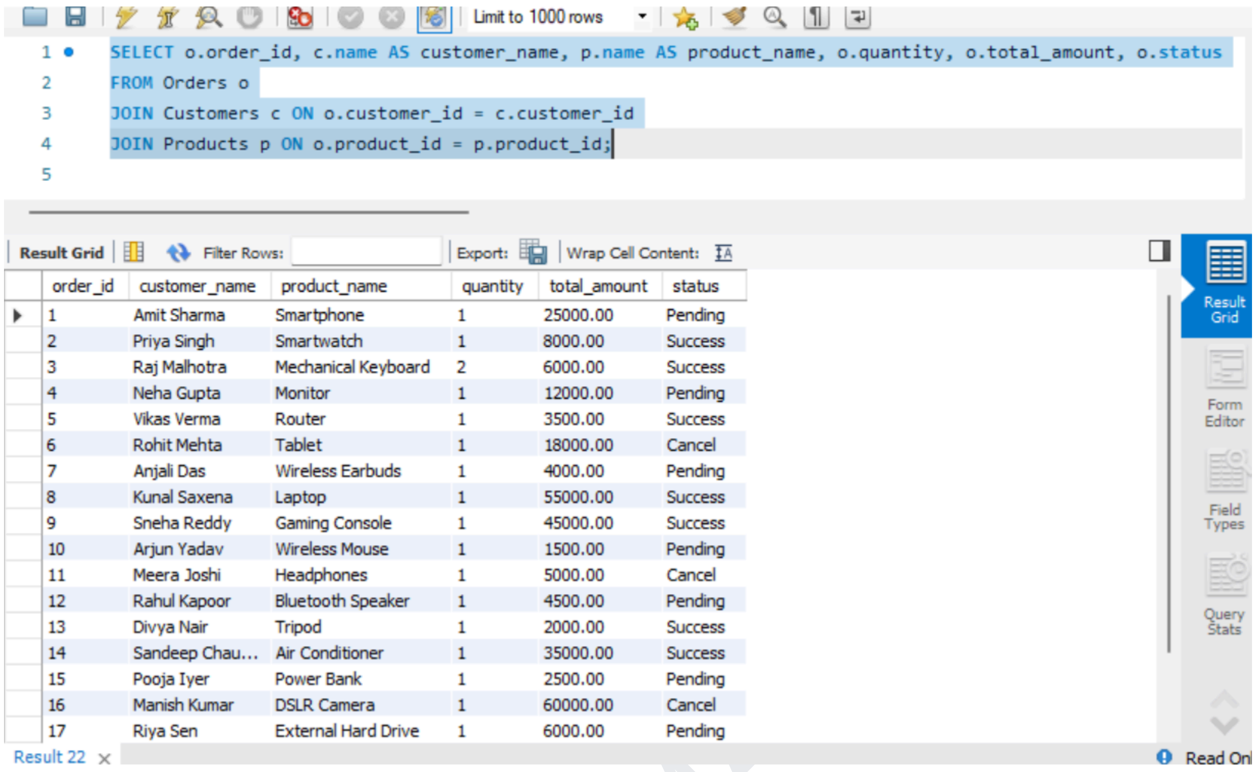
g. Calculate the total quantity of products ordered by each customer:



```sql
SELECT c.customer_id, c.name, COALESCE(SUM(o.quantity), 0) AS total_quantity
FROM Customers c
LEFT JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name;
```

customer_id	name	total_quantity
1	Amit Sharma	1
2	Priya Singh	1
3	Raj Malhotra	2
4	Neha Gupta	1
5	Vikas Verma	1
6	Rohit Mehta	1
7	Anjali Das	1
8	Kunal Saxena	1
9	Sneha Reddy	1
10	Arjun Yadav	1
11	Meera Joshi	1
12	Rahul Kapoor	1
13	Divya Nair	1
14	Sandeep Ch...	1
15	Pooja Iyer	1
16	Manish Kumar	1
17	Riya Sen	1

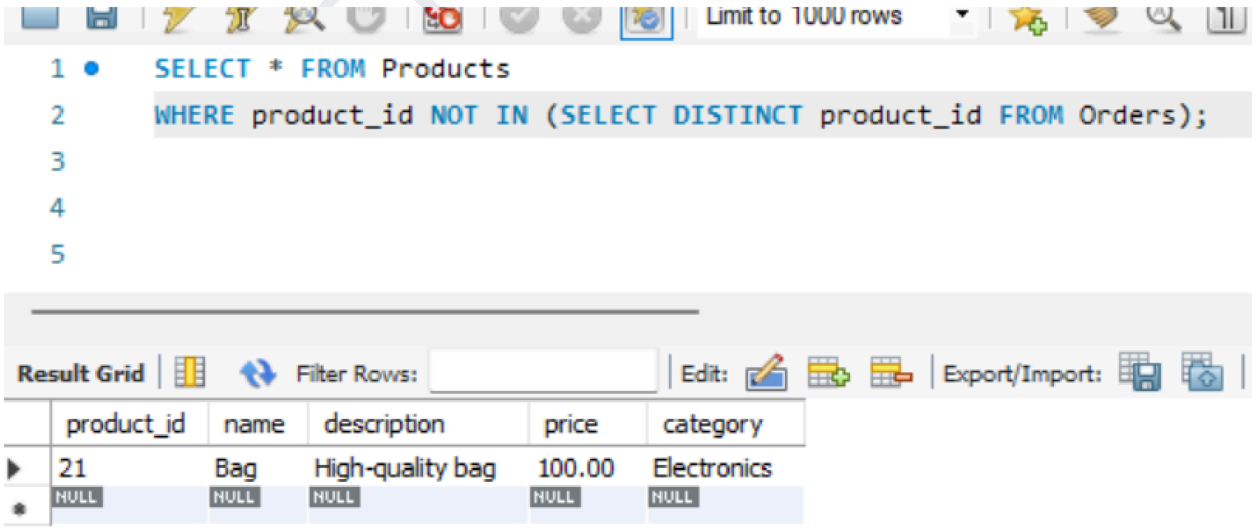
h. List all orders along with customer name and product name.



```sql
SELECT o.order_id, c.name AS customer_name, p.name AS product_name, o.quantity, o.total_amount, o.status
FROM Orders o
JOIN Customers c ON o.customer_id = c.customer_id
JOIN Products p ON o.product_id = p.product_id;
```

order_id	customer_name	product_name	quantity	total_amount	status
1	Amit Sharma	Smartphone	1	25000.00	Pending
2	Priya Singh	Smartwatch	1	8000.00	Success
3	Raj Malhotra	Mechanical Keyboard	2	6000.00	Success
4	Neha Gupta	Monitor	1	12000.00	Pending
5	Vikas Verma	Router	1	3500.00	Success
6	Rohit Mehta	Tablet	1	18000.00	Cancel
7	Anjali Das	Wireless Earbuds	1	4000.00	Pending
8	Kunal Saxena	Laptop	1	55000.00	Success
9	Sneha Reddy	Gaming Console	1	45000.00	Success
10	Arjun Yadav	Wireless Mouse	1	1500.00	Pending
11	Meera Joshi	Headphones	1	5000.00	Cancel
12	Rahul Kapoor	Bluetooth Speaker	1	4500.00	Pending
13	Divya Nair	Tripod	1	2000.00	Success
14	Sandeep Chau...	Air Conditioner	1	35000.00	Success
15	Pooja Iyer	Power Bank	1	2500.00	Pending
16	Manish Kumar	DSLR Camera	1	60000.00	Cancel
17	Riya Sen	External Hard Drive	1	6000.00	Pending

i. Find products that have never been ordered.



```sql
SELECT * FROM Products
WHERE product_id NOT IN (SELECT DISTINCT product_id FROM Orders);
```

product_id	name	description	price	category
21	Bag	High-quality bag	100.00	Electronics
NULL	NULL	NULL	NULL	NULL