Perform following queries:

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

Run SQLite

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

category	product_count
Accessories	1
Appliances	4
Electronics	8
Fashion	1
Furniture	2
Health	1
Home	1
Outdoor	1
Sports	1

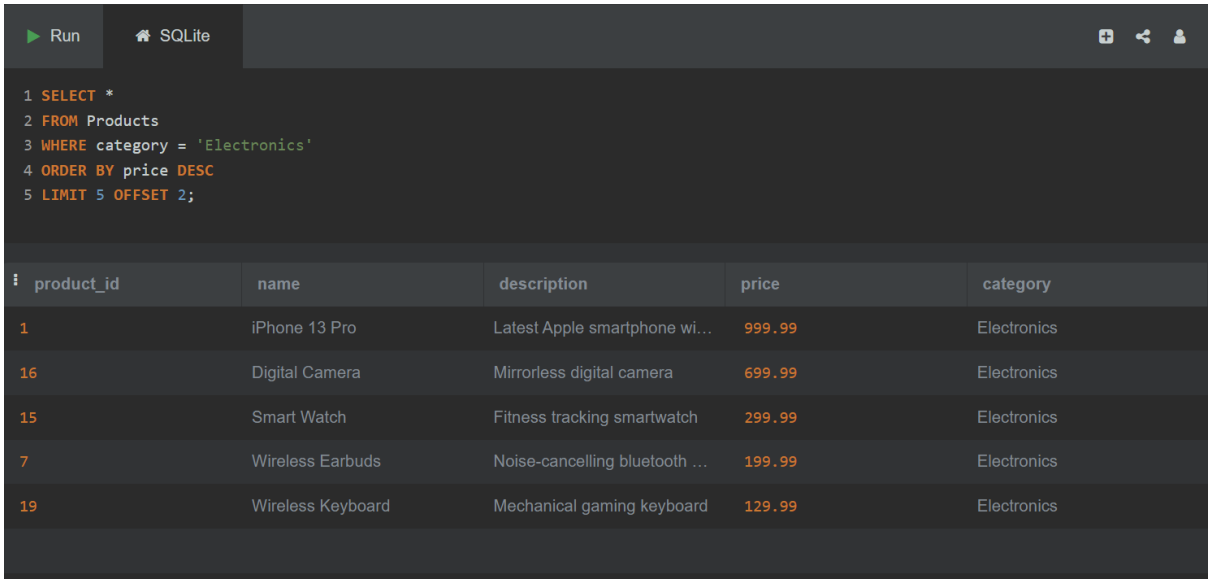
b. Retrieve all products that belong to the 'Electronics' category, have a price between $50 and $500, and whose name contains the letter 'a'

Run SQLite

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

product_id	name	description	price	category
4	Gaming Mouse	RGB wireless gaming mouse	79.99	Electronics
7	Wireless Earbuds	Noise-cancelling bluetooth …	199.99	Electronics
15	Smart Watch	Fitness tracking smartwatch	299.99	Electronics
19	Wireless Keyboard	Mechanical gaming keyboard	129.99	Electronics

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

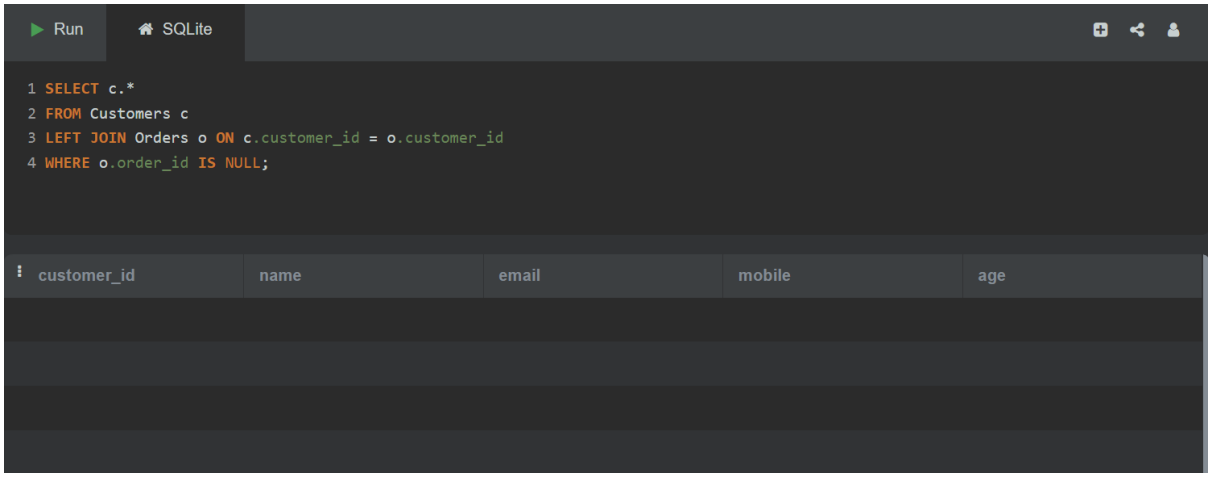


Run SQLite

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

product_id	name	description	price	category
1	iPhone 13 Pro	Latest Apple smartphone wi…	999.99	Electronics
16	Digital Camera	Mirrorless digital camera	699.99	Electronics
15	Smart Watch	Fitness tracking smartwatch	299.99	Electronics
7	Wireless Earbuds	Noise-cancelling bluetooth …	199.99	Electronics
19	Wireless Keyboard	Mechanical gaming keyboard	129.99	Electronics

d. Retrieve customers who have not placed any orders.

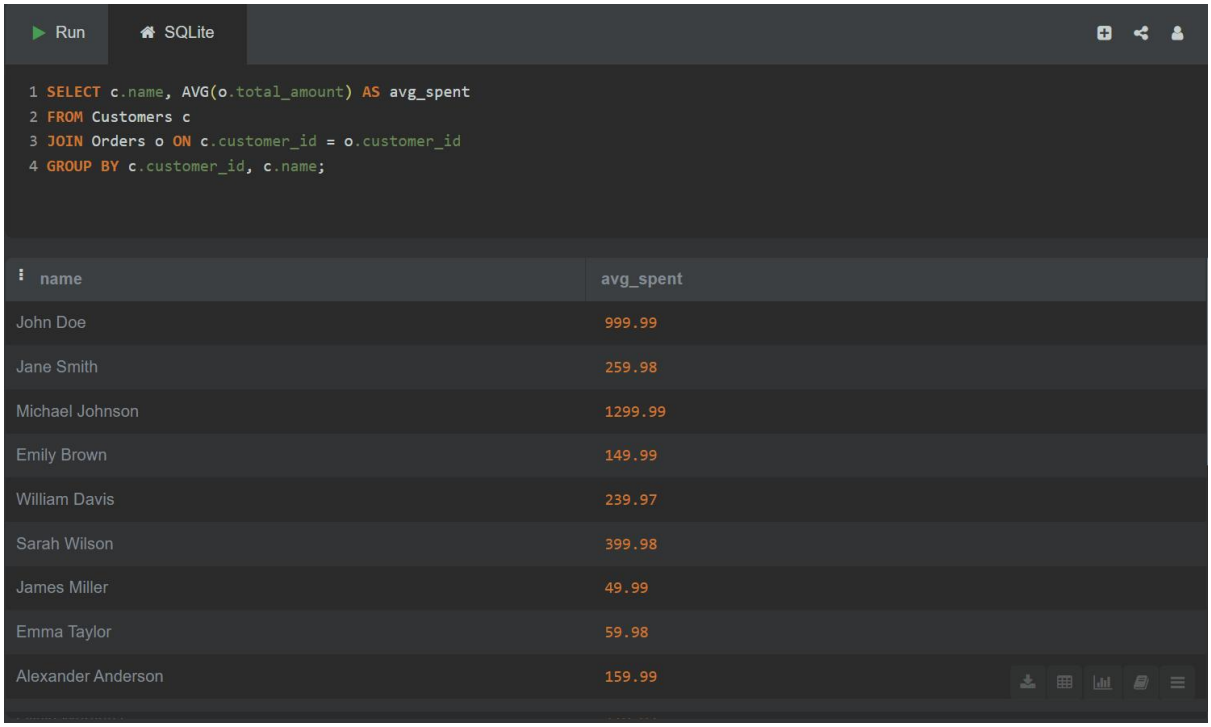


Run SQLite

```
SELECT c.*
FROM Customers c
LEFT JOIN Orders o ON c.customer_id = o.customer_id
WHERE o.order_id IS NULL;
```

customer_id	name	email	mobile	age

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

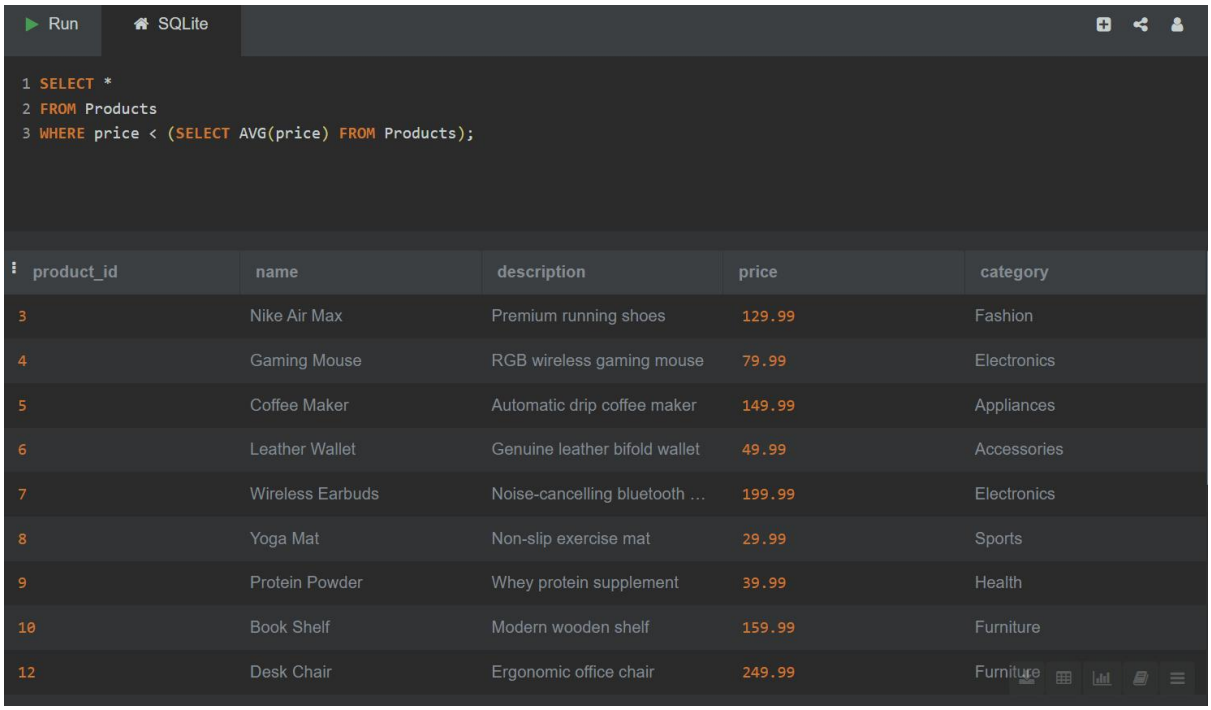


```
Run  SQLite

1 SELECT c.name, AVG(o.total_amount) AS avg_spent
2 FROM Customers c
3 JOIN Orders o ON c.customer_id = o.customer_id
4 GROUP BY c.customer_id, c.name;
```

name	avg_spent
John Doe	999.99
Jane Smith	259.98
Michael Johnson	1299.99
Emily Brown	149.99
William Davis	239.97
Sarah Wilson	399.98
James Miller	49.99
Emma Taylor	59.98
Alexander Anderson	159.99

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



```
Run  SQLite

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

product_id	name	description	price	category
3	Nike Air Max	Premium running shoes	129.99	Fashion
4	Gaming Mouse	RGB wireless gaming mouse	79.99	Electronics
5	Coffee Maker	Automatic drip coffee maker	149.99	Appliances
6	Leather Wallet	Genuine leather bifold wallet	49.99	Accessories
7	Wireless Earbuds	Noise-cancelling bluetooth ...	199.99	Electronics
8	Yoga Mat	Non-slip exercise mat	29.99	Sports
9	Protein Powder	Whey protein supplement	39.99	Health
10	Book Shelf	Modern wooden shelf	159.99	Furniture
12	Desk Chair	Ergonomic office chair	249.99	Furniture

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

Run SQLite

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

name	total_quantity
John Doe	1
Jane Smith	2
Michael Johnson	1
Emily Brown	1
William Davis	3
Sarah Wilson	2
James Miller	1
Emma Taylor	2
Alexander Anderson	1

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

Run SQLite

```
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	John Doe	iPhone 13 Pro	1	999.99	Success
2	Jane Smith	Nike Air Max	2	259.98	Pending
3	Michael Johnson	Samsung 4K TV	1	1299.99	Success
4	Emily Brown	Coffee Maker	1	149.99	Cancel
5	William Davis	Gaming Mouse	3	239.97	Success
6	Sarah Wilson	Wireless Earbuds	2	399.98	Success
7	James Miller	Leather Wallet	1	49.99	Pending
8	Emma Taylor	Yoga Mat	2	59.98	Success
9	Alexander Anderson	Book Shelf	1	159.99	Success

i. Find products that have never been ordered.