Database Name: ECommerceDB

Tables:

1. **Customers:**
- customer_id (Primary Key)
- name
- email
- address
2. **Orders:**
- order_id (Primary Key)
- customer_id (Foreign Key referencing Customers table)
- order_date
- total_amount
3. **Products:**
- product_id (Primary Key)
- name
- description
- price
4. **Order_Items:**
- order_item_id (Primary Key)
- order_id (Foreign Key referencing Orders table)
- product_id (Foreign Key referencing Products table)
- quantity
- unit_price

Customers:

customer_id	name	email	address

customer_id	name	email	address
4	Emily Johnson	emily@example.com	987 Oak Lane
5	Michael Brown	michael@example.com	321 Elm Avenue
6	Sarah Davis	sarah@example.com	654 Main Street
7	Jessica Lee	jessica@example.com	123 Maple Road
8	Brian Wilson	brian@example.com	456 Pine Street

Orders:

order_id	customer_id	order_date	total_amount

order_id	customer_id	order_date	total_amount
4	2	2023-06-05	120.00
5	4	2023-06-06	45.00
6	3	2023-06-07	80.00
7	5	2023-06-09	65.00
8	6	2023-06-10	95.00

Products:

product_id	name	description	price
4	Hoodie	Warm hoodie, available in	29.99

product_id	name	description	price
		multiple colors	
5	Dress	Elegant dress for special occasions	59.99
6	Watch	Stylish wristwatch with leather strap	99.99
7	Backpack	Spacious backpack for everyday use	49.99
8	Sunglasses	UV protection sunglasses	24.99

Order_Items:

order_item_id	order_id	product_id	quantity	unit_price

order_item_id	order_id	product_id	quantity	unit_price
5	4	2	2	39.99
6	4	3	1	49.99
7	5	4	1	29.99
8	6	6	1	99.99
9	6	8	2	24.99

1. Retrieve products with a price less than $50
2. Retrieve orders placed on or after June 5th, 2023:
3. Retrieve the customers' names and their corresponding order count:
4. Retrieve the products and their average prices:
5. Retrieve the average price of all products:

6. Retrieve the customers who have placed orders:

7. Retrieve the customers who have not placed any orders:

8. Retrieve the products with a price higher than the average price of all products:

9. Retrieve the customers who have placed orders with a total amount higher than the average total amount of all orders:

10. Retrieve the orders placed by customers who live at the same address as a specific customer (e.g., customer with ID 1):

11. Retrieve the customers who have placed the highest total amount order:

12. Retrieve the products that have been ordered by customers who have also ordered a specific product (e.g., product with ID 2):

13. Retrieve the customers who have placed orders with a total amount higher than the total amount of orders placed by customers living at the same address:

14. Retrieve the products that have been ordered more than the average quantity ordered for each product:

15. Retrieve the customers who have placed orders for all available products:

16. Retrieve the orders along with the corresponding customer information:

17. Retrieve the order items along with the corresponding product information:

18. Retrieve the customers who have placed orders and the total amount spent by each customer:

19. Retrieve the products that have been ordered and the total quantity ordered for each product:

20. Retrieve the orders along with the corresponding customer name and product name: