-- To create the ecommerce database:
```sql
CREATE DATABASE ecommerce;
```

-- To Select the ecommerce database:
```sql
USE ecommerce;
```

-- To create the customers table with id, name, email, and address:
```sql
CREATE TABLE customers (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  email VARCHAR(255) UNIQUE NOT NULL,
  address TEXT NOT NULL
);
```

-- To create the products table with id, name, price, and description:
```sql
CREATE TABLE products (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  price DECIMAL(10, 2) NOT NULL,
  description TEXT
);
```

-- To create the orders table with id, customer_id, order_date, and total_amount:
```sql
CREATE TABLE orders (
  id INT AUTO_INCREMENT PRIMARY KEY,
  customer_id INT,
  order_date DATE,
  total_amount DECIMAL(10, 2),
  FOREIGN KEY (customer_id) REFERENCES customers(id)
);
```


-- To insert sample customers into the customers table:
```sql
INSERT INTO customers (name, email, address) VALUES
('Alice', 'alice@example.com', '123 Apple St'),
('Bob', 'bob@example.com', '456 Banana Blvd'),
('Charlie', 'charlie@example.com', '789 Cherry Ct'),
('David', 'david@example.com', '1010 Date Dr'),
('Eve', 'eve@example.com', '1111 Elderberry Ln'),
('Frank', 'frank@example.com', '1212 Fig St'),
('Grace', 'grace@example.com', '1313 Grape Blvd'),
('Hank', 'hank@example.com', '1414 Hazelnut Ct'),
('Ivy', 'ivy@example.com', '1515 Iceberg Dr'),
('Jack', 'jack@example.com', '1616 Jackfruit Ln'),
('Kathy', 'kathy@example.com', '1717 Kiwi St'),
('Leo', 'leo@example.com', '1818 Lemon Blvd'),
('Mona', 'mona@example.com', '1919 Mango Ct'),
('Nina', 'nina@example.com', '2020 Nectarine Dr'),
('Oscar', 'oscar@example.com', '2121 Olive Ln'),
('Paul', 'paul@example.com', '2222 Papaya St'),
('Quincy', 'quincy@example.com', '2323 Quince Blvd'),
```

```
('Rita', 'rita@example.com', '2424 Raspberry Ct'),
('Sam', 'sam@example.com', '2525 Strawberry Dr'),
('Tina', 'tina@example.com', '2626 Tangerine Ln'),
('Uma', 'uma@example.com', '2727 Ugli Fruit St'),
('Vince', 'vince@example.com', '2828 Vanilla Blvd'),
('Wade', 'wade@example.com', '2929 Watermelon Ct'),
('Xena', 'xena@example.com', '3030 Xigua Dr'),
('Yara', 'yara@example.com', '3131 Yellow Passion Fruit Ln'),
('Zane', 'zane@example.com', '3232 Zucchini St'),
('Aiden', 'aiden@example.com', '3333 Apricot Blvd'),
('Bella', 'bella@example.com', '3434 Blueberry Ct'),
('Caleb', 'caleb@example.com', '3535 Currant Dr'),
('Diana', 'diana@example.com', '3636 Dragonfruit Ln'),
('Ethan', 'ethan@example.com', '3737 Elder Dr'),
('Fiona', 'fiona@example.com', '3838 Fig Ct'),
('George', 'george@example.com', '3939 Grape Dr'),
('Holly', 'holly@example.com', '4040 Hazelnut Blvd'),
('Ian', 'ian@example.com', '4141 Iceberg St'),
('Jane', 'jane@example.com', '4242 Jackfruit Ct'),
('Kyle', 'kyle@example.com', '4343 Kiwi Dr'),
('Laura', 'laura@example.com', '4444 Lemon Ln'),
('Mark', 'mark@example.com', '4545 Mango Blvd'),
('Nina', 'nina2@example.com', '4646 Nectarine St'),
('Oliver', 'oliver@example.com', '4747 Olive Ct'),
('Paula', 'paula@example.com', '4848 Papaya Dr'),
('Quinn', 'quinn@example.com', '4949 Quince Blvd'),
('Rachel', 'rachel@example.com', '5050 Raspberry Ct'),
('Steve', 'steve@example.com', '5151 Strawberry Dr'),
('Tara', 'tara@example.com', '5252 Tangerine Ln'),
('Uma', 'uma2@example.com', '5353 Ugli Fruit St'),
('Violet', 'violet@example.com', '5454 Vanilla Blvd'),
('Walt', 'walt@example.com', '5555 Watermelon Ct'),
('Johnson', 'johnson@example.com', '7389 Pine St');
```

```
```

--- Customers table:
```sql
select * from customers;
```

```
+----+---------+--------------------+---------------------------+
| id | name    | email              | address                   |
+----+---------+--------------------+---------------------------+
|  1 | Alice   | alice@example.com  | 123 Apple St              |
|  2 | Bob     | bob@example.com    | 456 Banana Blvd           |
|  3 | Charlie | charlie@example.com| 789 Cherry Ct             |
|  4 | David   | david@example.com  | 1010 Date Dr              |
|  5 | Eve     | eve@example.com    | 1111 Elderberry Ln        |
|  6 | Frank   | frank@example.com  | 1212 Fig St               |
|  7 | Grace   | grace@example.com  | 1313 Grape Blvd           |
|  8 | Hank    | hank@example.com   | 1414 Hazelnut Ct          |
|  9 | Ivy     | ivy@example.com    | 1515 Iceberg Dr           |
| 10 | Jack    | jack@example.com   | 1616 Jackfruit Ln         |
| 11 | Kathy   | kathy@example.com  | 1717 Kiwi St              |
| 12 | Leo     | leo@example.com    | 1818 Lemon Blvd           |
| 13 | Mona    | mona@example.com   | 1919 Mango Ct             |
| 14 | Nina    | nina@example.com   | 2020 Nectarine Dr         |
| 15 | Oscar   | oscar@example.com  | 2121 Olive Ln             |
| 16 | Paul    | paul@example.com   | 2222 Papaya St            |
| 17 | Quincy  | quincy@example.com | 2323 Quince Blvd          |
| 18 | Rita    | rita@example.com   | 2424 Raspberry Ct         |
| 19 | Sam     | sam@example.com    | 2525 Strawberry Dr        |
| 20 | Tina    | tina@example.com   | 2626 Tangerine Ln         |
| 21 | Uma     | uma@example.com    | 2727 Ugli Fruit St        |
| 22 | Vince   | vince@example.com  | 2828 Vanilla Blvd         |
| 23 | Wade    | wade@example.com   | 2929 Watermelon Ct        |
```

```
| 24 | Xena   | xena@example.com   | 3030 Xigua Dr            |
| 25 | Yara   | yara@example.com   | 3131 Yellow Passion Fruit Ln |
| 26 | Zane   | zane@example.com   | 3232 Zucchini St         |
| 27 | Aiden  | aiden@example.com  | 3333 Apricot Blvd        |
| 28 | Bella  | bella@example.com  | 3434 Blueberry Ct        |
| 29 | Caleb  | caleb@example.com  | 3535 Currant Dr          |
| 30 | Diana  | diana@example.com  | 3636 Dragonfruit Ln      |
| 31 | Ethan  | ethan@example.com  | 3737 Elder Dr            |
| 32 | Fiona  | fiona@example.com  | 3838 Fig Ct              |
| 33 | George | george@example.com | 3939 Grape Dr            |
| 34 | Holly  | holly@example.com  | 4040 Hazelnut Blvd       |
| 35 | Ian    | ian@example.com    | 4141 Iceberg St          |
| 36 | Jane   | jane@example.com   | 4242 Jackfruit Ct        |
| 37 | Kyle   | kyle@example.com   | 4343 Kiwi Dr             |
| 38 | Laura  | laura@example.com  | 4444 Lemon Ln            |
| 39 | Mark   | mark@example.com   | 4545 Mango Blvd          |
| 40 | Nina   | nina2@example.com  | 4646 Nectarine St        |
| 41 | Oliver | oliver@example.com | 4747 Olive Ct            |
| 42 | Paula  | paula@example.com  | 4848 Papaya Dr           |
| 43 | Quinn  | quinn@example.com  | 4949 Quince Blvd         |
| 44 | Rachel | rachel@example.com | 5050 Raspberry Ct        |
| 45 | Steve  | steve@example.com  | 5151 Strawberry Dr       |
| 46 | Tara   | tara@example.com   | 5252 Tangerine Ln        |
| 47 | Uma    | uma2@example.com   | 5353 Ugli Fruit St       |
| 48 | Violet | violet@example.com | 5454 Vanilla Blvd        |
| 49 | Walt   | walt@example.com   | 5555 Watermelon Ct       |
| 50 | Johnson | johnson@example.com | 7389 Pine St           |
+----+---------+-------------------+--------------------------+
```

-- To insert sample products into the products table:
```sql
```

```sql
INSERT INTO products (name, price, description) VALUES
('Eco-Friendly Water Bottle', 250.00, 'High-quality durable product ideal for everyday use'),
('Wireless Earbuds', 120.00, 'Premium product with excellent performance and longevity'),
('Smartphone Stand', 300.00, 'Affordable product with great value for money'),
('Bluetooth Speaker', 10.00, 'Innovative product with cutting-edge features'),
('Fitness Tracker', 500.00, 'Reliable product with consistent performance'),
('Portable Charger', 40.00, 'Ergonomic design for maximum comfort and efficiency'),
('Reusable Shopping Bag', 330.00, 'Eco-friendly product made with sustainable materials'),
('Travel Mug', 80.00, 'Compact product with sleek and modern design'),
('Laptop Sleeve', 270.00, 'Advanced technology for superior performance'),
('Noise-Canceling Headphones', 370.00, 'Top-rated product with outstanding customer reviews'),
('Smart Thermostat', 20.00, 'Versatile product suitable for various applications'),
('Digital Photo Frame', 130.00, 'High-performance product with exceptional durability'),
('Robot Vacuum', 450.00, 'Budget-friendly product with great features'),
('Electric Kettle', 60.00, 'High-efficiency product with low energy consumption'),
('LED Desk Lamp', 140.00, 'Luxurious product with premium quality materials'),
('Wireless Charger', 230.00, 'Compact and portable product for easy transportation'),
('Smart Light Bulbs', 150.00, 'User-friendly product with intuitive controls'),
('Air Purifier', 110.00, 'Sturdy product built to withstand heavy usage'),
('Smart Watch', 200.00, 'Elegant product with stylish design'),
('Portable Blender', 500.00, 'High-capacity product for large-scale usage'),
('Electric Toothbrush', 310.00, 'Precision-engineered product for accuracy and reliability'),
('Smart Lock', 70.00, 'Innovative product with advanced safety features'),
('Robot Mop', 190.00, 'Cost-effective product with great performance'),
('Streaming Stick', 320.00, 'Durable product made with high-quality materials'),
('Smart Scale', 410.00, 'Compact and efficient product for space-saving'),
('Wi-Fi Extender', 20.00, 'High-speed product for quick and efficient usage'),
('Dash Cam', 340.00, 'Eco-friendly product with recyclable materials'),
('Outdoor Security Camera', 220.00, 'Innovative product with multi-functional features'),
('Smart Doorbell', 160.00, 'Reliable product with long-lasting performance'),
('Wireless Mouse', 470.00, 'Stylish product with contemporary design'),
('Laptop Cooling Pad', 300.00, 'User-friendly product with easy setup and operation'),
('Bluetooth Keyboard', 240.00, 'Robust product with high-strength materials'),
```

```sql
('Smart Coffee Maker', 330.00, 'Versatile product for various applications'),
('Portable Fan', 10.00, 'Efficient product with energy-saving features'),
('Wireless Printer', 130.00, 'Premium product with luxurious features'),
('Wi-Fi Smart Plug', 360.00, 'Compact product ideal for small spaces'),
('Smart Sprinkler System', 160.00, 'Advanced product with innovative technology'),
('Electric Pressure Cooker', 240.00, 'High-quality product with superior performance'),
('Smart Air Fryer', 380.00, 'Affordable product with excellent value'),
('Noise Machine', 220.00, 'Elegant product with modern design'),
('Smart Door Sensor', 250.00, 'User-friendly product with easy-to-use features'),
('Water Leak Detector', 460.00, 'Durable product designed for long-term usage'),
('Smart Ceiling Fan', 90.00, 'Versatile product with multiple features'),
('UV Sanitizer', 20.00, 'Eco-friendly product with sustainable design'),
('Smart Garage Door Opener', 350.00, 'Reliable product with consistent quality'),
('Smart Bulb Socket', 310.00, 'Innovative product with cutting-edge technology'),
('Pet Camera', 100.00, 'Premium product with excellent craftsmanship'),
('Home Theater Projector', 140.00, 'High-efficiency product with low power consumption'),
('Bluetooth Tracker', 210.00, 'Stylish product with sleek and modern aesthetics'),
('Electric Fireplace', 470.00, 'Top-tier product with exceptional performance');
```

--- Products table:
```sql
select * from products;
```

```
+----+------------------------+--------+----------------------------------------------------+
| id | name                   | price  | description                                        |
+----+------------------------+--------+----------------------------------------------------+
|  1 | Eco-Friendly Water Bottle  | 250.00 | High-quality durable product ideal for everyday use      |
|  2 | Wireless Earbuds        | 120.00 | Premium product with excellent performance and longevity |
|  3 | Smartphone Stand        | 300.00 | Affordable product with great value for money         |
|  4 | Bluetooth Speaker       | 10.00 | Innovative product with cutting-edge features          |
```

5	Fitness Tracker	500.00	Reliable product with consistent performance
6	Portable Charger	40.00	Ergonomic design for maximum comfort and efficiency
7	Reusable Shopping Bag	330.00	Eco-friendly product made with sustainable materials
8	Travel Mug	80.00	Compact product with sleek and modern design
9	Laptop Sleeve	270.00	Advanced technology for superior performance
10	Noise-Canceling Headphones	370.00	Top-rated product with outstanding customer reviews
11	Smart Thermostat	20.00	Versatile product suitable for various applications
12	Digital Photo Frame	130.00	High-performance product with exceptional durability
13	Robot Vacuum	450.00	Budget-friendly product with great features
14	Electric Kettle	60.00	High-efficiency product with low energy consumption
15	LED Desk Lamp	140.00	Luxurious product with premium quality materials
16	Wireless Charger	230.00	Compact and portable product for easy transportation
17	Smart Light Bulbs	150.00	User-friendly product with intuitive controls
18	Air Purifier	110.00	Sturdy product built to withstand heavy usage
19	Smart Watch	200.00	Elegant product with stylish design
20	Portable Blender	500.00	High-capacity product for large-scale usage
21	Electric Toothbrush	310.00	Precision-engineered product for accuracy and reliability
22	Smart Lock	70.00	Innovative product with advanced safety features
23	Robot Mop	190.00	Cost-effective product with great performance
24	Streaming Stick	320.00	Durable product made with high-quality materials
25	Smart Scale	410.00	Compact and efficient product for space-saving
26	Wi-Fi Extender	20.00	High-speed product for quick and efficient usage
27	Dash Cam	340.00	Eco-friendly product with recyclable materials
28	Outdoor Security Camera	220.00	Innovative product with multi-functional features
29	Smart Doorbell	160.00	Reliable product with long-lasting performance
30	Wireless Mouse	470.00	Stylish product with contemporary design

31	Laptop Cooling Pad	300.00	User-friendly product with easy setup and operation
32	Bluetooth Keyboard	240.00	Robust product with high-strength materials
33	Smart Coffee Maker	330.00	Versatile product for various applications
34	Portable Fan	10.00	Efficient product with energy-saving features
35	Wireless Printer	130.00	Premium product with luxurious features
36	Wi-Fi Smart Plug	360.00	Compact product ideal for small spaces
37	Smart Sprinkler System	160.00	Advanced product with innovative technology
38	Electric Pressure Cooker	240.00	High-quality product with superior performance
39	Smart Air Fryer	380.00	Affordable product with excellent value
40	Noise Machine	220.00	Elegant product with modern design
41	Smart Door Sensor	250.00	User-friendly product with easy-to-use features
42	Water Leak Detector	460.00	Durable product designed for long-term usage
43	Smart Ceiling Fan	90.00	Versatile product with multiple features
44	UV Sanitizer	20.00	Eco-friendly product with sustainable design
45	Smart Garage Door Opener	350.00	Reliable product with consistent quality
46	Smart Bulb Socket	310.00	Innovative product with cutting-edge technology
47	Pet Camera	100.00	Premium product with excellent craftsmanship
48	Home Theater Projector	140.00	High-efficiency product with low power consumption
49	Bluetooth Tracker	210.00	Stylish product with sleek and modern aesthetics
50	Electric Fireplace	470.00	Top-tier product with exceptional performance

```
+---+-----------------------+-------+---------------------------------------------------+
```

-- To insert sample orders into the orders table:
```sql
INSERT INTO orders (customer_id, order_date, total_amount) VALUES
(1, '2024-03-05', 550.00),
(2, '2024-01-19', 850.00),
(3, '2024-02-11', 1300.00),
(4, '2024-04-15', 250.00),
(5, '2024-02-23', 1200.00),
(6, '2024-03-20', 650.00),
(7, '2024-04-05', 750.00),
(8, '2024-03-18', 450.00),
(9, '2024-02-10', 1500.00),
(10, '2024-01-27', 1150.00),
(11, '2024-04-12', 2000.00),
(12, '2024-02-02', 300.00),
(13, '2024-03-22', 1750.00),
(14, '2024-04-01', 100.00),
(15, '2024-01-30', 950.00),
(16, '2024-02-25', 1950.00),
(17, '2024-03-29', 600.00),
(18, '2024-04-04', 200.00),
(19, '2024-01-24', 1450.00),
(20, '2024-02-09', 1600.00),
(21, '2024-03-16', 1700.00),
(22, '2024-02-18', 1250.00),
(23, '2024-03-01', 1800.00),
(24, '2024-04-10', 350.00),
(25, '2024-01-15', 900.00),
(26, '2024-02-27', 700.00),
```

```sql
(27, '2024-03-25', 1350.00),
(28, '2024-04-08', 500.00),
(29, '2024-01-18', 1550.00),
(30, '2024-02-05', 1000.00),
(31, '2024-03-07', 2000.00),
(32, '2024-04-02', 800.00),
(33, '2024-01-20', 1800.00),
(34, '2024-02-13', 1050.00),
(35, '2024-03-10', 2500.00),
(36, '2024-04-06', 750.00),
(37, '2024-01-22', 2100.00),
(38, '2024-02-07', 1100.00),
(39, '2024-03-14', 2200.00),
(40, '2024-04-03', 600.00),
(41, '2024-01-16', 1450.00),
(42, '2024-02-12', 1150.00),
(43, '2024-03-11', 1900.00),
(44, '2024-04-09', 400.00),
(45, '2024-01-29', 1600.00),
(46, '2024-02-28', 1950.00),
(47, '2024-03-13', 650.00),
(48, '2024-04-11', 800.00),
(49, '2024-01-23', 1500.00),
(50, '2024-02-17', 1700.00);
```

--- Orders table:
```sql
select * from orders;
```

```
+----+-------------+------------+--------------+
| id | customer_id | order_date | total_amount |
```

#	ID	Date	Amount
1	1	2024-03-05	550.00
2	2	2024-01-19	850.00
3	3	2024-02-11	1300.00
4	4	2024-04-15	250.00
5	5	2024-02-23	1200.00
6	6	2024-03-20	650.00
7	7	2024-04-05	750.00
8	8	2024-03-18	450.00
9	9	2024-02-10	1500.00
10	10	2024-01-27	1150.00
11	11	2024-04-12	2000.00
12	12	2024-02-02	300.00
13	13	2024-03-22	1750.00
14	14	2024-04-01	100.00
15	15	2024-01-30	950.00
16	16	2024-02-25	1950.00
17	17	2024-03-29	600.00
18	18	2024-04-04	200.00
19	19	2024-01-24	1450.00
20	20	2024-02-09	1600.00
21	21	2024-03-16	1700.00
22	22	2024-02-18	1250.00
23	23	2024-03-01	1800.00
24	24	2024-04-10	350.00
25	25	2024-01-15	900.00
26	26	2024-02-27	700.00
27	27	2024-03-25	1350.00
28	28	2024-04-08	500.00
29	29	2024-01-18	1550.00
30	30	2024-02-05	1000.00
31	31	2024-03-07	2000.00
32	32	2024-04-02	800.00

```
| 33 |        33 | 2024-01-20 |    1800.00 |
| 34 |        34 | 2024-02-13 |    1050.00 |
| 35 |        35 | 2024-03-10 |    2500.00 |
| 36 |        36 | 2024-04-06 |     750.00 |
| 37 |        37 | 2024-01-22 |    2100.00 |
| 38 |        38 | 2024-02-07 |    1100.00 |
| 39 |        39 | 2024-03-14 |    2200.00 |
| 40 |        40 | 2024-04-03 |     600.00 |
| 41 |        41 | 2024-01-16 |    1450.00 |
| 42 |        42 | 2024-02-12 |    1150.00 |
| 43 |        43 | 2024-03-11 |    1900.00 |
| 44 |        44 | 2024-04-09 |     400.00 |
| 45 |        45 | 2024-01-29 |    1600.00 |
| 46 |        46 | 2024-02-28 |    1950.00 |
| 47 |        47 | 2024-03-13 |     650.00 |
| 48 |        48 | 2024-04-11 |     800.00 |
| 49 |        49 | 2024-01-23 |    1500.00 |
| 50 |        50 | 2024-02-17 |    1700.00 |
+----+------------+-----------+------------+
```

Task queries to Write:

-- Retrieve all customers who have placed an order in the last 30 days:
(Data which i retrieved is between first 4 months which is not equal to current date so that let me reconsider 30 days to 230 days for current day to interval day)
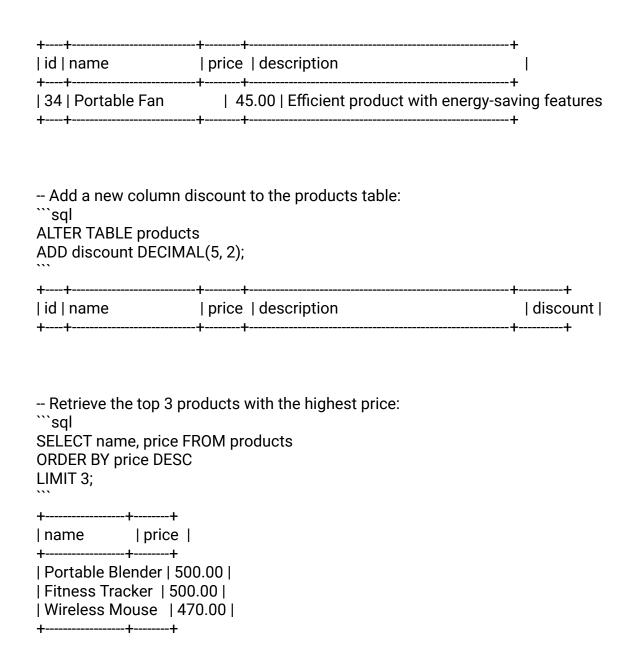```sql
SELECT DISTINCT customers.name
FROM customers
```

```
JOIN orders ON customers.id = orders.customer_id
WHERE order_date >= CURDATE() - INTERVAL 230 DAY;
```

+--------+
| name |
+--------+
| David |
| Grace |
| Kathy |
| Mona |
| Nina |
| Quincy |
| Rita |
| Xena |
| Aiden |
| Bella |
| Fiona |
| Jane |
| Rachel |
| Violet |
+--------+

-- Get the total amount of all orders placed by each customer:
```sql
SELECT customers.name, SUM(orders.total_amount) AS total_spent FROM customers
JOIN orders ON customers.id = orders.customer_id
GROUP BY customers.name;
```

+---------+-------------+
| name | total_spent |
+---------+-------------+

Alice	550.00
Bob	850.00
Charlie	1300.00
David	250.00
Eve	1200.00
Frank	650.00
Grace	750.00
Hank	450.00
Ivy	1500.00
Jack	1150.00
Kathy	2000.00
Leo	300.00
Mona	1750.00
Nina	700.00
Oscar	950.00
Paul	1950.00
Quincy	600.00
Rita	200.00
Sam	1450.00
Tina	1600.00
Uma	2350.00
Vince	1250.00
Wade	1800.00
Xena	350.00
Yara	900.00
Zane	700.00
Aiden	1350.00
Bella	500.00
Caleb	1550.00
Diana	1000.00
Ethan	2000.00
Fiona	800.00
George	1800.00

```
| Holly   |    1050.00 |
| Ian     |    2500.00 |
| Jane    |     750.00 |
| Kyle    |    2100.00 |
| Laura   |    1100.00 |
| Mark    |    2200.00 |
| Oliver  |    1450.00 |
| Paula   |    1150.00 |
| Quinn   |    1900.00 |
| Rachel  |     400.00 |
| Steve   |    1600.00 |
| Tara    |    1950.00 |
| Violet  |     800.00 |
| Walt    |    1500.00 |
| Johnson |    1700.00 |
+--------+------------+
```

-- Update the price of Product C to 45.00:
```sql
UPDATE products SET price = 45.00
WHERE name = 'Portable Fan';
```

--- Before update
```
+----+------------------------+--------+-------------------------------------------------+
| id | name                   | price  | description                                     |
+----+------------------------+--------+-------------------------------------------------+
| 34 | Portable Fan           | 10.00  | Efficient product with energy-saving features   |
+----+------------------------+--------+-------------------------------------------------+
```

--- After update

```
+----+-----------------------+-------+-------------------------------------------------------+
| id | name                  | price | description                                           |
+----+-----------------------+-------+-------------------------------------------------------+
| 34 | Portable Fan          | 45.00 | Efficient product with energy-saving features         |
+----+-----------------------+-------+-------------------------------------------------------+
```

-- Add a new column discount to the products table:
```sql
ALTER TABLE products
ADD discount DECIMAL(5, 2);
```

```
+----+-----------------------+-------+-------------------------------------------------------+----------+
| id | name                  | price | description                                           | discount |
+----+-----------------------+-------+-------------------------------------------------------+----------+
```

-- Retrieve the top 3 products with the highest price:
```sql
SELECT name, price FROM products
ORDER BY price DESC
LIMIT 3;
```

```
+-----------------+-------+
| name            | price |
+-----------------+-------+
| Portable Blender | 500.00 |
| Fitness Tracker  | 500.00 |
| Wireless Mouse   | 470.00 |
+-----------------+-------+
```

-- Get the names of customers who have ordered Product A:
(Consider 'Product A' as 'Wireless Charger')
```sql
SELECT DISTINCT customers.name FROM customers
JOIN orders ON customers.id = orders.customer_id
JOIN products ON products.id IN (SELECT id FROM products WHERE name = 'Wireless Charger');
```

```
+---------+
| name    |
+---------+
| Alice   |
| Bob     |
| Charlie |
| David   |
| Eve     |
| Frank   |
| Grace   |
| Hank    |
| Ivy     |
| Jack    |
| Kathy   |
| Leo     |
| Mona    |
| Nina    |
| Oscar   |
| Paul    |
| Quincy  |
| Rita    |
| Sam     |
| Tina    |
| Uma     |
```

```
| Vince  |
| Wade   |
| Xena   |
| Yara   |
| Zane   |
| Aiden  |
| Bella  |
| Caleb  |
| Diana  |
| Ethan  |
| Fiona  |
| George |
| Holly  |
| Ian    |
| Jane   |
| Kyle   |
| Laura  |
| Mark   |
| Oliver |
| Paula  |
| Quinn  |
| Rachel |
| Steve  |
| Tara   |
| Violet |
| Walt   |
| Johnson |
+--------+
```

-- Join the orders and customers tables to retrieve the customer's name and order date for each order:
```sql
```

```
SELECT customers.name, orders.order_date FROM customers
JOIN orders ON customers.id = orders.customer_id;
```

```
+---------+------------+
| name    | order_date |
+---------+------------+
| Alice   | 2024-03-05 |
| Bob     | 2024-01-19 |
| Charlie | 2024-02-11 |
| David   | 2024-04-15 |
| Eve     | 2024-02-23 |
| Frank   | 2024-03-20 |
| Grace   | 2024-04-05 |
| Hank    | 2024-03-18 |
| Ivy     | 2024-02-10 |
| Jack    | 2024-01-27 |
| Kathy   | 2024-04-12 |
| Leo     | 2024-02-02 |
| Mona    | 2024-03-22 |
| Nina    | 2024-04-01 |
| Oscar   | 2024-01-30 |
| Paul    | 2024-02-25 |
| Quincy  | 2024-03-29 |
| Rita    | 2024-04-04 |
| Sam     | 2024-01-24 |
| Tina    | 2024-02-09 |
| Uma     | 2024-03-16 |
| Vince   | 2024-02-18 |
| Wade    | 2024-03-01 |
| Xena    | 2024-04-10 |
| Yara    | 2024-01-15 |
| Zane    | 2024-02-27 |
| Aiden   | 2024-03-25 |
```

```
| Bella   | 2024-04-08 |
| Caleb   | 2024-01-18 |
| Diana   | 2024-02-05 |
| Ethan   | 2024-03-07 |
| Fiona   | 2024-04-02 |
| George  | 2024-01-20 |
| Holly   | 2024-02-13 |
| Ian     | 2024-03-10 |
| Jane    | 2024-04-06 |
| Kyle    | 2024-01-22 |
| Laura   | 2024-02-07 |
| Mark    | 2024-03-14 |
| Nina    | 2024-04-03 |
| Oliver  | 2024-01-16 |
| Paula   | 2024-02-12 |
| Quinn   | 2024-03-11 |
| Rachel  | 2024-04-09 |
| Steve   | 2024-01-29 |
| Tara    | 2024-02-28 |
| Uma     | 2024-03-13 |
| Violet  | 2024-04-11 |
| Walt    | 2024-01-23 |
| Johnson | 2024-02-17 |
+--------+-----------+
```

-- Retrieve the orders with a total amount greater than 150.00:
```sql
SELECT * FROM orders
WHERE total_amount > 150.00;
```

```
+----+------------+-----------+-------------+
```

| id | customer_id | order_date | total_amount |
+----+-------------+------------+--------------+
1	1	2024-03-05	550.00
2	2	2024-01-19	850.00
3	3	2024-02-11	1300.00
4	4	2024-04-15	250.00
5	5	2024-02-23	1200.00
6	6	2024-03-20	650.00
7	7	2024-04-05	750.00
8	8	2024-03-18	450.00
9	9	2024-02-10	1500.00
10	10	2024-01-27	1150.00
11	11	2024-04-12	2000.00
12	12	2024-02-02	300.00
13	13	2024-03-22	1750.00
15	15	2024-01-30	950.00
16	16	2024-02-25	1950.00
17	17	2024-03-29	600.00
18	18	2024-04-04	200.00
19	19	2024-01-24	1450.00
20	20	2024-02-09	1600.00
21	21	2024-03-16	1700.00
22	22	2024-02-18	1250.00
23	23	2024-03-01	1800.00
24	24	2024-04-10	350.00
25	25	2024-01-15	900.00
26	26	2024-02-27	700.00
27	27	2024-03-25	1350.00
28	28	2024-04-08	500.00
29	29	2024-01-18	1550.00
30	30	2024-02-05	1000.00
31	31	2024-03-07	2000.00
32	32	2024-04-02	800.00

```
| 33 |          33 | 2024-01-20 |     1800.00 |
| 34 |          34 | 2024-02-13 |     1050.00 |
| 35 |          35 | 2024-03-10 |     2500.00 |
| 36 |          36 | 2024-04-06 |      750.00 |
| 37 |          37 | 2024-01-22 |     2100.00 |
| 38 |          38 | 2024-02-07 |     1100.00 |
| 39 |          39 | 2024-03-14 |     2200.00 |
| 40 |          40 | 2024-04-03 |      600.00 |
| 41 |          41 | 2024-01-16 |     1450.00 |
| 42 |          42 | 2024-02-12 |     1150.00 |
| 43 |          43 | 2024-03-11 |     1900.00 |
| 44 |          44 | 2024-04-09 |      400.00 |
| 45 |          45 | 2024-01-29 |     1600.00 |
| 46 |          46 | 2024-02-28 |     1950.00 |
| 47 |          47 | 2024-03-13 |      650.00 |
| 48 |          48 | 2024-04-11 |      800.00 |
| 49 |          49 | 2024-01-23 |     1500.00 |
| 50 |          50 | 2024-02-17 |     1700.00 |
+----+------------+-----------+------------+
```

-- Normalize the database by creating a separate table for order items and updating the orders table to reference the order_items table:
First, create the order_items table:
```sql
CREATE TABLE order_items (
  id INT AUTO_INCREMENT PRIMARY KEY,
  order_id INT,
  product_id INT,
  quantity INT,
  FOREIGN KEY (order_id) REFERENCES orders(id),
  FOREIGN KEY (product_id) REFERENCES products(id));
```

Next, update the orders table:
```sql
ALTER TABLE orders
ADD COLUMN order_item_id INT,
ADD FOREIGN KEY (order_item_id) REFERENCES order_items(id);
```

-- Retrieve the average total of all orders:
```sql
SELECT AVG(total_amount) AS average_order_total
FROM orders;
```

```
+--------------------+
| average_order_total |
+--------------------+
|        1173.000000 |
+--------------------+
```