

-- Create Database

```
CREATE DATABASE ecommerce;
```

```
USE ecommerce;
```

-- Create Customers Table

```
CREATE TABLE customers (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    address VARCHAR(255)  
);
```

-- Create Products Table

```
CREATE TABLE products (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
    description TEXT,  
    discount DECIMAL(5, 2) DEFAULT 0.00  
);
```

-- Create Orders Table

```
CREATE TABLE orders (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    customer_id INT,  
    order_date DATE NOT NULL,  
    total_amount DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (customer_id) REFERENCES customers(id)  
);
```

-- Insert Sample Data into Customers

```
INSERT INTO customers (name, email, address) VALUES

('Rahul Sharma', 'rahul.sharma@example.com', 'Flat 12, Prestige Apartments, Bangalore, Karnataka'),

('Priya Patel', 'priya.patel@example.com', '45 Gandhi Road, Mumbai, Maharashtra'),

('Amit Kumar', 'amit.kumar@example.com', 'House 7, Green Colony, Delhi'),

('Anjali Reddy', 'anjali.reddy@example.com', '22 Park View, Hyderabad, Telangana'),

('Vikram Singh', 'vikram.singh@example.com', 'Sector 15, Noida, Uttar Pradesh'),

('Deepa Mehta', 'deepa.mehta@example.com', '15 Lake View, Ahmedabad, Gujarat'),

('Rajesh Gupta', 'rajesh.gupta@example.com', 'Apartment 3B, Salt Lake, Kolkata, West Bengal'),

('Shruti Iyer', 'shruti.iyer@example.com', '78 Brahmin Street, Chennai, Tamil Nadu'),

('Arun Nair', 'arun.nair@example.com', 'Villa 22, Kochi, Kerala'),

('Neha Joshi', 'neha.joshi@example.com', '56 University Road, Pune, Maharashtra'),

('Sanjay Desai', 'sanjay.desai@example.com', 'Flat 45, Satellite Area, Ahmedabad'),

('Kavita Bose', 'kavita.bose@example.com', '12 Park Street, Kolkata'),

('Manish Malhotra', 'manish.malhotra@example.com', 'DLF Phase 2, Gurgaon, Haryana'),

('Pooja Khanna', 'pooja.khanna@example.com', 'Sector 42, Chandigarh'),

('Suresh Babu', 'suresh.babu@example.com', '89 Residency Road, Bangalore, Karnataka');
```

-- Retrieving the inserted sample data from customers

```
SELECT * From customers;
```

Output:

id	name	email	address
1	Rahul Sharma	rahul.sharma@example.com	Flat 12, Prestige Apartments, Bangalore, Karnataka
2	Priya Patel	priya.patel@example.com	45 Gandhi Road, Mumbai, Maharashtra
3	Amit Kumar	amit.kumar@example.com	House 7, Green Colony, Delhi
4	Anjali Reddy	anjali.reddy@example.com	22 Park View, Hyderabad, Telangana
5	Vikram Singh	vikram.singh@example.com	Sector 15, Noida, Uttar Pradesh
6	Deepa Mehta	deepa.mehta@example.com	15 Lake View, Ahmedabad, Gujarat
7	Rajesh Gupta	rajesh.gupta@example.com	Apartment 3B, Salt Lake, Kolkata, West Bengal
8	Shruti Iyer	shruti.iyer@example.com	78 Brahmin Street, Chennai, Tamil Nadu
9	Arun Nair	arun.nair@example.com	Villa 22, Kochi, Kerala
10	Neha Joshi	neha.joshi@example.com	56 University Road, Pune, Maharashtra
11	Sanjay Desai	sanjay.desai@example.com	Flat 45, Satellite Area, Ahmedabad
12	Kavita Bose	kavita.bose@example.com	12 Park Street, Kolkata
13	Manish Malhotra	manish.malhotra@example.com	DLF Phase 2, Gurgaon, Haryana
14	Pooja Khanna	pooja.khanna@example.com	Sector 42, Chandigarh
15	Suresh Babu	suresh.babu@example.com	89 Residency Road, Bangalore, Karnataka

-- Insert Sample Data into Products (15 records with real product names)

INSERT INTO products (name, price, description) VALUES

('Apple iPhone 13', 54999.00, 'Powerful smartphone with advanced dual-camera system'),
('Samsung 65" QLED 4K Smart TV', 129999.00, 'Immersive viewing experience with Quantum Dot technology'),
('Kindle Paperwhite', 13999.00, 'High-resolution e-reader with adjustable warm light'),
('Noise ColorFit Ultra Smart Watch', 2999.00, 'Fitness tracker with health monitoring features'),
('Sony WH-1000XM4 Noise Cancelling Headphones', 29999.00, 'Premium wireless headphones with industry-leading noise cancellation'),
('Xiaomi Mi Robot Vacuum', 24999.00, 'Smart robotic vacuum cleaner with advanced navigation'),
('OnePlus Buds Pro', 9999.00, 'Wireless earbuds with active noise cancellation'),
('Dyson V11 Absolute Vacuum Cleaner', 52999.00, 'Powerful cordless vacuum with intelligent suction'),
('LG 1.5 Ton Inverter Split AC', 44999.00, 'Energy-efficient air conditioner with cooling technology'),
('Fitbit Versa 3 Smartwatch', 18999.00, 'Advanced fitness and health smartwatch'),
('Canon EOS R6 Mirrorless Camera', 159999.00, 'Professional-grade full-frame mirrorless camera'),
('Bose SoundLink Revolve+ Bluetooth Speaker', 19999.00, 'Portable 360-degree sound speaker'),
('HP Spectre x360 Laptop', 139999.00, 'Premium convertible laptop with Intel Core i7 processor'),
('Apple AirPods Pro', 19900.00, 'Wireless earbuds with active noise cancellation'),
('Mi Smart Electric Air Purifier', 12999.00, 'Advanced air purification with HEPA filter');

-- Retrieving the inserted sample data from products

SELECT * From products;

id	name	price	description
1	Apple iPhone 13	54999.00	Powerful smartphone with advanced dual-camera system
2	Samsung 65" QLED 4K Smart TV	129999.00	Immersive viewing experience with Quantum Dot technology
3	Kindle Paperwhite	13999.00	High-resolution e-reader with adjustable warm light
4	Noise ColorFit Ultra Smart Watch	2999.00	Fitness tracker with health monitoring features
5	Sony WH-1000XM4 Noise Cancelling Headphones	29999.00	Premium wireless headphones with industry-leading noise cancellation
6	Xiaomi Mi Robot Vacuum	24999.00	Smart robotic vacuum cleaner with advanced navigation
7	OnePlus Buds Pro	9999.00	Wireless earbuds with active noise cancellation
8	Dyson V11 Absolute Vacuum Cleaner	52999.00	Powerful cordless vacuum with intelligent suction
9	LG 1.5 Ton Inverter Split AC	44999.00	Energy-efficient air conditioner with cooling technology
10	Fitbit Versa 3 Smartwatch	18999.00	Advanced fitness and health smartwatch
11	Canon EOS R6 Mirrorless Camera	159999.00	Professional-grade full-frame mirrorless camera
12	Bose SoundLink Revolve+ Bluetooth Speaker	19999.00	Portable 360-degree sound speaker
13	HP Spectre x360 Laptop	139999.00	Premium convertible laptop with Intel Core i7 processor
14	Apple AirPods Pro	19900.00	Wireless earbuds with active noise cancellation
15	Mi Smart Electric Air Purifier	12999.00	Advanced air purification with HEPA filter

-- Insert Sample Data into Orders

INSERT INTO orders (customer_id, order_date, total_amount) VALUES

(1, '2024-06-10', 54999.00),
(2, '2016-04-15', 159999.00),
(3, '2024-03-20', 44999.00),
(4, '2024-01-25', 52999.00),
(5, '2023-03-30', 29999.00),
(6, '2019-02-05', 19999.00),
(7, '2024-09-10', 13999.00),
(8, '2023-07-15', 139999.00),
(9, '2024-07-20', 24999.00),
(10, '2024-07-25', 18999.00),
(11, '2023-11-30', 9999.00),
(12, '2024-08-05', 12999.00),
(13, '2014-12-10', 19900.00),
(14, '2004-05-15', 2999.00),
(15, '2022-08-20', 129999.00);

-- Retrieving the inserted sample data from orders

SELECT * From orders;

id	customer_id	order_date	total_amount
1	1	2024-06-10	54999.00
2	2	2016-04-15	159999.00
3	3	2024-03-20	44999.00
4	4	2024-01-25	52999.00
5	5	2023-03-30	29999.00
6	6	2019-02-05	19999.00
7	7	2024-09-10	13999.00
8	8	2023-07-15	139999.00
9	9	2024-07-20	24999.00
10	10	2024-07-25	18999.00
11	11	2024-11-30	9999.00
12	12	2024-08-05	12999.00
13	13	2024-12-10	19900.00
14	14	2004-05-15	2999.00
15	15	2022-08-20	129999.00
16	3	2024-09-11	13999.00
17	8	2024-10-16	139999.00
18	3	2023-07-20	24999.00
19	10	2024-07-25	18999.00
20	10	2024-12-03	9999.00
21	7	2023-10-05	12999.00
22	3	2020-07-15	159999.00
23	5	2021-03-20	44999.00
24	4	2024-11-30	52999.00

-- Query 1: Retrieve customers who placed an order in the last 30 days

SELECT DISTINCT c.id, c.name, c.email

FROM customers c

JOIN orders o ON c.id = o.customer_id

WHERE o.order_date >= DATE(NOW()) - INTERVAL 30 DAY;

id	name	email
11	Sanjay Desai	sanjay.desai@example.com
13	Manish Malhotra	manish.malhotra@example.com
10	Neha Joshi	neha.joshi@example.com
4	Anjali Reddy	anjali.reddy@example.com

-- Query 2: Total amount of orders placed by each customer

```
SELECT c.id, c.name, SUM(o.total_amount) AS total_customer_orders
```

```
FROM customers c
```

```
LEFT JOIN orders o ON c.id = o.customer_id
```

```
GROUP BY c.id;
```

id	name	total_customer_orders
1	Rahul Sharma	54999.00
2	Priya Patel	159999.00
3	Amit Kumar	243996.00
4	Anjali Reddy	105998.00
5	Vikram Singh	74998.00
6	Deepa Mehta	19999.00
7	Rajesh Gupta	26998.00
8	Shruti Iyer	279998.00
9	Arun Nair	24999.00
10	Neha Joshi	47997.00
11	Sanjay Desai	9999.00
12	Kavita Bose	12999.00
13	Manish Malhotra	19900.00
14	Pooja Khanna	2999.00
15	Suresh Babu	129999.00

-- Query 3: Update price of a specific product (e.g., OnePlus Buds Pro)

UPDATE products

SET price = 45.00

WHERE name = 'OnePlus Buds Pro';

SELECT * FROM products;

SELECT * FROM products WHERE name = 'OnePlus Buds Pro';

id	name	price	description
1	Apple iPhone 13	54999.00	Powerful smartphone with advanced dual-camera system
2	Samsung 65" QLED 4K Smart TV	129999.00	Immersive viewing experience with Quantum Dot technology
3	Kindle Paperwhite	13999.00	High-resolution e-reader with adjustable warm light
4	Noise ColorFit Ultra Smart Watch	2999.00	Fitness tracker with health monitoring features
5	Sony WH-1000XM4 Noise Cancelling Headphones	29999.00	Premium wireless headphones with industry-leading noise cancellation
6	Xiaomi Mi Robot Vacuum	24999.00	Smart robotic vacuum cleaner with advanced navigation
7	OnePlus Buds Pro	45.00	Wireless earbuds with active noise cancellation
8	Dyson V11 Absolute Vacuum Cleaner	52999.00	Powerful cordless vacuum with intelligent suction
9	LG 1.5 Ton Inverter Split AC	44999.00	Energy-efficient air conditioner with cooling technology
10	Fitbit Versa 3 Smartwatch	18999.00	Advanced fitness and health smartwatch
11	Canon EOS R6 Mirrorless Camera	159999.00	Professional-grade full-frame mirrorless camera
12	Bose SoundLink Revolve+ Bluetooth Speaker	19999.00	Portable 360-degree sound speaker
13	HP Spectre x360 Laptop	139999.00	Premium convertible laptop with Intel Core i7 processor
14	Apple AirPods Pro	19900.00	Wireless earbuds with active noise cancellation
15	Mi Smart Electric Air Purifier	12999.00	Advanced air purification with HEPA filter

id	name	price	description
7	OnePlus Buds Pro	45.00	Wireless earbuds with active noise cancellation

-- Query 4: Add a new column "discount" to the products table

ALTER TABLE products

ADD COLUMN discount DECIMAL(5, 2) DEFAULT 0.00;

SELECT * FROM products;

id	name	price	description	discount
1	Apple iPhone 13	54999.00	Powerful smartphone with advanced dual-camera system	0.00
2	Samsung 65" QLED 4K Smart TV	129999.00	Immersive viewing experience with Quantum Dot technology	0.00
3	Kindle Paperwhite	13999.00	High-resolution e-reader with adjustable warm light	0.00
4	Noise ColorFit Ultra Smart Watch	2999.00	Fitness tracker with health monitoring features	0.00
5	Sony WH-1000XM4 Noise Cancelling Headphones	29999.00	Premium wireless headphones with industry-leading noise cancellation	0.00
6	Xiaomi Mi Robot Vacuum	24999.00	Smart robotic vacuum cleaner with advanced navigation	0.00
7	OnePlus Buds Pro	45.00	Wireless earbuds with active noise cancellation	0.00
8	Dyson V11 Absolute Vacuum Cleaner	52999.00	Powerful cordless vacuum with intelligent suction	0.00
9	LG 1.5 Ton Inverter Split AC	44999.00	Energy-efficient air conditioner with cooling technology	0.00
10	Fitbit Versa 3 Smartwatch	18999.00	Advanced fitness and health smartwatch	0.00
11	Canon EOS R6 Mirrorless Camera	159999.00	Professional-grade full-frame mirrorless camera	0.00
12	Bose SoundLink Revolve+ Bluetooth Speaker	19999.00	Portable 360-degree sound speaker	0.00
13	HP Spectre x360 Laptop	139999.00	Premium convertible laptop with Intel Core i7 processor	0.00
14	Apple AirPods Pro	19900.00	Wireless earbuds with active noise cancellation	0.00
15	Mi Smart Electric Air Purifier	12999.00	Advanced air purification with HEPA filter	0.00

-- Query 5: Top 3 products with highest price

SELECT id, name, price

FROM products

ORDER BY price DESC

LIMIT 3;

id	name	price
11	Canon EOS R6 Mirrorless Camera	159999.00
13	HP Spectre x360 Laptop	139999.00
2	Samsung 65" QLED 4K Smart TV	129999.00

-- Query 9: Normalize the database by creating an order_items table

```
CREATE TABLE order_items (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    order_id INT NOT NULL,  
    product_id INT NOT NULL,  
    quantity INT NOT NULL,  
    FOREIGN KEY (order_id) REFERENCES orders(id),  
    FOREIGN KEY (product_id) REFERENCES products(id)  
);
```

-- Insert Sample Data into Order Items

```
INSERT INTO order_items (order_id, product_id, quantity) VALUES  
(1, 1, 1),  
(2, 11, 1),  
(3, 9, 1),  
(4, 8, 1),  
(5, 5, 1),  
(6, 12, 1),  
(7, 3, 1),  
(8, 13, 1),  
(9, 6, 1),
```


(10, 10, 1),
(11, 7, 1),
(12, 15, 1),
(13, 14, 1),
(14, 4, 1),
(15, 2, 1),
(16, 3, 1),
(17, 13, 1),
(18, 6, 1),
(19, 10, 1),
(20, 7, 1),
(21, 15, 1),
(22, 11, 1),
(23, 9, 1),
(24, 8, 1);

SELECT * FROM order_items;

id	order_id	product_id	quantity
1	1	1	1
2	2	11	1
3	3	9	1
4	4	8	1
5	5	5	1
6	6	12	1
7	7	3	1
8	8	13	1
9	9	6	1
10	10	10	1
11	11	7	1
12	12	15	1
13	13	14	1
14	14	4	1
15	15	2	1
16	16	3	1
17	17	13	1
18	18	6	1
19	19	10	1
20	20	7	1
21	21	15	1
22	22	11	1
23	23	9	1
24	24	8	1

-- Query 6: Names of customers who ordered a specific product (e.g., LG 1.5 Ton Inverter Split AC)

SELECT DISTINCT c.name

FROM customers c

JOIN orders o ON c.id = o.customer_id

JOIN order_items oi ON o.id = oi.order_id

JOIN products p ON oi.product_id = p.id

```
WHERE p.name = 'LG 1.5 Ton Inverter Split AC';
```

```
+-----+
| name   |
+-----+
| Amit Kumar   |
| Vikram Singh |
+-----+
```

```
-- Query 7: Join orders and customers to get customer name and order date
```

```
SELECT c.id, c.name, o.order_date, o.id
```

```
FROM orders o
```

```
JOIN customers c ON o.customer_id = c.id;
```

id	name	order_date	id
1	Rahul Sharma	2024-06-10	1
2	Priya Patel	2016-04-15	2
3	Amit Kumar	2024-03-20	3
4	Anjali Reddy	2024-01-25	4
5	Vikram Singh	2023-03-30	5
6	Deepa Mehta	2019-02-05	6
7	Rajesh Gupta	2024-09-10	7
8	Shruti Iyer	2023-07-15	8
9	Arun Nair	2024-07-20	9
10	Neha Joshi	2024-07-25	10
11	Sanjay Desai	2024-11-30	11
12	Kavita Bose	2024-08-05	12
13	Manish Malhotra	2024-12-10	13
14	Pooja Khanna	2004-05-15	14
15	Suresh Babu	2022-08-20	15
3	Amit Kumar	2024-09-11	16
8	Shruti Iyer	2024-10-16	17
3	Amit Kumar	2023-07-20	18
10	Neha Joshi	2024-07-25	19
10	Neha Joshi	2024-12-03	20
7	Rajesh Gupta	2023-10-05	21
3	Amit Kumar	2020-07-15	22
5	Vikram Singh	2021-03-20	23
4	Anjali Reddy	2024-11-30	24

-- Query 8: Orders with total amount greater than 15000

SELECT *

FROM orders

WHERE total_amount > 15000;

id	customer_id	order_date	total_amount
1	1	2024-06-10	54999.00
2	2	2016-04-15	159999.00
3	3	2024-03-20	44999.00
4	4	2024-01-25	52999.00
5	5	2023-03-30	29999.00
6	6	2019-02-05	19999.00
8	8	2023-07-15	139999.00
9	9	2024-07-20	24999.00
10	10	2024-07-25	18999.00
13	13	2024-12-10	19900.00
15	15	2022-08-20	129999.00
17	8	2024-10-16	139999.00
18	3	2023-07-20	24999.00
19	10	2024-07-25	18999.00
22	3	2020-07-15	159999.00
23	5	2021-03-20	44999.00
24	4	2024-11-30	52999.00

-- Query 9: Normalize the database by creating an order_items table

```
CREATE TABLE order_items (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    product_id INT NOT NULL,
    quantity INT NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(id),
    FOREIGN KEY (product_id) REFERENCES products(id)
);
```

-- Insert Sample Data into Order Items

```
INSERT INTO order_items (order_id, product_id, quantity) VALUES
(1, 1, 1),
```

(2, 11, 1),
(3, 9, 1),
(4, 8, 1),
(5, 5, 1),
(6, 12, 1),
(7, 3, 1),
(8, 13, 1),
(9, 6, 1),
(10, 10, 1),
(11, 7, 1),
(12, 15, 1),
(13, 14, 1),
(14, 4, 1),
(15, 2, 1),
(16, 3, 1),
(17, 13, 1),
(18, 6, 1),
(19, 10, 1),
(20, 7, 1),
(21, 15, 1),
(22, 11, 1),
(23, 9, 1),
(24, 8, 1);

SELECT * FROM order_items;

id	order_id	product_id	quantity
1	1	1	1
2	2	11	1
3	3	9	1
4	4	8	1
5	5	5	1
6	6	12	1
7	7	3	1
8	8	13	1
9	9	6	1
10	10	10	1
11	11	7	1
12	12	15	1
13	13	14	1
14	14	4	1
15	15	2	1
16	16	3	1
17	17	13	1
18	18	6	1
19	19	10	1
20	20	7	1
21	21	15	1
22	22	11	1
23	23	9	1
24	24	8	1

-- Query 10: Average total of all orders

```
SELECT ROUND(AVG(total_amount), 2) AS order_total_avg
FROM orders;
```

+-----+
order_total_avg
+-----+
50661.54
+-----+