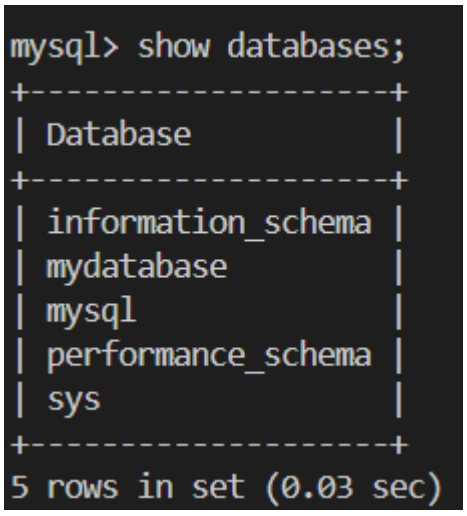
SQL TASK

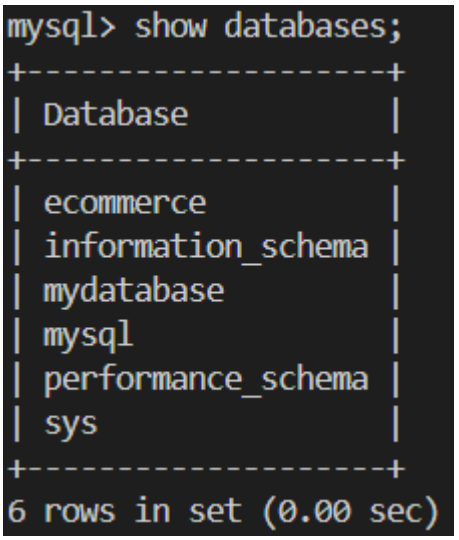
CREATING A DATABASE named as 'ecommerce' :

```
CREATE DATABASE ecommerce;
```

BEFORE :



AFTER :



CREATING TABLES (customers, orders, products) :

```
USE ecommerce;
```

CUSTOMERS TABLE :

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

```
mysql> desc customers;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| id      | int          | NO   | PRI | NULL    | auto_increment |
| name    | varchar(255) | NO   |     | NULL    |                |
| email   | varchar(255) | NO   | UNI | NULL    |                |
| address | text         | YES  |     | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

ORDERS TABLE :

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

```
mysql> desc orders;
+--------------+---------------+------+-----+---------+----------------+
| Field        | Type          | Null | Key | Default | Extra          |
+--------------+---------------+------+-----+---------+----------------+
| id           | int           | NO   | PRI | NULL    | auto_increment |
| customer_id  | int           | NO   | MUL | NULL    |                |
| order_date   | date          | NO   |     | NULL    |                |
| total_amount | decimal(10,2) | NO   |     | NULL    |                |
+--------------+---------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

PRODUCTS TABLE :

```
CREATE TABLE products (
    id INT AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    description VARCHAR(255),
    PRIMARY KEY(id)
);
```

```
mysql> desc products;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| id          | int           | NO   | PRI | NULL    | auto_increment |
| name        | varchar(255)  | NO   |     | NULL    |                |
| price       | decimal(10,2) | NO   |     | NULL    |                |
| description | text          | YES  |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

INSERTING SAMPLE DATA INTO TABLES :

CUSTOMERS TABLE :

```
-- Insert sample data into customers table
INSERT INTO customers (name, email, address) VALUES
('John Doe', 'john.doe@example.com', '123 Elm Street'),
('Jane Smith', 'jane.smith@example.com', '456 Oak Avenue'),
('Sam Wilson', 'sam.wilson@example.com', '789 Pine Road'),
('Anna Brown', 'anna.brown@example.com', '321 Maple Lane'),
('Mike Davis', 'mike.davis@example.com', '654 Cedar Court');
```

```
mysql> select * from customers;
+----+------------+-----------------------+-----------------+
| id | name       | email                 | address         |
+----+------------+-----------------------+-----------------+
|  1 | John Doe   | john.doe@example.com  | 123 Elm Street  |
|  2 | Jane Smith | jane.smith@example.com | 456 Oak Avenue  |
|  3 | Sam Wilson | sam.wilson@example.com | 789 Pine Road   |
|  4 | Anna Brown | anna.brown@example.com | 321 Maple Lane  |
|  5 | Mike Davis | mike.davis@example.com | 654 Cedar Court |
+----+------------+-----------------------+-----------------+
5 rows in set (0.00 sec)
```

ORDERS TABLE :

```
-- Insert sample data into orders table
INSERT INTO orders (customer_id, order_date, total_amount) VA
(3, '2024-10-01', 35.00),
(5, '2024-10-05', 50.50),
(1, '2024-10-10', 69.00),
(2, '2024-10-15', 110.00),
(4, '2024-10-20', 42.00),
(3, '2024-10-25', 85.50), -- Product A + Product B
(1, '2024-10-30', 179.00), -- Product D + Product C
(5, '2024-11-03', 77.00), -- Product A + Product E
(4, '2024-11-07', 110.00), -- Product D
(2, '2024-11-12', 92.50), -- Product B + Product E
(5, '2024-11-15', 151.00), -- Product D + Product A
(3, '2024-11-18', 35.00),  -- Product A
(4, '2024-11-21', 50.50),  -- Product B
(5, '2024-11-25', 138.00), -- Product C + Product E + Product
(2, '2024-11-28', 220.00), -- Product D + Product D
(5, '2024-12-01', 42.00),  -- Product E
(3, '2024-12-05', 119.50), -- Product C + Product B
(4, '2024-12-10', 104.00), -- Product E + Product C
(2, '2024-12-15', 69.00),  -- Product C
(3, '2024-12-20', 85.50);  -- Product A + Product B
```

```
mysql> select * from orders;
+----+-------------+------------+--------------+
| id | customer_id | order_date | total_amount |
+----+-------------+------------+--------------+
|  1 |           3 | 2024-10-01 |        35.00 |
|  2 |           5 | 2024-10-05 |        50.50 |
|  3 |           1 | 2024-10-10 |        69.00 |
|  4 |           2 | 2024-10-15 |       110.00 |
|  5 |           4 | 2024-10-20 |        42.00 |
|  6 |           3 | 2024-10-25 |        85.50 |
|  7 |           1 | 2024-10-30 |       179.00 |
|  8 |           5 | 2024-11-03 |        77.00 |
|  9 |           4 | 2024-11-07 |       110.00 |
| 10 |           2 | 2024-11-12 |        92.50 |
| 11 |           5 | 2024-11-15 |       151.00 |
| 12 |           3 | 2024-11-18 |        35.00 |
| 13 |           4 | 2024-11-21 |        50.50 |
| 14 |           5 | 2024-11-25 |       138.00 |
| 15 |           2 | 2024-11-28 |       220.00 |
| 16 |           5 | 2024-12-01 |        42.00 |
| 17 |           3 | 2024-12-05 |       119.50 |
| 18 |           4 | 2024-12-10 |       104.00 |
| 19 |           2 | 2024-12-15 |        69.00 |
| 20 |           3 | 2024-12-20 |        85.50 |
+----+-------------+------------+--------------+
20 rows in set (0.03 sec)
```

PRODUCTS TABLE :

```
-- Insert sample data into products table
INSERT INTO products (name, price, description) VALUES
('Product A', 35.00, 'This is description of Product A'),
('Product B', 50.50, 'This is description of Product B'),
('Product C', 69.00, 'This is description of Product C'),
('Product D', 110.00, 'This is description of Product D'),
('Product E', 42.00, 'This is description of Product E');
```

```
mysql> select * from products;
+----+-----------+--------+-----------------------------------+
| id | name      | price  | description                       |
+----+-----------+--------+-----------------------------------+
|  1 | Product A |  35.00 | This is description of Product A |
|  2 | Product B |  50.50 | This is description of Product B |
|  3 | Product C |  69.00 | This is description of Product C |
|  4 | Product D | 110.00 | This is description of Product D |
|  5 | Product E |  42.00 | This is description of Product E |
+----+-----------+--------+-----------------------------------+
5 rows in set (0.00 sec)
```

QUERIES :

1. Retrieve all customers who have placed an order in the last 30 days :

```
SELECT DISTINCT c.name, c.email, c.address
FROM customers AS c
JOIN orders AS o ON c.id = o.customer_id
WHERE o.order_date BETWEEN CURDATE() - INTERVAL 30 DAY AND CU
```

```
mysql> SELECT DISTINCT c.name, c.email, c.address
    -> FROM customers AS c
    -> JOIN orders AS o ON c.id = o.customer_id
    -> WHERE o.order_date BETWEEN CURDATE() - INTERVAL 30 DAY AND CURDATE();
+------------+----------------------+----------------+
| name       | email                | address        |
+------------+----------------------+----------------+
| Jane Smith | jane.smith@example.com | 456 Oak Avenue  |
| Sam Wilson | sam.wilson@example.com | 789 Pine Road   |
| Anna Brown | anna.brown@example.com | 321 Maple Lane  |
| Mike Davis | mike.davis@example.com | 654 Cedar Court |
+------------+----------------------+----------------+
4 rows in set (0.00 sec)
```

2. Get the total amount of all orders placed by each customer :

```
SELECT c.name AS CUSTOMER_NAME, sum(o.total_amount) AS TOTAL_
FROM customers AS c
```

```
JOIN orders AS o ON c.id=o.customer_id
GROUP BY c.name;
```

```
mysql> SELECT c.name AS CUSTOMER_NAME, sum(o.total_amount) AS TOTAL_AMOUNT
    -> FROM customers AS c
    -> JOIN orders AS o ON c.id=o.customer_id
    -> GROUP BY c.name;
+---------------+--------------+
| CUSTOMER_NAME | TOTAL_AMOUNT |
+---------------+--------------+
| John Doe      |       248.00 |
| Jane Smith    |       491.50 |
| Sam Wilson    |       360.50 |
| Anna Brown    |       306.50 |
| Mike Davis    |       458.50 |
+---------------+--------------+
5 rows in set (0.01 sec)
```

3. Update the price of Product C to 45.00 :

```
UPDATE products
SET price=45.00
WHERE id=3;
```

```
mysql> select * from products;
+----+-----------+--------+-----------------------------------+
| id | name      | price  | description                       |
+----+-----------+--------+-----------------------------------+
|  1 | Product A |  35.00 | This is description of Product A |
|  2 | Product B |  50.50 | This is description of Product B |
|  3 | Product C |  45.00 | This is description of Product C |
|  4 | Product D | 110.00 | This is description of Product D |
|  5 | Product E |  42.00 | This is description of Product E |
+----+-----------+--------+-----------------------------------+
5 rows in set (0.00 sec)
```

4. Add a new column discount to the products table :

```
ALTER TABLE products
ADD discount DECIMAL(5,2) DEFAULT 0;
```

```
mysql> select * from products;
+----+-----------+--------+-----------------------------------+----------+
| id | name      | price  | description                       | discount |
+----+-----------+--------+-----------------------------------+----------+
|  1 | Product A |  35.00 | This is description of Product A |     0.00 |
|  2 | Product B |  50.50 | This is description of Product B |     0.00 |
|  3 | Product C |  45.00 | This is description of Product C |     0.00 |
|  4 | Product D | 110.00 | This is description of Product D |     0.00 |
|  5 | Product E |  42.00 | This is description of Product E |     0.00 |
+----+-----------+--------+-----------------------------------+----------+
5 rows in set (0.00 sec)
```

```
mysql> select * from products;
+----+-----------+--------+-----------------------------------+----------+
| id | name      | price  | description                       | discount |
+----+-----------+--------+-----------------------------------+----------+
|  1 | Product A |  35.00 | This is description of Product A |    10.00 |
|  2 | Product B |  50.50 | This is description of Product B |    15.00 |
|  3 | Product C |  45.00 | This is description of Product C |     8.00 |
|  4 | Product D | 110.00 | This is description of Product D |    18.00 |
|  5 | Product E |  42.00 | This is description of Product E |    12.00 |
+----+-----------+--------+-----------------------------------+----------+
5 rows in set (0.00 sec)
```

5. Retrieve the top 3 products with the highest price :

```
SELECT name, price, description
FROM products
ORDER BY price DESC
LIMIT 3;
```

```
mysql> SELECT name, price, description
    -> FROM products
    -> ORDER BY price DESC
    -> LIMIT 3;
+-----------+--------+---------------------------------+
| name      | price  | description                     |
+-----------+--------+---------------------------------+
| Product D | 110.00 | This is description of Product D |
| Product B |  50.50 | This is description of Product B |
| Product C |  45.00 | This is description of Product C |
+-----------+--------+---------------------------------+
3 rows in set (0.00 sec)
```

6. Normalize the database by creating a separate table for order items and updating the orders table to reference the 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)
);
```

```
mysql> desc order_items;
+------------+------+------+-----+---------+----------------+
| Field      | Type | Null | Key | Default | Extra          |
+------------+------+------+-----+---------+----------------+
| id         | int  | NO   | PRI | NULL    | auto_increment |
| order_id   | int  | NO   | MUL | NULL    |                |
| product_id | int  | NO   | MUL | NULL    |                |
| quantity   | int  | NO   |     | NULL    |                |
+------------+------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

```
mysql> select * from order_items;
+----+----------+------------+----------+
| id | order_id | product_id | quantity |
+----+----------+------------+----------+
|  1 |        1 |          1 |        1 |
|  2 |        2 |          2 |        1 |
|  3 |        3 |          3 |        1 |
|  4 |        4 |          4 |        1 |
|  5 |        5 |          5 |        1 |
|  6 |        6 |          1 |        1 |
|  7 |        6 |          2 |        1 |
|  8 |        7 |          4 |        1 |
|  9 |        7 |          3 |        1 |
| 10 |        8 |          3 |        1 |
| 11 |        8 |          5 |        1 |
| 12 |        9 |          4 |        1 |
| 13 |       10 |          2 |        1 |
| 14 |       10 |          5 |        1 |
| 15 |       11 |          4 |        1 |
| 16 |       11 |          1 |        1 |
| 17 |       12 |          1 |        1 |
| 18 |       13 |          2 |        1 |
| 19 |       14 |          3 |        1 |
| 20 |       14 |          5 |        1 |
| 21 |       14 |          1 |        1 |
| 22 |       15 |          4 |        2 |
| 23 |       16 |          5 |        1 |
| 24 |       17 |          4 |        1 |
| 25 |       17 |          2 |        1 |
| 26 |       18 |          5 |        1 |
| 27 |       18 |          3 |        1 |
| 28 |       19 |          3 |        1 |
| 29 |       20 |          1 |        1 |
| 30 |       20 |          2 |        1 |
+----+----------+------------+----------+
30 rows in set (0.00 sec)
```

7. Get the names of customers who have ordered Product A :

```
SELECT DISTINCT customers.name
FROM customers
JOIN orders ON customers.id = orders.customer_id
JOIN order_items ON orders.id = order_items.order_id
JOIN products ON order_items.product_id = products.id
WHERE products.name = 'Product A';
```

```
mysql> SELECT DISTINCT customers.name
    -> FROM customers
    -> JOIN orders ON customers.id = orders.customer_id
    -> JOIN order_items ON orders.id = order_items.order_id
    -> JOIN products ON order_items.product_id = products.id
    -> WHERE products.name = 'Product A';
+------------+
| name       |
+------------+
| Sam Wilson |
| Mike Davis |
+------------+
2 rows in set (0.00 sec)
```

8. Join the orders and customers tables to retrieve the customer's name and order date for each order :

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

```
mysql> SELECT customers.name, orders.order_date
    -> FROM customers
    -> JOIN orders ON customers.id = orders.customer_id
    -> ORDER BY orders.order_date ASC;
+------------+------------+
| name       | order_date |
+------------+------------+
| Sam Wilson | 2024-10-01 |
| Mike Davis | 2024-10-05 |
| John Doe   | 2024-10-10 |
| Jane Smith | 2024-10-15 |
| Anna Brown | 2024-10-20 |
| Sam Wilson | 2024-10-25 |
| John Doe   | 2024-10-30 |
| Mike Davis | 2024-11-03 |
| Anna Brown | 2024-11-07 |
| Jane Smith | 2024-11-12 |
| Mike Davis | 2024-11-15 |
| Sam Wilson | 2024-11-18 |
| Anna Brown | 2024-11-21 |
| Mike Davis | 2024-11-25 |
| Jane Smith | 2024-11-28 |
| Mike Davis | 2024-12-01 |
| Sam Wilson | 2024-12-05 |
| Anna Brown | 2024-12-10 |
| Jane Smith | 2024-12-15 |
| Sam Wilson | 2024-12-20 |
+------------+------------+
20 rows in set (0.00 sec)
```

9. Retrieve the orders with a total amount greater than 150.00 :

```
SELECT * FROM orders
WHERE total_amount>150.00;
```

```
mysql> SELECT * FROM orders
    -> WHERE total_amount>150.00;
+----+-------------+------------+--------------+
| id | customer_id | order_date | total_amount |
+----+-------------+------------+--------------+
|  7 |           1 | 2024-10-30 |       179.00 |
| 11 |           5 | 2024-11-15 |       151.00 |
| 15 |           2 | 2024-11-28 |       220.00 |
+----+-------------+------------+--------------+
3 rows in set (0.00 sec)
```

10. Retrieve the average total of all orders :

```
SELECT AVG(total_amount) AS average_order_total
FROM (
    SELECT order_id,
    SUM(products.price * order_items.quantity) AS total_amoun
    FROM order_items
    JOIN products ON order_items.product_id = products.id
    GROUP BY order_id
) AS order_totals;
```

```
mysql> SELECT AVG(total_amount) AS average_order_total
    -> FROM (
    ->     SELECT order_id,
    ->     SUM(products.price * order_items.quantity) AS total_amount
    ->     FROM order_items
    ->     JOIN products ON order_items.product_id = products.id
    ->     GROUP BY order_id
    -> ) AS order_totals;
+--------------------+
| average_order_total |
+--------------------+
|          90.250000 |
+--------------------+
1 row in set (0.03 sec)
```