TITLE: Ecommerce Data Analysis using SQL.

AIM: To perform data analysis using SQL on an e-commerce dataset by creating a database and tables, inserting sample data, and applying various SQL queries such as SELECT, WHERE, ORDER BY, GROUP BY, JOINS, subqueries, aggregate functions, views, and indexes to extract meaningful insights.

PROCEDURE:

1. Creating a Database.

```
mysql> CREATE DATABASE ecom;
Query OK, 1 row affected (0.01 sec)

mysql> use ecom;
Database changed
```

2. Create the ecommerce schema and insert data

```
mysql> CREATE TABLE customers (
    ->     customer_id INT PRIMARY KEY,
    ->     name VARCHAR(50),
    ->     city VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE TABLE products (
    ->     product_id INT PRIMARY KEY,
    ->     name VARCHAR(50),
    ->     category VARCHAR(50),
    ->     price DECIMAL(10,2)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE orders (
    ->     order_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     order_date DATE,
    ->     amount DECIMAL(10,2),
    ->     FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE order_items (
    ->     order_item_id INT PRIMARY KEY,
    ->     order_id INT,
    ->     product_id INT,
    ->     quantity INT,
    ->     FOREIGN KEY (order_id) REFERENCES orders(order_id),
    ->     FOREIGN KEY (product_id) REFERENCES products(product_id)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> INSERT INTO customers VALUES
    -> (1, 'Alice', 'New York'),
    -> (2, 'Bob', 'Los Angeles'),
    -> (3, 'Charlie', 'Chicago');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM customers;
+-------------+---------+-------------+
| customer_id | name    | city        |
+-------------+---------+-------------+
|           1 | Alice   | New York    |
|           2 | Bob     | Los Angeles |
|           3 | Charlie | Chicago     |
+-------------+---------+-------------+
3 rows in set (0.00 sec)
```

```
mysql> INSERT INTO products VALUES
    -> (1, 'Laptop', 'Electronics', 1000.00),
    -> (2, 'Phone', 'Electronics', 500.00),
    -> (3, 'Desk Chair', 'Furniture', 150.00);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM products;
+------------+-----------+-------------+---------+
| product_id | name      | category    | price   |
+------------+-----------+-------------+---------+
|          1 | Laptop    | Electronics | 1000.00 |
|          2 | Phone     | Electronics |  500.00 |
|          3 | Desk Chair | Furniture  |  150.00 |
+------------+-----------+-------------+---------+
```

```
mysql> INSERT INTO orders VALUES
    -> (1, 1, '2024-05-01', 1200.00),
    -> (2, 2, '2024-05-02', 500.00),
    -> (3, 1, '2024-05-03', 150.00);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM orders;
+----------+-------------+------------+---------+
| order_id | customer_id | order_date | amount  |
+----------+-------------+------------+---------+
|        1 |           1 | 2024-05-01 | 1200.00 |
|        2 |           2 | 2024-05-02 |  500.00 |
|        3 |           1 | 2024-05-03 |  150.00 |
+----------+-------------+------------+---------+
3 rows in set (0.00 sec)
```

```
mysql> INSERT INTO order_items VALUES
    -> (1, 1, 1, 1),
    -> (2, 1, 3, 1),
    -> (3, 2, 2, 1),
    -> (4, 3, 3, 1);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM order_items;
+---------------+----------+------------+----------+
| order_item_id | order_id | product_id | quantity |
+---------------+----------+------------+----------+
|             1 |        1 |          1 |        1 |
|             2 |        1 |          3 |        1 |
|             3 |        2 |          2 |        1 |
|             4 |        3 |          3 |        1 |
+---------------+----------+------------+----------+
4 rows in set (0.00 sec)
```

3. Sample queries for the task
   - Select customers from New York ordered by name

```
mysql> SELECT * FROM customers WHERE city = 'New York' ORDER BY name;
+-------------+-------+----------+
| customer_id | name  | city     |
+-------------+-------+----------+
|           1 | Alice | New York |
+-------------+-------+----------+
1 row in set (0.00 sec)
```

   - Sum of amounts per customer

```
mysql> SELECT customer_id, SUM(amount) AS total_spent
    -> FROM orders
    -> GROUP BY customer_id;
+-------------+-------------+
| customer_id | total_spent |
+-------------+-------------+
|           1 |     1350.00 |
|           2 |      500.00 |
+-------------+-------------+
2 rows in set (0.00 sec)
```

   - Join customers with orders

```
mysql> SELECT c.name, o.order_id, o.amount
    -> FROM customers c
    -> INNER JOIN orders o ON c.customer_id = o.customer_id;
+-------+----------+---------+
| name  | order_id | amount  |
+-------+----------+---------+
| Alice |        1 | 1200.00 |
| Alice |        3 |  150.00 |
| Bob   |        2 |  500.00 |
+-------+----------+---------+
3 rows in set (0.00 sec)
```

   - Subquery: Customers who spent more than $1000

```
mysql> SELECT * FROM customers
    -> WHERE customer_id IN (
    ->   SELECT customer_id FROM orders WHERE amount > 1000
    -> );
+-------------+-------+----------+
| customer_id | name  | city     |
+-------------+-------+----------+
|           1 | Alice | New York |
+-------------+-------+----------+
1 row in set (0.00 sec)
```

   - Create a view with customer order summary

```
mysql> CREATE VIEW customer_order_summary AS
    -> SELECT c.name, COUNT(o.order_id) AS total_orders, SUM(o.amount) AS total_spent
    -> FROM customers c
    -> JOIN orders o ON c.customer_id = o.customer_id
    -> GROUP BY c.customer_id;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SELECT * FROM customer_order_summary;
+-------+--------------+-------------+
| name  | total_orders | total_spent |
+-------+--------------+-------------+
| Alice |            2 |     1350.00 |
| Bob   |            1 |      500.00 |
+-------+--------------+-------------+
2 rows in set (0.02 sec)
```

- Create index for optimization

```
mysql> CREATE INDEX idx_customer_id ON orders(customer_id);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> SHOW INDEX FROM orders;
+--------+------------+----------------+--------------+-------------+-----------+-------------+----------+--------+------+------------
--+---------+---------------+---------+------------+
| Table  | Non_unique | Key_name       | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_typ
e | Comment | Index_comment | Visible | Expression |
+--------+------------+----------------+--------------+-------------+-----------+-------------+----------+--------+------+------------
--+---------+---------------+---------+------------+
| orders |          0 | PRIMARY        |            1 | order_id    | A         |           3 |     NULL |   NULL |      | BTREE
  |         |               | YES     | NULL       |
| orders |          1 | idx_customer_id |            1 | customer_id | A         |           2 |     NULL |   NULL | YES  | BTREE
  |         |               | YES     | NULL       |
+--------+------------+----------------+--------------+-------------+-----------+-------------+----------+--------+------+------------
--+---------+---------------+---------+------------+
2 rows in set (0.06 sec)
```

CONCLUSION:

We now have a complete ecommerce database and analysis queries to fulfill the requirements:

- Created tables and inserted data
- Used SELECT, WHERE, ORDER BY, GROUP BY
- Used JOINs and subqueries
- Created views
- Created index for optimization