E commerce

Customer table

create table Customer (customer_id INT Primary key,name varchar(255),email varchar(255),password varchar(255));

```
mysql> create table customer(customer_id int primary key,first_name varchar(255),last_name varchar(255),email varchar(255),password varchar(255));
Query OK, 0 rows affected (0.04 sec)

mysql> create table customers(customer_id int primary key,first_name varchar(255),last_name varchar(255),email varchar(255),address varchar(255));
Query OK, 0 rows affected (0.33 sec)

mysql> insert into customers values(1,'john','doe','johndoe@example.com','123 Main St,city');
Query OK, 1 row affected (0.01 sec)

mysql> insert into customers values(2,'jane','smith','janesmith@example.com','456 Eim St,Town');
Query OK, 1 row affected (0.01 sec)

mysql> insert into customers values(3,'robert','johnson','robert@example.com','789 Oak St,Village');
Query OK, 1 row affected (0.01 sec)

mysql> insert into customers values(4,'sarah','brown','sarah@example.com','101 Pine St,Suburb');
Query OK, 1 row affected (0.00 sec)

mysql> insert into customers values(5,'david','lee','david@example.com','234 Cedar St,District');
Query OK, 1 row affected (0.31 sec)

mysql> insert into customers values(6,'laura','hall','laura@example.com','567 Birch St,Country');
Query OK, 1 row affected (0.02 sec)

mysql> insert into customers values(7,'michael','davis','michael@example.com','890 Maple St,State');
Query OK, 1 row affected (0.01 sec)

mysql> insert into customers values(8,'emma','wilson','emma@example.com','321 Redwood St,Country');
Query OK, 1 row affected (0.02 sec)
```

```
mysql> insert into customers values(9,'william','taylor','william@example.com','432 Spruce St,Province');
Query OK, 1 row affected (0.00 sec)

mysql> insert into customers values(10,'olivia','adams','olivia@example.com','765 Fir St,Territory');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from customers;
+-------------+------------+-----------+----------------------+------------------------+
| customer_id | first_name | last_name | email                | address                |
+-------------+------------+-----------+----------------------+------------------------+
|           1 | john       | doe       | johndoe@example.com  | 123 Main St,city       |
|           2 | jane       | smith     | janesmith@example.com | 456 Eim St,Town        |
|           3 | robert     | johnson   | robert@example.com   | 789 Oak St,Village     |
|           4 | sarah      | brown     | sarah@example.com    | 101 Pine St,Suburb     |
|           5 | david      | lee       | david@example.com    | 234 Cedar St,District  |
|           6 | laura      | hall      | laura@example.com    | 567 Birch St,Country   |
|           7 | michael    | davis     | michael@example.com  | 890 Maple St,State     |
|           8 | emma       | wilson    | emma@example.com     | 321 Redwood St,Country |
|           9 | william    | taylor    | william@example.com  | 432 Spruce St,Province |
|          10 | olivia     | adams     | olivia@example.com   | 765 Fir St,Territory   |
+-------------+------------+-----------+----------------------+------------------------+
10 rows in set (0.02 sec)
```

Product table

create table products(product_id int primary key,name varchar(255),price decimal,decription text,stockQuantity int);

```
mysql> create table products(product_id int primary key,name varchar(255),price decimal,decription text,stockQuantity int);
Query OK, 0 rows affected (0.17 sec)

mysql> insert into products values(1,'laptop',800,'high-performance laptop',10);
Query OK, 1 row affected (0.11 sec)

mysql> insert into products values(2,'smartphone',600,'latest smartphone',15);
Query OK, 1 row affected (0.02 sec)

mysql> insert into products values(3,'tablet',300,'portable',20);
Query OK, 1 row affected (0.02 sec)

mysql> insert into products values(4,'headphones',150,'noise-canceling',30);
Query OK, 1 row affected (0.02 sec)

mysql> insert into products values(5,'TV',900,'4K smart tv',5);
Query OK, 1 row affected (0.01 sec)

mysql> insert into products values(6,'coffee maker',50,'automatic coffee maker',25);
Query OK, 1 row affected (0.01 sec)

mysql> insert into products values(7,'refrigerator',700,'energy-efficient',10);
Query OK, 1 row affected (0.11 sec)

mysql> insert into products values(8,'microwave oven',80,'counter top microwave',15);
Query OK, 1 row affected (0.00 sec)

mysql> insert into products values(9,'blender',70,'high speed blender',20);
Query OK, 1 row affected (0.01 sec)

mysql> insert into products values(10,'vaccum cleaner',120,'bagless vaccum cleaner',10);
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from products;
+------------+----------------+-------+--------------------------+---------------+
| product_id | name           | price | decription               | stockQuantity |
+------------+----------------+-------+--------------------------+---------------+
|          1 | laptop         |   800 | high-performance laptop  |            10 |
|          2 | smartphone     |   600 | latest smartphone        |            15 |
|          3 | tablet         |   300 | portable                 |            20 |
|          4 | headphones     |   150 | noise-canceling          |            30 |
|          5 | TV             |   900 | 4K smart tv              |             5 |
|          6 | coffee maker   |    50 | automatic coffee maker   |            25 |
|          7 | refrigerator   |   700 | energy-efficient         |            10 |
|          8 | microwave oven |    80 | counter top microwave    |            15 |
|          9 | blender        |    70 | high speed blender       |            20 |
|         10 | vaccum cleaner |   120 | bagless vaccum cleaner   |            10 |
+------------+----------------+-------+--------------------------+---------------+
10 rows in set (0.00 sec)
```

Carts table

create table carts(cart_id int primary key,customer_id int,product_id int,quantity int,foreign key(customer_id) REFERENCES customers(customer_id),foreign key(product_id) REFERENCES products(product_id));

```
mysql> create table carts(cart_id int primary key,customer_id int,product_id int,quantity int,foreign key(customer_id) REFERENCES customers(customer_id),foreign key(pro
duct_id) REFERENCES products(product_id));
Query OK, 0 rows affected (0.06 sec)

mysql> insert into carts values(1,1,1,2);
Query OK, 1 row affected (0.10 sec)

mysql> insert into carts values(2,1,3,1);
Query OK, 1 row affected (0.02 sec)

mysql> insert into carts values(3,2,2,3);
Query OK, 1 row affected (0.02 sec)

mysql> insert into carts values(4,3,4,4);
Query OK, 1 row affected (0.01 sec)

mysql> insert into carts values(5,3,5,2);
Query OK, 1 row affected (0.09 sec)

mysql> insert into carts values(6,4,6,1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into carts values(7,5,1,1);
Query OK, 1 row affected (0.10 sec)

mysql> insert into carts values(8,6,10,2);
Query OK, 1 row affected (0.02 sec)

mysql> insert into carts values(9,6,9,3);
Query OK, 1 row affected (0.09 sec)

mysql> insert into carts values(10,7,7,2);
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from carts;
+---------+-------------+------------+----------+
| cart_id | customer_id | product_id | quantity |
+---------+-------------+------------+----------+
|       1 |           1 |          1 |        2 |
|       2 |           1 |          3 |        1 |
|       3 |           2 |          2 |        3 |
|       4 |           3 |          4 |        4 |
|       5 |           3 |          5 |        2 |
|       6 |           4 |          6 |        1 |
|       7 |           5 |          1 |        1 |
|       8 |           6 |         10 |        2 |
|       9 |           6 |          9 |        3 |
|      10 |           7 |          7 |        2 |
+---------+-------------+------------+----------+
10 rows in set (0.00 sec)
```

Orders table

create table orders(order_id int primary key,customer_id int,order_date date,total_price decimal,shippimg_address text,foreign key(customer_id) REFERENCES customers(customer_id));

```
mysql> create table orders(order_id int primary key,customer_id int,order_date date,total_price decimal,shipping_address text,foreign key(customer_id) REFERENCES custom
ers(customer_id));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> select * from orders;
+----------+-------------+------------+-------------+------------------+
| order_id | customer_id | order_date | total_price | shippimg_address |
+----------+-------------+------------+-------------+------------------+
|        1 |           1 | 2023-01-05 |        1200 | new york         |
|        2 |           2 | 2023-02-10 |         900 | india            |
|        3 |           3 | 2023-03-15 |         300 | delhi            |
|        4 |           4 | 2023-04-20 |         150 | china            |
|        5 |           5 | 2023-05-25 |        1800 | pakistan         |
|        6 |           6 | 2023-06-30 |         400 | agra             |
|        7 |           7 | 2023-07-05 |         700 | jamu             |
|        8 |           8 | 2023-08-10 |         160 | india            |
|        9 |           9 | 2023-09-15 |         140 | kerala           |
|       10 |          10 | 2024-10-20 |        1400 | dubai            |
+----------+-------------+------------+-------------+------------------+
10 rows in set (0.00 sec)
```

Order items table

create table order_item(order_item_id int primary key,order_id int,product_id int,quantity int,item_amount decimal,foreign key(order_id) REFERENCES orders(order_id),foreign key(product_id) REFERENCES products(product_id));

```
mysql> create table order_item(order_item_id int primary key,order_id int,product_id int,quantity int,item_amount decimal,foreign key(order_id) REFERENCES orders(order_
id),foreign key(product_id) REFERENCES products(product_id));
Query OK, 0 rows affected (0.26 sec)

mysql> insert into order_item values(2,1,3,1,300);
Query OK, 1 row affected (0.02 sec)

mysql> insert into order_item values(3,2,2,3,1800);
Query OK, 1 row affected (0.01 sec)

mysql> insert into order_item values(4,3,5,2,1800);
Query OK, 1 row affected (0.01 sec)

mysql> insert into order_item values(5,4,4,4,600);
Query OK, 1 row affected (0.06 sec)

mysql> insert into order_item values(6,4,6,1,50);
Query OK, 1 row affected (0.19 sec)

mysql> insert into order_item values(7,5,1,1,800);
Query OK, 1 row affected (0.12 sec)

mysql> insert into order_item values(8,5,2,2,1200);
Query OK, 1 row affected (0.02 sec)

mysql> insert into order_item values(9,6,10,2,240);
Query OK, 1 row affected (0.01 sec)

mysql> insert into order_item values(10,6,9,3,210);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from order_item;
+---------------+----------+------------+----------+-------------+
| order_item_id | order_id | product_id | quantity | item_amount |
+---------------+----------+------------+----------+-------------+
|             2 |        1 |          3 |        1 |         300 |
|             3 |        2 |          2 |        3 |        1800 |
|             4 |        3 |          5 |        2 |        1800 |
|             5 |        4 |          4 |        4 |         600 |
|             6 |        4 |          6 |        1 |          50 |
|             7 |        5 |          1 |        1 |         800 |
|             8 |        5 |          2 |        2 |        1200 |
|             9 |        6 |         10 |        2 |         240 |
|            10 |        6 |          9 |        3 |         210 |
+---------------+----------+------------+----------+-------------+
9 rows in set (0.00 sec)
```

Queries

1.Update refrigerator product price to 800.

update products set price = 800 where name ='refrigerator';

```
mysql> select * from products;
+------------+----------------+-------+-------------------------+---------------+
| product_id | name           | price | decription              | stockQuantity |
+------------+----------------+-------+-------------------------+---------------+
|          1 | laptop         |   800 | high-performance laptop |            10 |
|          2 | smartphone     |   600 | latest smartphone       |            15 |
|          3 | tablet         |   300 | portable                |            20 |
|          4 | headphones     |   150 | noise-canceling         |            30 |
|          5 | TV             |   900 | 4K smart tv             |             5 |
|          6 | coffee maker   |    50 | automatic coffee maker  |            25 |
|          7 | refrigerator   |   700 | energy-efficient        |            10 |
|          8 | microwave oven |    80 | counter top microwave   |            15 |
|          9 | blender        |    70 | high speed blender      |            20 |
|         10 | vaccum cleaner |   120 | bagless vaccum cleaner  |            10 |
+------------+----------------+-------+-------------------------+---------------+
10 rows in set (0.00 sec)

mysql> update products set price = 800 where name ='refrigerator';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from products;
+------------+----------------+-------+-------------------------+---------------+
| product_id | name           | price | decription              | stockQuantity |
+------------+----------------+-------+-------------------------+---------------+
|          1 | laptop         |   800 | high-performance laptop |            10 |
|          2 | smartphone     |   600 | latest smartphone       |            15 |
|          3 | tablet         |   300 | portable                |            20 |
|          4 | headphones     |   150 | noise-canceling         |            30 |
|          5 | TV             |   900 | 4K smart tv             |             5 |
|          6 | coffee maker   |    50 | automatic coffee maker  |            25 |
|          7 | refrigerator   |   800 | energy-efficient        |            10 |
|          8 | microwave oven |    80 | counter top microwave   |            15 |
|          9 | blender        |    70 | high speed blender      |            20 |
|         10 | vaccum cleaner |   120 | bagless vaccum cleaner  |            10 |
+------------+----------------+-------+-------------------------+---------------+
10 rows in set (0.00 sec)
```

2.Remove all cart items for a specific customer.

delete from carts where customer_id=1;

```
mysql> select * from carts;
+---------+-------------+------------+----------+
| cart_id | customer_id | product_id | quantity |
+---------+-------------+------------+----------+
|       1 |           1 |          1 |        2 |
|       2 |           1 |          3 |        1 |
|       3 |           2 |          2 |        3 |
|       4 |           3 |          4 |        4 |
|       5 |           3 |          5 |        2 |
|       6 |           4 |          6 |        1 |
|       7 |           5 |          1 |        1 |
|       8 |           6 |         10 |        2 |
|       9 |           6 |          9 |        3 |
|      10 |           7 |          7 |        2 |
+---------+-------------+------------+----------+
10 rows in set (0.00 sec)

mysql> delete from carts where customer_id=1;
Query OK, 2 rows affected (0.14 sec)

mysql> select * from carts
    -> select * from carts;
ERROR 1064 (42000): You have an error in your SQL synta
 carts' at line 2
mysql> select * from carts;
+---------+-------------+------------+----------+
| cart_id | customer_id | product_id | quantity |
+---------+-------------+------------+----------+
|       3 |           2 |          2 |        3 |
|       4 |           3 |          4 |        4 |
|       5 |           3 |          5 |        2 |
|       6 |           4 |          6 |        1 |
|       7 |           5 |          1 |        1 |
|       8 |           6 |         10 |        2 |
|       9 |           6 |          9 |        3 |
|      10 |           7 |          7 |        2 |
+---------+-------------+------------+----------+
8 rows in set (0.00 sec)
```

3.Retrieve Products Priced Below $100.

select * from products where price < 100;

```
mysql> select * from products;
+------------+----------------+-------+-------------------------+---------------+
| product_id | name           | price | decription              | stockQuantity |
+------------+----------------+-------+-------------------------+---------------+
|          1 | laptop         |   800 | high-performance laptop |            10 |
|          2 | smartphone     |   600 | latest smartphone       |            15 |
|          3 | tablet         |   300 | portable                |            20 |
|          4 | headphones     |   150 | noise-canceling         |            30 |
|          5 | TV             |   900 | 4K smart tv             |             5 |
|          6 | coffee maker   |    50 | automatic coffee maker  |            25 |
|          7 | refrigerator   |   800 | energy-efficient        |            10 |
|          8 | microwave oven |    80 | counter top microwave   |            15 |
|          9 | blender        |    70 | high speed blender      |            20 |
|         10 | vaccum cleaner |   120 | bagless vaccum cleaner  |            10 |
+------------+----------------+-------+-------------------------+---------------+
10 rows in set (0.00 sec)

mysql> select * from products where price < 100;
+------------+----------------+-------+-------------------------+---------------+
| product_id | name           | price | decription              | stockQuantity |
+------------+----------------+-------+-------------------------+---------------+
|          6 | coffee maker   |    50 | automatic coffee maker  |            25 |
|          8 | microwave oven |    80 | counter top microwave   |            15 |
|          9 | blender        |    70 | high speed blender      |            20 |
+------------+----------------+-------+-------------------------+---------------+
3 rows in set (0.00 sec)
```

4.Find Products with Stock Quantity Greater Than 5.

select * from products where stockQuantity > 5;

```
mysql> select * from products;
+------------+----------------+-------+-------------------------+---------------+
| product_id | name           | price | decription              | stockQuantity |
+------------+----------------+-------+-------------------------+---------------+
|          1 | laptop         |   800 | high-performance laptop |            10 |
|          2 | smartphone     |   600 | latest smartphone       |            15 |
|          3 | tablet         |   300 | portable                |            20 |
|          4 | headphones     |   150 | noise-canceling         |            30 |
|          5 | TV             |   900 | 4K smart tv             |             5 |
|          6 | coffee maker   |    50 | automatic coffee maker  |            25 |
|          7 | refrigerator   |   800 | energy-efficient        |            10 |
|          8 | microwave oven |    80 | counter top microwave   |            15 |
|          9 | blender        |    70 | high speed blender      |            20 |
|         10 | vaccum cleaner |   120 | bagless vaccum cleaner  |            10 |
+------------+----------------+-------+-------------------------+---------------+
10 rows in set (0.00 sec)

mysql> select * from products where stockQuantity > 5;
+------------+----------------+-------+-------------------------+---------------+
| product_id | name           | price | decription              | stockQuantity |
+------------+----------------+-------+-------------------------+---------------+
|          1 | laptop         |   800 | high-performance laptop |            10 |
|          2 | smartphone     |   600 | latest smartphone       |            15 |
|          3 | tablet         |   300 | portable                |            20 |
|          4 | headphones     |   150 | noise-canceling         |            30 |
|          6 | coffee maker   |    50 | automatic coffee maker  |            25 |
|          7 | refrigerator   |   800 | energy-efficient        |            10 |
|          8 | microwave oven |    80 | counter top microwave   |            15 |
|          9 | blender        |    70 | high speed blender      |            20 |
|         10 | vaccum cleaner |   120 | bagless vaccum cleaner  |            10 |
+------------+----------------+-------+-------------------------+---------------+
9 rows in set (0.00 sec)
```

5.Retrieve Orders with Total Amount Between $500 and $1000.

select * from orders where total_price between 500 and 1000;

```
mysql> select * from orders;
+----------+-------------+------------+-------------+------------------+
| order_id | customer_id | order_date | total_price | shippimg_address |
+----------+-------------+------------+-------------+------------------+
|        1 |           1 | 2023-01-05 |        1200 | new york         |
|        2 |           2 | 2023-02-10 |         900 | india            |
|        3 |           3 | 2023-03-15 |         300 | delhi            |
|        4 |           4 | 2023-04-20 |         150 | china            |
|        5 |           5 | 2023-05-25 |        1800 | pakistan         |
|        6 |           6 | 2023-06-30 |         400 | agra             |
|        7 |           7 | 2023-07-05 |         700 | jamu             |
|        8 |           8 | 2023-08-10 |         160 | india            |
|        9 |           9 | 2023-09-15 |         140 | kerala           |
|       10 |          10 | 2024-10-20 |        1400 | dubai            |
+----------+-------------+------------+-------------+------------------+
10 rows in set (0.01 sec)

mysql> select * from orders where total_price between 500 and 1000;
+----------+-------------+------------+-------------+------------------+
| order_id | customer_id | order_date | total_price | shippimg_address |
+----------+-------------+------------+-------------+------------------+
|        2 |           2 | 2023-02-10 |         900 | india            |
|        7 |           7 | 2023-07-05 |         700 | jamu             |
+----------+-------------+------------+-------------+------------------+
2 rows in set (0.02 sec)
```

6.Find Products which name end with letter ‘r’.

select * from products where name like '%r';

```
mysql> select * from products where name like '%r';
+------------+----------------+-------+------------------------+---------------+
| product_id | name           | price | decription             | stockQuantity |
+------------+----------------+-------+------------------------+---------------+
|          6 | coffee maker   |    50 | automatic coffee maker |            25 |
|          7 | refrigerator   |   800 | energy-efficient       |            10 |
|          9 | blender        |    70 | high speed blender     |            20 |
|         10 | vaccum cleaner |   120 | bagless vaccum cleaner |            10 |
+------------+----------------+-------+------------------------+---------------+
4 rows in set (0.02 sec)
```

7.Retrieve Cart Items for Customer 5

select * from carts where customer_id = 5;

```
mysql> select * from carts where customer_id = 5;
+---------+-------------+------------+----------+
| cart_id | customer_id | product_id | quantity |
+---------+-------------+------------+----------+
|       7 |           5 |          1 |        1 |
+---------+-------------+------------+----------+
1 row in set (0.01 sec)
```

8.Find Customers Who Placed Orders in 2023.

select DISTINCT c.* from customers c inner join orders o on c.customer_id = o.customer_id where year(order_date) = 2023;

```
mysql> select DISTINCT c.* from customers c inner join orders o on c.customer_id = o.customer_id where year(order_date) = 2023;
+-------------+------------+-----------+----------------------+-----------------------+
| customer_id | first_name | last_name | email                | address               |
+-------------+------------+-----------+----------------------+-----------------------+
|           1 | john       | doe       | johndoe@example.com  | 123 Main St,city      |
|           2 | jane       | smith     | janesmith@example.com | 456 Eim St,Town       |
|           3 | robert     | johnson   | robert@example.com   | 789 Oak St,Village    |
|           4 | sarah      | brown     | sarah@example.com    | 101 Pine St,Suburb    |
|           5 | david      | lee       | david@example.com    | 234 Cedar St,District |
|           6 | laura      | hall      | laura@example.com    | 567 Birch St,Country  |
|           7 | michael    | davis     | michael@example.com  | 890 Maple St,State    |
|           8 | emma       | wilson    | emma@example.com     | 321 Redwood St,Country |
|           9 | william    | taylor    | william@example.com  | 432 Spruce St,Province |
+-------------+------------+-----------+----------------------+-----------------------+
9 rows in set (0.03 sec)

mysql> select * from orders;
+----------+-------------+------------+-------------+-----------------+
| order_id | customer_id | order_date | total_price | shippimg_address |
+----------+-------------+------------+-------------+-----------------+
|        1 |           1 | 2023-01-05 |        1200 | new york        |
|        2 |           2 | 2023-02-10 |         900 | india           |
|        3 |           3 | 2023-03-15 |         300 | delhi           |
|        4 |           4 | 2023-04-20 |         150 | china           |
|        5 |           5 | 2023-05-25 |        1800 | pakistan        |
|        6 |           6 | 2023-06-30 |         400 | agra            |
|        7 |           7 | 2023-07-05 |         700 | jamu            |
|        8 |           8 | 2023-08-10 |         160 | india           |
|        9 |           9 | 2023-09-15 |         140 | kerala          |
|       10 |          10 | 2024-10-20 |        1400 | dubai           |
+----------+-------------+------------+-------------+-----------------+
10 rows in set (0.00 sec)
```

9.Determine the Minimum Stock Quantity for Each Product Category.

Select min(stockQuantity) as min_stock from products;

```
mysql> select min(stockQuantity) as min_stock from products;
+-----------+
| min_stock |
+-----------+
|         5 |
+-----------+
1 row in set (0.00 sec)
```

10.Calculate the Total Amount Spent by Each Customer.

select customer_id,sum(total_price) as total_spent from orders GROUP BY customer_id;

```
mysql> select customer_id,sum(total_price) as total_spent from orders GROUP BY customer_id;
+-------------+-------------+
| customer_id | total_spent |
+-------------+-------------+
|           1 |        1200 |
|           2 |         900 |
|           3 |         300 |
|           4 |         150 |
|           5 |        1800 |
|           6 |         400 |
|           7 |         700 |
|           8 |         160 |
|           9 |         140 |
|          10 |        1400 |
+-------------+-------------+
10 rows in set (0.02 sec)
```

11.Find the Average Order Amount for Each Customer.

select avg(total_price) as avg_order_amount from orders;

select customer_id,avg(total_price) as avg_order_amount from orders GROUP BY customer_id;

```
mysql> select avg(total_price) as avg_price from orders;
+-----------+
| avg_price |
+-----------+
|  715.0000 |
+-----------+
1 row in set (0.05 sec)
```

12.Count the Number of Orders Placed by Each Customer.

select customer_id,count(order_id) from orders GROUP BY customer_id;

```
mysql>  select customer_id,count(order_id) from orders GROUP BY customer_id;
+-------------+-----------------+
| customer_id | count(order_id) |
+-------------+-----------------+
|           1 |               1 |
|           2 |               1 |
|           3 |               1 |
|           4 |               1 |
|           5 |               1 |
|           6 |               1 |
|           7 |               1 |
|           8 |               1 |
|           9 |               1 |
|          10 |               1 |
+-------------+-----------------+
10 rows in set (0.00 sec)
```

13.Find the Maximum Order Amount for Each Customer.

```
mysql> select customer_id,max(total_price) from orders GROUP BY customer_id;
+-------------+------------------+
| customer_id | max(total_price) |
+-------------+------------------+
|           1 |             1200 |
|           2 |              900 |
|           3 |              300 |
|           4 |              150 |
|           5 |             1800 |
|           6 |              400 |
|           7 |              700 |
|           8 |              160 |
|           9 |              140 |
|          10 |             1400 |
+-------------+------------------+
10 rows in set (0.02 sec)
```

14.Get Customers Who Placed Orders Totaling Over $1000.

select c.* from customers c inner join(select customer_id from orders GROUP BY customer_id HAVING sum(total_price) > 1000) o on c.customer_id = o.customer_id;

```
mysql> select c.* from customers c inner join(select customer_id from orders GROUP BY customer_id HAVING sum(total_price) > 1000) o on c.customer_id = o.customer_id;
+-------------+------------+-----------+-------------------+--------------------+
| customer_id | first_name | last_name | email             | address            |
+-------------+------------+-----------+-------------------+--------------------+
|           1 | john       | doe       | johndoe@example.com | 123 Main St,city     |
|           5 | david      | lee       | david@example.com   | 234 Cedar St,District |
|          10 | olivia     | adams     | olivia@example.com  | 765 Fir St,Territory |
+-------------+------------+-----------+-------------------+--------------------+
3 rows in set (0.00 sec)
```

15.Subquery to Find Products Not in the Cart.

select * from products where product_id NOT IN (select DISTINCT product_id from carts);

```
mysql> select * from products where product_id NOT IN (select DISTINCT product_id from carts);
+------------+----------------+-------+-----------------------+---------------+
| product_id | name           | price | decription            | stockQuantity |
+------------+----------------+-------+-----------------------+---------------+
|          3 | tablet         |   300 | portable              |            20 |
|          8 | microwave oven |    80 | counter top microwave |            15 |
+------------+----------------+-------+-----------------------+---------------+
2 rows in set (0.02 sec)
```

16.Subquery to Find Customers Who Haven't Placed Orders.

select * from customers where customer_id NOT IN (select DISTINCT customer_id from orders);

```
mysql> select * from customers where customer_id NOT IN (select DISTINCT customer_id from orders);
Empty set (0.00 sec)
```

17.Subquery to Calculate the Percentage of Total Revenue for a Product.

select (total_product_revenue / total_revenue)* 100 as percentage from (select sum(total_price) as total_product_revenue from orders) as product_revenue cross join (select sum(total_price) as total_revenue from orders) as total_revenue;

```
mysql> select (total_product_revenue / total_revenue)* 100 as percentage from (select sum(total_price) as total_product_
revenue from orders) as product_revenue cross join (select sum(total_price) as total_revenue from orders) as total_reven
ue;
+------------+
| percentage |
+------------+
|   100.0000 |
+------------+
1 row in set (0.01 sec)
```

18.**Subquery to Find Products with Low Stock.**

select * from products where stockQuantity < (select avg(stockQuantity) from products);

```
mysql> select * from products where stockQuantity < (select avg(stockQuantity) from products);
+------------+---------------+-------+-------------------------+---------------+
| product_id | name          | price | decription              | stockQuantity |
+------------+---------------+-------+-------------------------+---------------+
|          1 | laptop        |   800 | high-performance laptop |            10 |
|          2 | smartphone    |   600 | latest smartphone       |            15 |
|          5 | TV            |   900 | 4K smart tv             |             5 |
|          7 | refrigerator  |   800 | energy-efficient        |            10 |
|          8 | microwave oven|    80 | counter top microwave   |            15 |
|         10 | vaccum cleaner|   120 | bagless vaccum cleaner  |            10 |
+------------+---------------+-------+-------------------------+---------------+
6 rows in set (0.00 sec)
```

19.Subquery to Find Customers Who Placed High-Value Orders.

select * from customers where customer_id in (select customer_id from orders GROUP BY customer_id HAVING sum(total_price) > 1000);

```
mysql> select * from customers where customer_id in (select customer_id from orders GROUP BY customer_id HAVING sum(total_price) > 1000);
+-------------+------------+-----------+--------------------+---------------------+
| customer_id | first_name | last_name | email              | address             |
+-------------+------------+-----------+--------------------+---------------------+
|           1 | john       | doe       | johndoe@example.com | 123 Main St,city    |
|           5 | david      | lee       | david@example.com  | 234 Cedar St,District |
|          10 | olivia     | adams     | olivia@example.com | 765 Fir St,Territory |
+-------------+------------+-----------+--------------------+---------------------+
3 rows in set (0.02 sec)
```