ECOMMERECE

create database ecommerce;

use ecommerce;

create table customer (

   customer_id int primary key,

  first_name varchar(255) not null,

      last_name varchar(255) not null,

   email varchar(255) unique not null,

      address varchar(255) not null

);

create table products(

      product_id int primary key,

  product_name varchar(255) not null,

      description varchar(255) not null,

  price decimal(10,2) not null,

  stockQuantity int

);

create table cart(

   cart_id int primary key,

      customer_id int,

      product_id int,

      constraint fk_customer foreign key (customer_id) references customer(customer_id),

      constraint fk_products foreign key (product_id) references products(product_id),

      quantity int

      );

create table orders(

      order_id int primary key,

      customer_id int,

      constraint fk_customer1 foreign key (customer_id) references customer(customer_id),

      order_date date not null,

      total_price decimal(10,2),

```
        );
create table order_items(

        order_item_id int primary key,

        order_id int,

        product_id int,

        constraint fk_orders foreign key (order_id) references orders(order_id),

        constraint fk_products1 foreign key (product_id) references products(product_id),

        quantity int,

        item_amount decimal(10,2) not null

);


insert into customer values

('1','John','Doe','johndoe@example.com','123 Main StCity') ,

('2','Jane','Smith','janesmith@example.com','456 Elm 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,County' ),

('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');

select * from customer;
```

	customer_id	first_name	last_name	email	address
1	1	John	Doe	johndoe@example.com	123 Main StCity
2	2	Jane	Smith	janesmith@example.com	456 Elm St,Town
3	3	Robert	Johnson	robert@example.com	789 Oak St,Village
4	4	Sarah	Brown	sarah@example.com	101 Pine St,Suburb
5	5	David	Lee	david@example.com	234 Cedar St,District
6	6	Laura	Hall	laura@example.com	567 Birch St,County
7	7	Michael	Davis	michael@example.com	890 Maple St,State
8	8	Emma	Wilson	emma@example.com	321 Redwood St,Country
9	9	William	Taylor	william@example.com	432 Spruce St, Province
10	10	Olivia	Adams	olivia@example.com	765 Fir St,Territory

insert into products values

('1','Laptop', 'High-performance laptop','800.0','10'),

('2','Smartphone','Latest smartphone ','600.00','15'),

('3','Tablet','Portable tablet','300.00','20'),

('4','Headphones','Noise-cancelling','150.00','30'),

('5','TV','4K Smart TV','900.00','5'),

('6','Coffee Maker','Automatic coffee maker','50.00','25'),

('7','Refrigerator','Energy-efficient ','700.00 ','10'),

('8','Microwave Oven','Countertop microwave','80.00 ','15'),

('9','Blender','High-speed blender','70.00 ','20'),

('10','Vacuum Cleaner','Bagless vacuum cleaner','120.00 ','10');

SELECT * FROM PRODUCTS;

	product_id	product_name	description	price	stockQuantity
1	1	Laptop	High-performance laptop	800.00	10
2	2	Smartphone	Latest smartphone	600.00	15
3	3	Tablet	Portable tablet	300.00	20
4	4	Headphones	Noise-cancelling	150.00	30
5	5	TV	4K Smart TV	900.00	5
6	6	Coffee Maker	Automatic coffee maker	50.00	25
7	7	Refrigerator	Energy-efficient	700.00	10
8	8	Microwave Oven	Countertop microwave	80.00	15
9	9	Blender	High-speed blender	70.00	20
10	10	Vacuum Cleaner	Bagless vacuum cleaner	120.00	10

insert into cart values

('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');

select * from cart;

	cart_id	customer_id	product_id	quantity
1	1	1	1	2
2	2	1	3	1
3	3	2	2	3
4	4	3	4	4
5	5	3	5	2
6	6	4	6	1
7	7	5	1	1
8	8	6	10	2
9	9	6	9	3
10	10	7	7	2

insert into orders values

('1', '1', '2023-01-05','1200.00'),

('2','2','2023-02-10','900.00'),

('3', '3', '2023-03-15','300.00'),

('4' ,'4', '2023-04-20','150.00 '),

('5', '5', '2023-05-25', '1800.00'),

('6', '6', '2023-06-30', '400.00'),

('7' ,'7', '2023-07-05', '700.00'),

('8', '8', '2023-08-10', '160.00'),

('9', '9', '2023-09-15','140.00'),

('10', '10', '2023-10-20', '1400.00');

select * from orders;

	order_id	customer_id	order_date	total_price
1	1	1	2023-01-05	1200.00
2	2	2	2023-02-10	900.00
3	3	3	2023-03-15	300.00
4	4	4	2023-04-20	150.00
5	5	5	2023-05-25	1800.00
6	6	6	2023-06-30	400.00
7	7	7	2023-07-05	700.00
8	8	8	2023-08-10	160.00
9	9	9	2023-09-15	140.00
10	10	10	2023-10-20	1400.00

insert into order_items values

('1', '1', '1', '2', '1600.00'),

('2', '1', '3', '1', '300.00'),

('3', '2', '2', '3', '1800.00'),

('4', '3', '5', '2', '1800.00'),

('5', '4', '4', '4', '600.00'),

('6', '4', '6', '1', '50.00'),

('7', '5', '1', '1', '800.00'),

('8', '5', '2', '2', '1200.00'),

('9', '6', '10', '2', '240.00'),

('10', '6', '9', '3', '210.00');

select * from order_items;

	order_item_id	order_id	product_id	quantity	item_amount
1	1	1	1	2	1600.00
2	2	1	3	1	300.00
3	3	2	2	3	1800.00
4	4	3	5	2	1800.00
5	5	4	4	4	600.00
6	6	4	6	1	50.00
7	7	5	1	1	800.00
8	8	5	2	2	1200.00
9	9	6	10	2	240.00
10	10	6	9	3	210.00

-- 1. update refrigerator product price to 800

update products set price = 800.00 where product_name like 'refrigerator';

7	7	Refrigerator	Energy-efficient	800.00	10

-- 2. remove all cart items for a specific customer

delete from cart where customer_id = 4;

	cart_id	customer_id	product_id	quantity
1	1	1	1	2
2	2	1	3	1
3	3	2	2	3
4	4	3	4	4
5	5	3	5	2
6	7	5	1	1
7	8	6	10	2
8	9	6	9	3
9	10	7	7	2

-- 3. retrieve products priced below $100

select * from products where price < 100.00;

	product_id	product_name	description	price	stockQuantity
1	6	Coffee Maker	Automatic coffee maker	50.00	25
2	8	Microwave Oven	Countertop microwave	80.00	15
3	9	Blender	High-speed blender	70.00	20

-- 4. find products with stock quantity greater than 5

select * from products where stockquantity > 5;

	product_id	product_name	description	price	stockQuantity
1	1	Laptop	High-performance laptop	800.00	10
2	2	Smartphone	Latest smartphone	600.00	15
3	3	Tablet	Portable tablet	300.00	20
4	4	Headphones	Noise-cancelling	150.00	30
5	6	Coffee Maker	Automatic coffee maker	50.00	25
6	7	Refrigerator	Energy-efficient	800.00	10
7	8	Microwave Oven	Countertop microwave	80.00	15
8	9	Blender	High-speed blender	70.00	20
9	10	Vacuum Cleaner	Bagless vacuum cleaner	120.00	10

-- 5. retrieve orders with total amount between $500 and $1000

select * from orders where total_price between 500 and 1000;

	order_id	customer_id	order_date	total_price
1	2	2	2023-02-10	900.00
2	7	7	2023-07-05	700.00

-- 6. find products which name ends with letter 'r'

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

	product_id	product_name	description	price	stockQuantity
1	6	Coffee Maker	Automatic coffee maker	50.00	25
2	7	Refrigerator	Energy-efficient	800.00	10
3	9	Blender	High-speed blender	70.00	20
4	10	Vacuum Cleaner	Bagless vacuum cleaner	120.00	10

-- 7. retrieve cart items for customer 5

select * from cart where customer_id = 5;

	cart_id	customer_id	product_id	quantity
1	7	5	1	1

-- 8. find customers who placed orders in 2023

select customer_id from orders where year(order_date)=2023;

	customer_id
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

-- 9. determine the minimum stock quantity for each product category

select min(stockquantity) as minimum_stock_quantity from products;

	minimum_stock_quantity
1	5

-- 10. calculate the total amount spent by each customer

select customer_id, sum(total_price) as total_amount_spent from orders group by customer_id;

	customer_id	total_amount_spent
1	1	1200.00
2	2	900.00
3	3	300.00
4	4	150.00
5	5	1800.00
6	6	400.00
7	7	700.00
8	8	160.00
9	9	140.00
10	10	1400.00

-- 11. find the average order amount for each customer

select customer_id, avg(total_price) as average_order_amount from orders group by customer_id;

	customer_id	average_order_amount
1	1	1200.000000
2	2	900.000000
3	3	300.000000
4	4	150.000000
5	5	1800.000000
6	6	400.000000
7	7	700.000000
8	8	160.000000
9	9	140.000000
10	10	1400.000000

-- 12. count the number of orders placed by each customer

select customer_id,count(order_id) as number_of_orders from orders group by customer_id;

	customer_id	number_of_orders
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1

-- 13. find the maximum order amount for each customer

select customer_id,max(total_price) as maximum_order_amount from orders group by customer_id;

	customer_id	maximum_order_amount
1	1	1200.00
2	2	900.00
3	3	300.00
4	4	150.00
5	5	1800.00
6	6	400.00
7	7	700.00
8	8	160.00
9	9	140.00
10	10	1400.00

-- 14. get customers who placed orders totaling over $1000

select customer_id from orders group by customer_id having sum(total_price)>1000;

	customer_id
1	1
2	5
3	10

-- 15. subquery to find products not in the cart

select * from products where product_id not in (select product_id from cart);

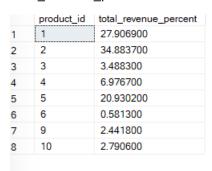|	product_id	product_name	description	price	stockQuantity
1	6	Coffee Maker	Automatic coffee maker	50.00	25
2	8	Microwave Oven	Countertop microwave	80.00	15

-- 16. subquery to find customers who haven't placed orders

select * from customer where customer_id not in (select customer_id from orders);

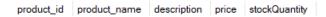| customer_id	first_name	last_name	email	address

-- 17. subquery to calculate the percentage of total revenue for a product

select product_id,(sum(item_amount)/(select sum(item_amount) from order_items) * 100) as total_revenue_percent from order_items group by product_id;

	product_id	total_revenue_percent
1	1	27.906900
2	2	34.883700
3	3	3.488300
4	4	6.976700
5	5	20.930200
6	6	0.581300
7	9	2.441800
8	10	2.790600

-- 18. subquery to find products with low stock

select * from products where stockquantity<5;

product_id	product_name	description	price	stockQuantity

-- 19. subquery to find customers who placed high-value orders

select customer_id from orders where total_price > 1500;

	customer_id
1	5