```
ELECTRONIC_GADGET_DB
use electronicgadget_db;
show databases;
show tables;
desc product;
/*insert into customer(first_name,email,address)values
('aravin','ara@gamil.com','chennai'),
('sara','sara@gamil.com','banglore'),
('praveen','pra@gamil.com','delhi'),
('paramesh','para@gamil.com','mumbai'),
('rameesh','ram@gamil.com','chennai');
insert into product(product_name,price)values
('mobile',25000),
('laptop',50000),
('tablet',60000),
('tv',100000),
('monitor',54263);
desc inventory;
insert into inventory(id,quantity_in_stock,last_stock_update,product_id)values
(1,'12','2024-02-19',2),
(2,'8','2024-03-01',1),
(3,'15','2023-10-04',4),
(4,'10','2023-01-09',3);
desc orders;
insert into orders(order_date,total_amount,quantity,product_id,customer_id)
values
('2024-01-10','120000',4,2,2),
('2024-01-10','120000',2,1,3),
('2024-11-11','220000',1,1,1),
('2023-05-19','200000',5,3,4);*/
```

/*task2*/

/*
1.Write an SQL query to retrieve the names and emails of all customers.
2. Write an SQL query to list all orders with their order dates and corresponding customer names.*/

```
select first_name,email from customer;

select * from orders;
```

/*3. Write an SQL query to insert a new customer record into the "Customers" table. Include customer information such as name, email, and address.
4. Write an SQL query to update the prices of all electronic gadgets in the "Products" table by increasing them by 10%.*/

```
insert into customer(first_name,email,address)values
('srivin','ari@gamil.com','banglore');
select * from product;

UPDATE product
SET price = price * 1.1
WHERE product_name= 'mobile';
```

/*5. Write an SQL query to delete a specific order and its associated order details from the "Orders" and "OrderDetails" tables. Allow users to input the order ID as a parameter.
6. Write an SQL query to insert a new order into the "Orders" table. Include the customer ID, order date, and any other necessary information.

7. Write an SQL query to update the contact information (e.g., email and address) of a specific customer in the "Customers" table. Allow users to input the customer ID and new contact information.*/

```
delete from orders where id =3;

insert into orders(order_date,total_amount,quantity,product_id,customer_id)
values ('2024-11-11','220000',1,1,1);

select * from customer;

update customer set email='arav.gmail.com' where id=1;
```

/*8. Write an SQL query to recalculate and update the total cost of each order in the "Orders" table based on the prices and quantities in the "OrderDetails" table.

9. Write an SQL query to delete all orders and their associated order details for a specific customer from the "Orders" and "OrderDetails" tables. Allow users to input the customer ID as a parameter.*/

```
select sum(price),id from product group by id;

delete from orders where customer_id=2;
```

/*10. Write an SQL query to insert a new electronic gadget product into the "Products" table, including product name, category, price, and any other relevant details.

11. Write an SQL query to update the status of a specific order in the "Orders" table (e.g., from "Pending" to "Shipped"). Allow users to input the order ID and the new status.

12. Write an SQL query to calculate and update the number of orders placed by each customer in the "Customers" table based on the data in the "Orders" table*/

```
insert into product(product_name,price)values
('keyboard',25000);

select * from orders;

select sum(id),id from orders group by customer_id;
/*

Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

1. Write an SQL query to retrieve a list of all orders along with customer information (e.g.,
customer name) for each order.
2. Write an SQL query to find the total revenue generated by each electronic gadget product.
Include the product name and the total revenue.*/

select * from customer;

SELECT p.product_name, SUM(o.total_amount) AS total_revenue
FROM orders o JOIN product p ON o.product_id = p.id
GROUP BY p.product_name;

/*3. Write an SQL query to list all customers who have made at least one purchase. Include their
names and contact information.
4. Write an SQL query to find the most popular electronic gadget, which is the one with the highest
total quantity ordered. Include the product name and the total quantity ordered.*/

select customer_id,quantity from orders group by customer_id;

select o.product_id,p.product_name from orders o JOIN product p ON o.product_id=p.id
 order by quantity desc limit 1;
```

/*5. Write an SQL query to retrieve a list of electronic gadgets along with their corresponding categories.
6. Write an SQL query to calculate the average order value for each customer. Include the customer's name and their average order value.*/

```
select * from product;

select avg(o.total_amount),p.id,c.first_name from orders o
JOIN product p ON o.product_id=p.id order by o.customer_id;
```

/*7. Write an SQL query to find the order with the highest total revenue. Include the order ID, customer information, and the total revenue.
8. Write an SQL query to list electronic gadgets and the number of times each product has been ordered.*/

```
 select product_id,total_amount from orders order by total_amount desc limit 1;

SELECT p.product_name, COUNT(o.id) AS order_count
FROM orders o
JOIN product p ON o.product_id = p.id
GROUP BY p.product_name;
```

/*9. Write an SQL query to find customers who have purchased a specific electronic gadget product. Allow users to input the product name as a parameter.
10. Write an SQL query to calculate the total revenue generated by all orders placed within a specific time period. Allow users to input the start and end dates as parameters.*/

```
select c.first_name ,p.product_name from customer c JOIN orders o ON o.customer_id=c.id JOIN product p
ON o.product_id=p.id having p.product_name='mobile';

select sum(total_amount),id from orders group by id;
```

/*Task 4. Subquery and its type:

1. Write an SQL query to find out which customers have not placed any orders.*/

```
select id from customer where NOT EXISTS (select customer_id from orders);
```

/*

2. Write an SQL query to find the total number of products available for sale.*/

```
select count(product_name) from product;
```

/*3. Write an SQL query to calculate the total revenue generated by TechShop.*/

```
select sum(total_amount) as revenue from orders;
```

/*4. Write an SQL query to calculate the average quantity ordered for products in a specific category. Allow users to input the category name as a parameter.*/

```
select avg(quantity) from orders group by product_id;
```

/*5. Write an SQL query to calculate the total revenue generated by a specific customer. Allow users to input the customer ID as a parameter.*/

```
select sum(total_amount),customer_id from orders where customer_id=3;
```

/*6. Write an SQL query to find the customers who have placed the most orders. List their names and the number of orders they've placed.*/

```
select c.first_name,o.id,count(o.id) from customer c JOIN orders o ON c.id=o.customer_id
 group by o.id order by count(o.id) desc limit 1;
```

/*7. Write an SQL query to find the most popular product category, which is the one with the highest total quantity ordered across all orders.*/

```
select product_id,quantity from orders order by quantity desc limit 1;
```

/*8. Write an SQL query to find the customer who has spent the most money (highest total revenue)

on electronic gadgets. List their name and total spending.

9. Write an SQL query to calculate the average order value (total revenue divided by the number of orders) for all customers.

*/

select c.first_name,sum(o.total_amount) as amount,o.quantity from customer c,product p,orders o where o.product_id=p.id and o.customer_id=c.id order by amount desc limit 1;

select sum(total_amount)/count(id)as avg from orders;

/*10. Write an SQL query to find the total number of orders placed by each customer and list their names along with the order count.*/

select count(o.id),o.id,c.first_name from orders o,customer c where o.customer_id=c.id group by customer_id;