ASSIGNMENT 1:

EXERCISE:

Create two tables

1. Products: columns (product_id,product_name,category and price)

table

Data Output | Messages | Notifications

	product_id [PK] integer	product_name character varying (50)	category character varying (50)	price integer
1	1	Laptop	Electronics	1000
2	3	Table	Furniture	150
3	5	Book	Books	20
4	2	Mobile	Electronics	500

2. Orders : columns(order_id, customer_name, product_id, quantity, order_date)

Table

Data Output | Messages | Notifications

	order_id [PK] integer	customer_name character varying (50)	product_id integer	quantity integer	order_date date
1	1	John Doe	1	2	2024-07-01
2	2	Jane Smith	3	1	2024-07-01
3	3	Alice Johnson	2	3	2024-07-02
4	4	Bob Brown	5	5	2024-07-02

QUESTIONS:

1. perform CRUD

Query

```
CREATE table Products(
        product_id int primary key,
        product_name varchar(50),
        category varchar(50), price int
);

CREATE table Orders(
        order_id int primary key,
        customer_name varchar(50),
        product_id int,
        quantity int,
        order_date date
);

INSERT INTO Products (product_id, product_name, category, price)
VALUES
(1, 'Laptop', 'Electronics', 1000),
(2, 'Smartphone', 'Electronics', 500),
(3, 'Table', 'Furniture', 150),
(4, 'Chair', 'Furniture', 75),
(5, 'Book', 'Books', 20);

update products
        set product_name='Mobile'
        where product_id=2;

delete from products
        where product_name='Chair';

select * from products;

INSERT INTO Orders (order_id, customer_name, product_id, quantity, order_date)
VALUES
```

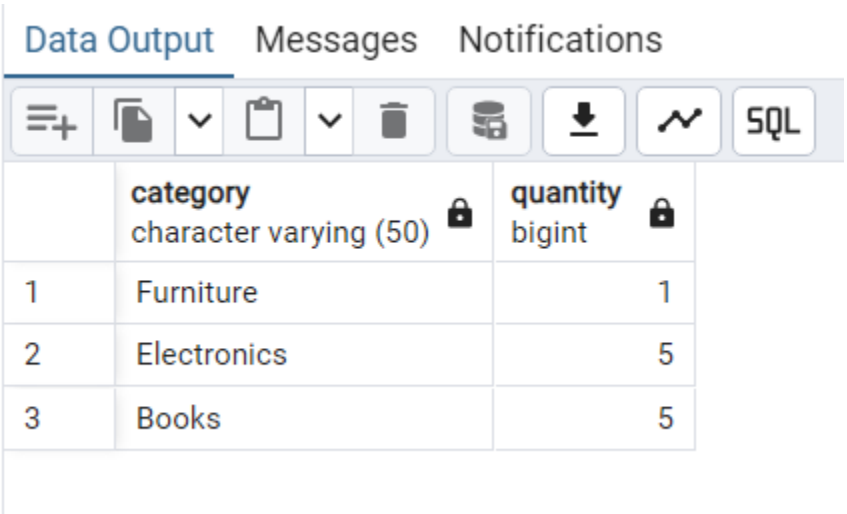
```
(1, 'John Doe', 1, 2, '2024-07-01'),
(2, 'Jane Smith', 3, 1, '2024-07-01'),
(3, 'Alice Johnson', 2, 3, '2024-07-02'),
(4, 'Bob Brown', 5, 5, '2024-07-02'),
(5, 'Charlie Black', 4, 4, '2024-07-03');

update orders
        set customer_name = 'Ganesh Saud'
        where order_id=5;

delete from orders
        where product_id=4;

select * from orders;
```

2. Calculate the total quantity ordered for each product category in the orders table.

Query

```
SELECT category, SUM(quantity) AS quantity
FROM products p JOIN orders o ON p.product_id=o.product_id
GROUP BY category;
```

Output



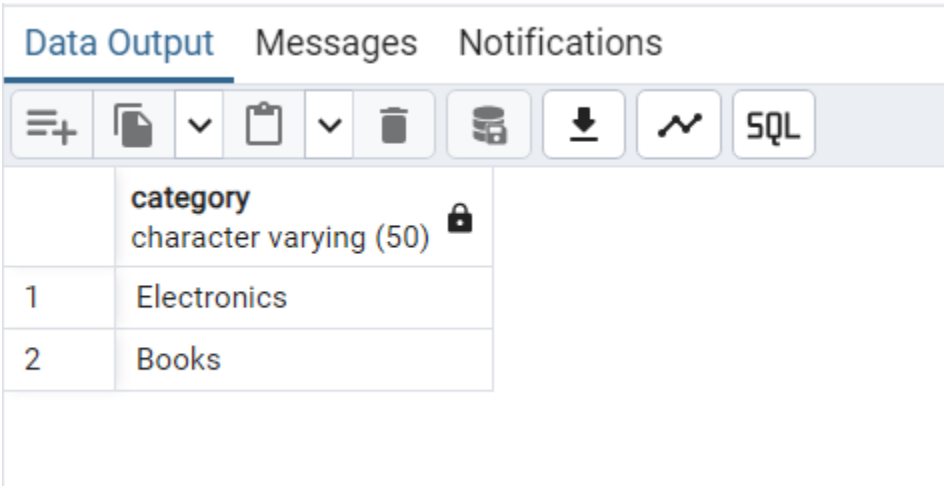
	category character varying (50)	quantity bigint
1	Furniture	1
2	Electronics	5
3	Books	5

3. Find categories where the total number of products ordered is greater than 4.

Query

```
SELECT category
FROM Orders o JOIN Products p ON o.product_id = p.product_id
GROUP BY category
HAVING SUM(quantity) > 4;
```

Output



	category character varying (50)
1	Electronics
2	Books