TASK 4

1.Basic Data Retrieval

```sql
SELECT * FROM customers;
SELECT * FROM products;
SELECT * FROM orders;
```

order_id	customer_id	product_id	order_date
1	1	1	2024-05-10
2	1	4	2024-05-15
3	2	2	2024-06-12
4	2	8	2024-06-20
5	3	3	2024-07-01
6	3	5	2024-07-03
7	4	6	2024-07-15

1 2 3

customer_id	name	email	country
1	Alice Johnson	alice@example.com	USA
2	Bob Smith	bob@example.com	Canada
3	Charlie Brown	charlie@example.com	UK
4	Diana Adams	diana@example.com	USA
5	Ethan Clark	ethan@example.com	Australia
6	Fiona Davis	fiona@example.com	Canada
7	George Miller	george@example.com	Germany

1 2 3

product_id	product_name	category	price
1	Laptop	Electronics	1200
2	Smartphone	Electronics	800
3	Tablet	Electronics	500
4	Headphones	Accessories	150
5	Wireless Mouse	Accessories	40
6	Office Chair	Furniture	300
7	Desk Lamp	Furniture	70

1 2 3

Filtering & Conditions

- **Find all orders after July 1, 2024**

```sql
SELECT * FROM orders
WHERE order_date > '2024-07-01';
```

customer_id	name	email	country
1	Alice Johnson	alice@example.com	USA
2	Bob Smith	bob@example.com	Canada
3	Charlie Brown	charlie@example.com	UK
4	Diana Adams	diana@example.com	USA
5	Ethan Clark	ethan@example.com	Australia
6	Fiona Davis	fiona@example.com	Canada
7	George Miller	george@example.com	Germany

1 2 3

- List customers who bought more than 2 items in a single order

```sql
SELECT * FROM orders
WHERE quantity > 2;
```

order_id	customer_id	product_id	order_date	quantity
4	2	8	2024-06-20	3
13	7	4	2024-08-15	4

2.Aggregates & Grouping

- **Total quantity sold per product**

```sql
SELECT p.product_name, SUM(o.quantity) AS total_sold
FROM orders o
JOIN products p ON o.product_id = p.product_id
GROUP BY p.product_name
ORDER BY total_sold DESC;
```

product_name	total_sold
Headphones	6
Backpack	4
Wireless Mouse	3
Tablet	3
Laptop	3
Smartphone	2
Desk Lamp	2
Office Chair	1

- Average order quantity per country

```sql
SELECT c.country, AVG(o.quantity) AS avg_quantity
FROM orders o
JOIN customers c ON o.customer_id = c.customer_id
GROUP BY c.country;
```

country	avg_quantity
Australia	1.5
Canada	1.5
Germany	4
UK	1.5
USA	1.3333333333333333

3.JOIN Queries

- **Show customer name, product name, and order date**

```sql
SELECT c.name, p.product_name, o.order_date
FROM orders o
JOIN customers c ON o.customer_id = c.customer_id
JOIN products p ON o.product_id = p.product_id;
```

name	product_name	order_date
Alice Johnson	Laptop	2024-05-10
Alice Johnson	Headphones	2024-05-15
Bob Smith	Smartphone	2024-06-12
Bob Smith	Backpack	2024-06-20
Charlie Brown	Tablet	2024-07-01
Charlie Brown	Wireless Mouse	2024-07-03
Diana Adams	Office Chair	2024-07-15
Diana Adams	Smartphone	2024-07-16
Ethan Clark	Desk Lamp	2024-08-02
Ethan Clark	Backpack	2024-08-03
Fiona Davis	Laptop	2024-08-10

- List all customers with their orders (including those without orders)

```sql
SELECT c.name, o.order_id, o.order_date
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id;
```

name	order_id	order_date
Alice Johnson	1	2024-05-10
Alice Johnson	2	2024-05-15
Bob Smith	3	2024-06-12
Bob Smith	4	2024-06-20
Charlie Brown	5	2024-07-01
Charlie Brown	6	2024-07-03
Diana Adams	7	2024-07-15
Diana Adams	8	2024-07-16
Ethan Clark	9	2024-08-02
Ethan Clark	10	2024-08-03

4.Subqueries

- **Find products more expensive than the average price**

```sql
SELECT product_name, price
FROM products
WHERE price > (SELECT AVG(price) FROM products);
```

product_name	price
Laptop	1200
Smartphone	800
Tablet	500

- List customers who have ordered 'Laptop'

```sql
SELECT name
FROM customers
WHERE customer_id IN (
    SELECT customer_id
    FROM orders
    WHERE product_id = (
        SELECT product_id FROM products WHERE product_name = 'Laptop'
    )
);
```

name
Alice Johnson
Fiona Davis
Hannah Scott

5.Views

- **Create a view of high-value orders (> $500 total price)**

```sql
DROP VIEW IF EXISTS high_value_orders;

CREATE VIEW high_value_orders AS
SELECT o.order_id, c.name, p.product_name, (o.quantity * p.price) AS total_price
FROM orders o
JOIN customers c ON o.customer_id = c.customer_id
JOIN products p ON o.product_id = p.product_id
WHERE (o.quantity * p.price) > 500;

SELECT * FROM high_value_orders;
```

order_id	name	product_name	total_price
1	Alice Johnson	Laptop	1200
3	Bob Smith	Smartphone	800
8	Diana Adams	Smartphone	800
11	Fiona Davis	Laptop	1200
13	George Miller	Headphones	600
14	Hannah Scott	Tablet	1000
15	Hannah Scott	Laptop	1200

- Indexes

```sql
CREATE INDEX idx_product_category ON products(category);

SELECT * FROM products
WHERE category = 'Electronics';
```

product_id	product_name	category	price
1	Laptop	Electronics	1200
2	Smartphone	Electronics	800
3	Tablet	Electronics	500