Project 3

Title: Online Store Order Management System (PostgreSQL)

Task 1 Database Creation:

Question:

Create a database named OnlineStore

Query:

```
CREATE DATABASE OnlineStore;

USE OnlineStore;
```

Task 2 Create tables:

Question:

Customers (CUSTOMER_ID, NAME, EMAIL, PHONE, ADDRESS)

Query:

```
CREATE TABLE Customers (

CUSTOMER_ID SERIAL PRIMARY KEY,

NAME TEXT NOT NULL,

EMAIL TEXT NOT NULL,

PHONE NUMERIC NOT NULL,

ADDRESS TEXT

);
```

Question:

Products (PRODUCT_ID, PRODUCT_NAME, CATEGORY, PRICE, STOCK)

Query:

```
CREATE TABLE Products (
PRODUCT_ID SERIAL PRIMARY KEY,
PRODUCT_NAME TEXT NOT NULL,
CATEGORY TEXT NOT NULL,
PRICE NUMERIC(10, 2) NOT NULL,
STOCK INTEGER NOT NULL
);
```

Question:

Orders (ORDER_ID, CUSTOMER_ID, PRODUCT_ID, QUANTITY, ORDER_DATE) Set up foreign keys linking Orders to Customers and Products.

Query:

```
CREATE TABLE Orders (
ORDER_ID SERIAL PRIMARY KEY,
CUSTOMER_ID INT REFERENCES Customers(CUSTOMER_ID),
PRODUCT_ID INT REFERENCES Products(PRODUCT_ID),
QUANTITY INT NOT NULL,
ORDER_DATE DATE NOT NULL
);
```

Task 3 Data Creation:

Question:

Insert sample into Customers table

Query:

```
INSERT INTO Customers (NAME, EMAIL, PHONE, ADDRESS) VALUES
('Elan', 'elan@gmail.com', '9034567890', '123 Bharat St'),
('David', 'david@gmail.com', '9345678901', '456 Thomas St'),
('Charlie', 'charlie@gmail.com', '9456789012', '789 Pine St'),
('Diana', 'diana@gmail.com', '9567890123', '321 Gandhi St'),
('Lora', 'lora@gmail.com', '9967890123', '321 Anandha St'),
('Carol', 'carol@gamil.com', '9678901234', '654 Joseph St');
```

Question:

Insert sample into Products table

Query:

```
INSERT INTO Products (PRODUCT_NAME, CATEGORY, PRICE, STOCK) VALUES
('Laptop', 'Electronics', 1200.00, 5),
('Phone', 'Electronics', 800.00, 0),
('Desk Chair', 'Furniture', 150.00, 10),
('Notebook', 'Stationery', 5.00, 50),
('Headphones', 'Electronics', 100.00, 0),
('Phone', 'Electronics', 700.00, 0),
('Pen', 'Stationery', 2.00, 100);
```

Question:

Insert sample into Orders table

Query:

```
INSERT INTO Orders (CUSTOMER_ID, PRODUCT_ID, QUANTITY, ORDER_DATE)
VALUES
(1, 1, 1, '2025-08-01'),
(1, 4, 5, '2025-08-03'),
(2, 3, 2, '2025-07-15'),
(2, 6, 10, '2025-07-16'),
(3, 2, 1, '2025-06-10'),
(3, 5, 2, '2025-08-10'),
(4, 3, 1, '2025-01-05'),
(1, 1, 1, '2024-08-01'),
(5, 2, 1, '2024-01-01'),
(5, 1, 1, '2025-08-20'),
(2, 4, 5, '2023-08-03');
```

Task 4 Order Management:

Question:

a) Retrieve all orders placed by a specific customer.

Query:

```
SELECT O.*
FROM Orders O
JOIN Customers C ON O.CUSTOMER_ID = C.CUSTOMER_ID
WHERE C.NAME = 'Elan';
```

OUTPUT:

ORDER_ID	CUSTOMER_ID	PRODUCT_ID	QUANTITY	ORDER_DATE
1	1	1	1	2025-08-01
2	1	4	5	2025-08-03
8	1	1	1	2024-08-01

Question:

b) Find products that are out of stock

Query:

```
SELECT * FROM Products
WHERE STOCK = 0;
```

OUTPUT:

PRODUCT_ID	PRODUCT_NAME	CATEGORY	PRICE	STOCK
2	Phone	Electronics	800.00	0
5	Headphones	Electronics	100.00	0
6	Phone	Electronics	700.00	0

Question:

c) Calculate the total revenue generated per product

Query:

```
SELECT
P.PRODUCT_NAME,
SUM(O.QUANTITY * P.PRICE) AS TOTAL_REVENUE
FROM Orders O
JOIN Products P ON O.PRODUCT_ID = P.PRODUCT_ID
GROUP BY P.PRODUCT_NAME;
```

OUTPUT:

PRODUCT_NAME	TOTAL_REVENUE
Laptop	3600.00
Notebook	50.00
Desk Chair	450.00
Phone	8600.00
Headphones	200.00

Question:

d) Retrieve the top 5 customers by total purchase amount.

Query:

```
SELECT
C.NAME,
SUM(O.QUANTITY * P.PRICE) AS TOTAL_SPENT
FROM Orders O
JOIN Customers C ON O.CUSTOMER_ID = C.CUSTOMER_ID
JOIN Products P ON O.PRODUCT_ID = P.PRODUCT_ID
```

```
GROUP BY C.NAME
ORDER BY TOTAL_SPENT DESC
LIMIT 5;
```

OUTPUT:

NAME	TOTAL_SPENT
David	7325.00
Elan	2425.00
Lora	2000.00
Charlie	1000.00
Diana	150.00

Question:

e) Find customers who placed orders in at least two different product categories.

Query:

```
SELECT C.NAME
FROM Orders O
JOIN Customers C ON O.CUSTOMER_ID = C.CUSTOMER_ID
JOIN Products P ON O.PRODUCT_ID = P.PRODUCT_ID
GROUP BY C.NAME
HAVING COUNT(DISTINCT P.CATEGORY) >= 2;
```

OUTPUT:

NAME
David
Elan

Task 5 Analytics:

Question:

a) Find the month with the highest total sales.

Query:

```
SELECT
DATE_FORMAT(O.ORDER_DATE, '%Y-%m') AS MONTH,
SUM(O.QUANTITY * P.PRICE) AS TOTAL_SALES
FROM Orders O
JOIN Products P ON O.PRODUCT_ID = P.PRODUCT_ID
GROUP BY DATE_FORMAT(O.ORDER_DATE, '%Y-%m')
ORDER BY TOTAL_SALES DESC
LIMIT 1;
```

OUTPUT:

MONTH	TOTAL_SALES
2025-07	7300.00

Question:

b) Identify products with no orders in the last 6 months.

Query:

```
SELECT DISTINCT
p.PRODUCT_ID,
p.PRODUCT_NAME,
```

```
p.CATEGORY
FROM
Products p
LEFT JOIN
Orders o
ON p.PRODUCT_ID = o.PRODUCT_ID
AND o.ORDER_DATE >= DATE_SUB(CURRENT_DATE, INTERVAL 6 MONTH)
WHERE
o.PRODUCT_ID IS NULL;
```

OUTPUT:

PRODUCT_ID	PRODUCT_NAME	CATEGORY
7	Pen	Stationery

Question:

c) Retrieve customers who have never placed an order.

Query:

```
SELECT * FROM Customers
WHERE CUSTOMER_ID NOT IN (
SELECT DISTINCT CUSTOMER_ID FROM Orders
);
```

OUTPUT:

CUSTOMER_ID	NAME	EMAIL	PHONE	ADDRESS
6	Carol	carol@gamil.com	9678901234	654 Joseph St

Question:

d) Calculate the average order value across all orders.

Query:

```
SELECT
ROUND(AVG(O.QUANTITY * P.PRICE),0) AS AVERAGE_ORDER_VALUE
FROM Orders O
JOIN Products P ON O.PRODUCT_ID = P.PRODUCT_ID;
```

OUTPUT:

AVERAGE_ORDER_VALUE
1173