Assignment-3

Name :- M. Poliveddy

Subject code :- CSA0563

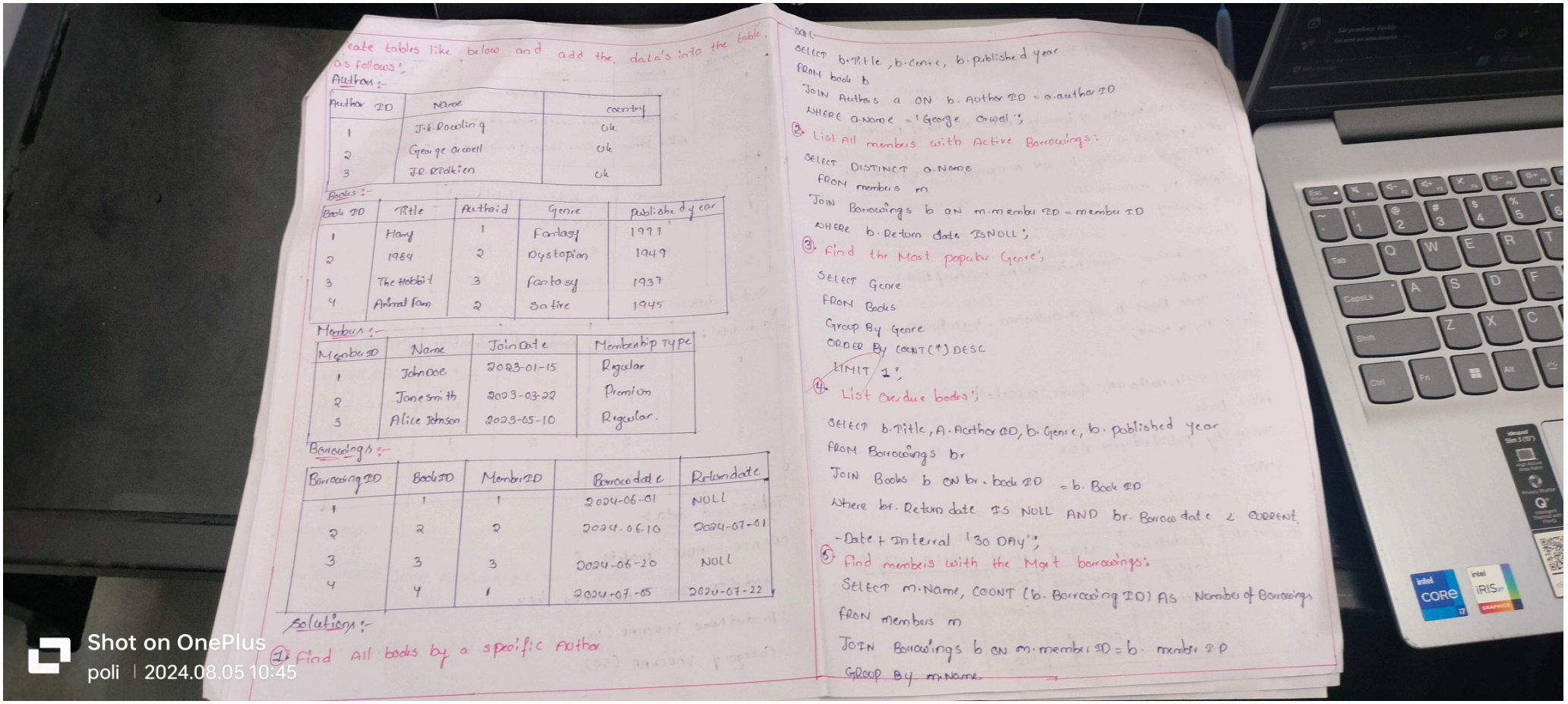
Subject name :- Data base management Systems

Register no :- 192325056

Department :- AI&ML

Date of Submission :- 31/07/2024

18/20



Create tables like below and add the data's into the table as follows:

Authors:-

Author ID	Name	country
1	J.K.Rowling	Uk
2	George Orwell	Uk
3	J.R.R Tolkien	Uk

Books:-

Book ID	Title	Authid	Genre	Published year
1	Harry	1	Fantasy	1997
2	1984	2	Dystopian	1949
3	The Hobbit	3	fantasy	1937
4	Animal Farm	2	Satire	1945

Members:-

MemberID	Name	JoinDate	Membership Type
1	John Doe	2023-01-15	Regular
2	Jane Smith	2023-03-22	Premium
3	Alice Johnson	2023-05-10	Regular.

Borrowings:-

BorrowingID	BookID	MemberID	Borrowdate	Returndate
1	1	1	2024-06-01	NULL
2	2	2	2024-06-10	2024-07-01
3	3	3	2024-06-20	NULL
4	4	1	2024-07-05	2024-07-22

Solutions:-

1. Find All books by a specific Author

```
SQL:-
SELECT b.Title, b.Genre, b.published year
FROM book b
JOIN Authors a ON b.AuthorID = a.authorID
WHERE a.Name = 'George Orwell';
```

2. List All members with Active Borrowings:

```
SELECT DISTINCT a.Name
FROM members m
JOIN Borrowings b ON m.memberID = member ID
WHERE b.Return date IS NULL;
```

3. Find the Most popular Genre;

```
SELECT Genre
FROM Books
Group By Genre
ORDER By COUNT(*) DESC
LIMIT 1;
```

4. List Overdue books;

```
SELECT b.Title, A.AcrthorID, b.Genre, b.published year
FROM Borrowings br
JOIN Books b ON br.bookID = b.BookID
Where br.Return date IS NULL AND br.Borrow date < CURRENT
-Date + Interval '30 DAY';
```

5. Find members with the Most borrowings:

```
SELECT m.Name, COUNT (b.BorrowingID) AS Number of Borrowings
FROM members m
JOIN Borrowings b ON m.memberID = b. member ID
GROUP By m.Name.
```

ORDER BY number of Borrowings DESC
LIMIT 1;

6. Calculate the avg no of books Borrowed per member.

SELECT avg(number of borrowing 20) AS Number of Borrowings
FROM members m
LEFT JOIN Borrowings b ON m.member ID = b.member ID
GROUP BY a.member ID
) Subquery

7. List all Authors and the number of books they have written

SELECT a.name, COUNT(b.book ID) AS number of books
FROM Authors a
LEFT JOIN Books b on a.Author ID = b.Author ID
Group By a.name;

8. SE

SELECT Title, Author ID, Genre, published year
FROM Books
ORDER BY published year ASC
LIMIT 1;

9. Create table like below and add the data's into the table as follows.

Product ID	product name	category	price	stock Quantity
1	Laptop	Electronics	1000	50
2	Smartphone	Electronics	600	200
3	T-shirt	clothing	20	500
4	Jeans	clothing	40	300

Customers :-

Customer ID	Name	Email	Join date
1	John doe	John.doe@example.com	2023-01-01
2	Smartphone	Jane.Smith@Example.com	2023-02-15
3	Alice Johnson	alice.Johnson@Example.com	2023-03-10

orders:-

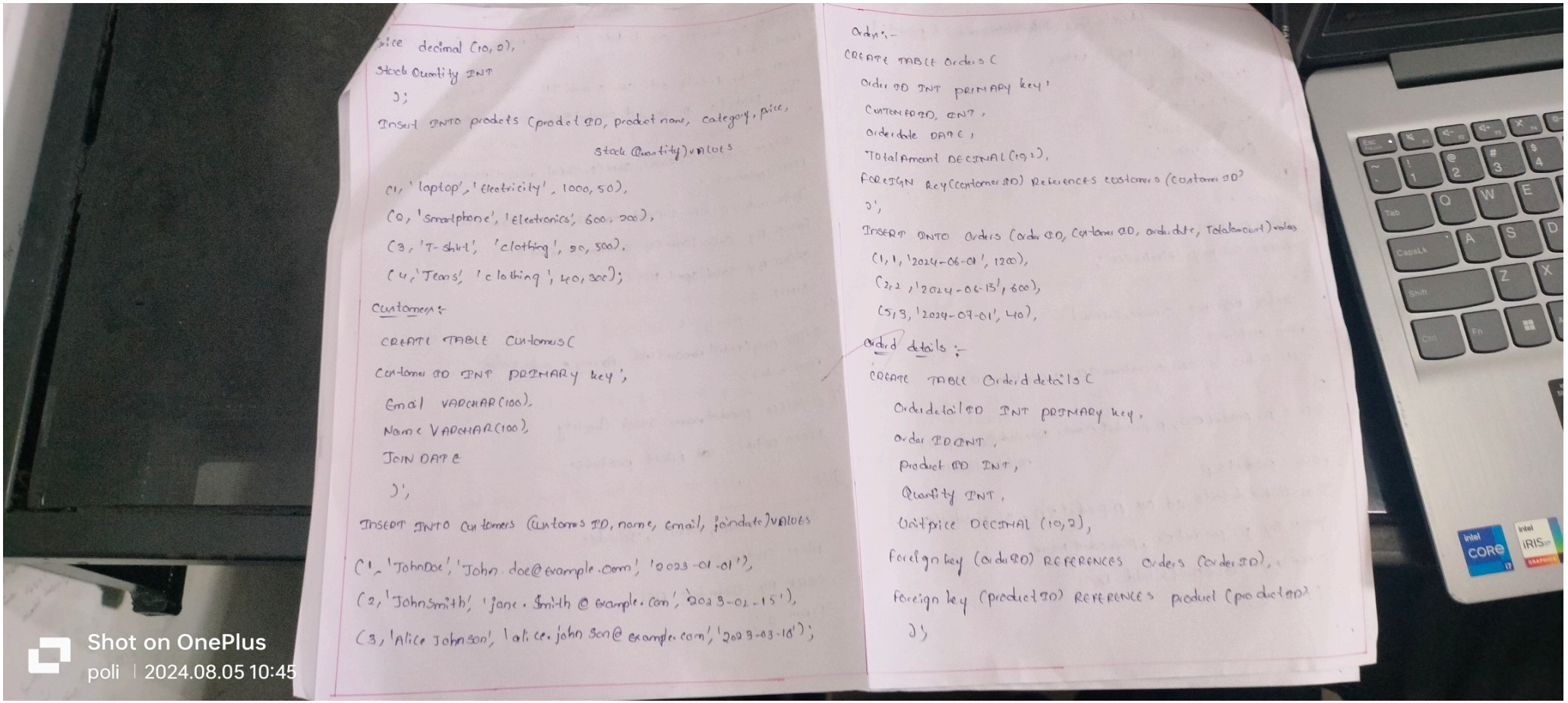
Order ID	Customer ID	Order date	total amount
1	1	2024-06-01	1200
2	2	2024-06-15	600
3	3	2024-07-01	40

Order details :-

order detail id	order id	product id	Quantity	Unit price
1	1	1	1	1000
2	1	2	1	200
3	2	2	1	600
4	3	4	1	40

Tables and data:-

```
CREATE TABLE products (
    product ID INT PRIMARY KEY,
    Product Name VARCHAR (100),
    Category VARCHAR (50),
```



price decimal (10, 2),
Stock Quantity INT
);
Insert INTO products (product ID, product name, category, price,
stock Quantity) values

(1, 'laptop', 'Electricity', 1000, 50),
(2, 'smartphone', 'Electronics', 600, 200),
(3, 'T-shirt', 'clothing', 20, 500),
(4, 'Jeans', 'clothing', 40, 300);

Customers :-

CREATE TABLE Customers (
Customer ID INT PRIMARY key',
Email VARCHAR(100),
Name VARCHAR(100),
Join DATE
);

INSERT INTO Customers (Customer ID, name, email, joindate) VALUES

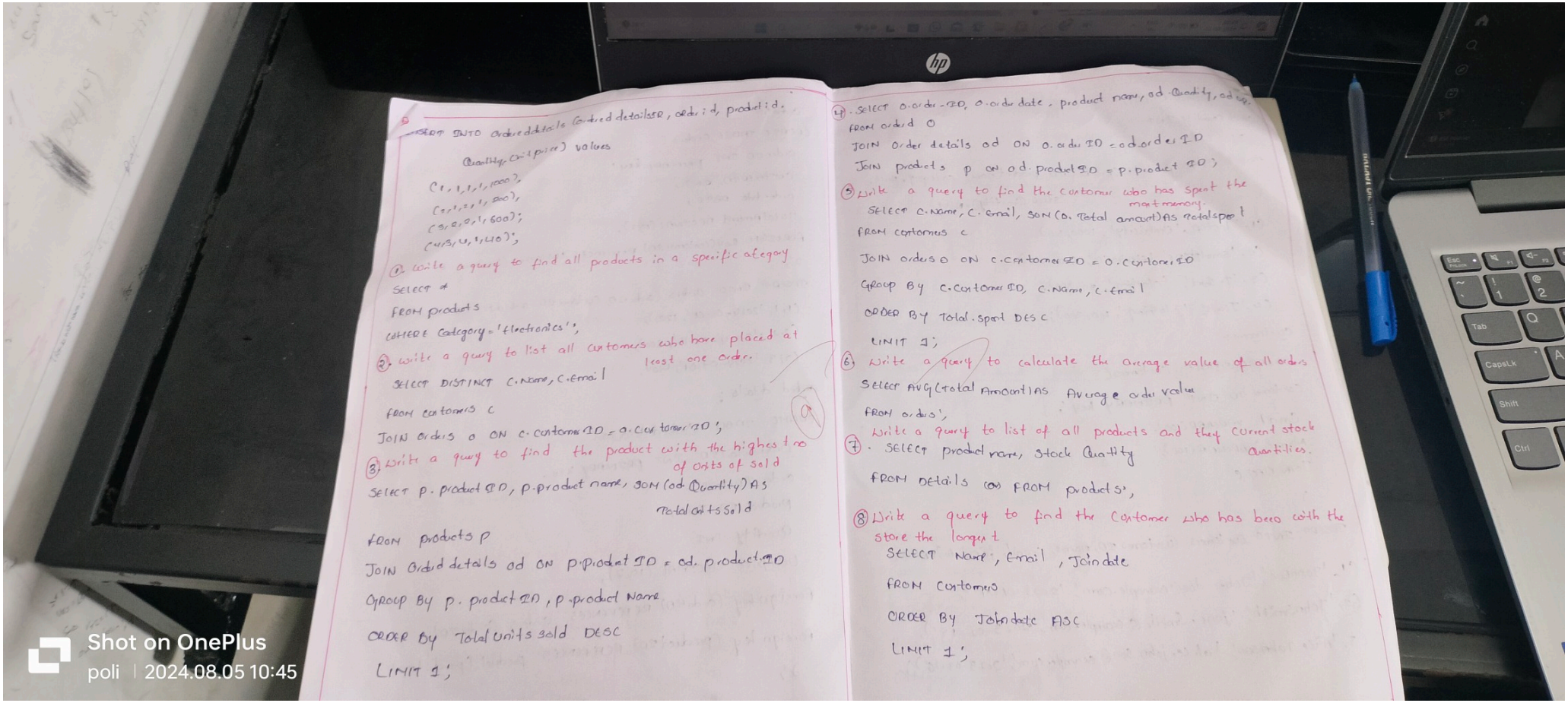
(1, 'JohnDoe', 'John.doe@example.com', '2023-01-01'),
(2, 'JohnSmith', 'jane.smith@example.com', '2023-02-15'),
(3, 'Alice Johnson', 'alice.johnson@example.com', '2023-03-10');

Orders :-

CREATE TABLE Orders (
Order ID INT PRIMARY key'
Customer ID, INT,
Orderdate DATE,
TotalAmount DECIMAL(10,2),
FOREIGN Key(Customer ID) References customers (Customer ID)
);
INSERT INTO Orders (Order ID, Customer ID, order date, Totalamount) values
(1, 1, '2024-06-01', 1200),
(2, 2, '2024-06-15', 600),
(3, 3, '2024-07-01', 40),

order details :-

CREATE TABLE Orderddetails (
Orderdetail ID INT PRIMARY key,
order ID INT,
Product ID INT,
Quantity INT,
Unitprice DECIMAL (10,2),
foreign key (order ID) REFERENCES Orders (Order ID),
foreign key (product ID) REFERENCES product (product ID)
);



INSERT INTO orderdetails (orderdetailsID, orderid, productid,
Quantity, Unitprice) values
(1, 1, 1, 1000),
(2, 1, 2, 200),
(3, 2, 3, 600),
(4, 3, 4, 140);

① Write a query to find all products in a specific category
SELECT *
FROM products
WHERE Category = 'Electronics';

② Write a query to list all customers who have placed at least one order.
SELECT DISTINCT C.Name, C.Email
FROM customers C
JOIN orders o ON C.customerID = o.customerID;

③ Write a query to find the product with the highest no of units of sold
SELECT p.productID, p.product name, SUM(od.Quantity) AS Total units sold
FROM products P
JOIN Orderdetails od ON p.productID = od.productID
GROUP BY p.productID, p.product Name
ORDER BY Total units sold DESC
LIMIT 1;

④. SELECT o.order_ID, o.orderdate, product name, od.Quantity, od.
FROM orders O
JOIN Order details od ON o.orderID = od.orderID
JOIN products p ON od.productID = p.productID;

⑤ Write a query to find the customer who has spent the most money.
SELECT C.Name, C.Email, SUM(O.Total amount) AS totalspent
FROM customers C
JOIN orders O ON C.customerID = O.customerID
GROUP BY C.customerID, C.Name, C.Email
ORDER BY Total.spent DESC
LIMIT 1;

⑥ Write a query to calculate the average value of all orders
SELECT AVG(Total Amount) AS Average order value
FROM orders;

⑦ Write a query to list of all products and their current stock quantities.
SELECT product name, stock Quantity
FROM Details (or) FROM products;

⑧ Write a query to find the customer who has been with the store the longest
SELECT Name, Email, Joindate
FROM customers
ORDER BY Joindate ASC
LIMIT 1;