- 1. Write SQL queries to create the necessary tables.
- 2. Write SQL queries to insert sample data into each table.
- 3. Write SQL queries to retrieve a list of all books, customers, and orders.

Books Table Creation:

```
CREATE TABLE Books (
ISBN VARCHAR(13) PRIMARY KEY,
Title VARCHAR(255),
Author VARCHAR(255),
Genre VARCHAR(50),
Price DECIMAL(10, 2),
QuantityInStock INT
);

INSERT INTO Books
VALUES
('ISBN1234567890', 'Space', 'Vinayak More', 'Fiction', 25.99, 50),
('ISBN2345678901', 'BLackHole', 'Tonny Drew', 'Mystery', 19.99, 30),
('ISBN3456789012', 'WW2', 'Shinzani Jain', 'Science Fiction', 30.50, 40);
```

SELECT * FROM Books;



Customers Table Creation:

```
CREATE TABLE Customers (
CustomerID INT PRIMARY KEY,
Name VARCHAR(255),
Email VARCHAR(255),
PhoneNumber VARCHAR(20),
ShippingAddress VARCHAR(255)
);
```

INSERT INTO Customers

VALUES

- (1, 'sandesh sawant', 'sandesh4@gmail.com', '1234567890', 'Andheri'),
- (2, 'Raja Donathula', 'Raja@gmail.com', '9876543210', 'Borivali');

SELECT * FROM Customers;



Orders Table Creation:

```
CREATE TABLE Orders (
OrderID INT PRIMARY KEY,
CustomerID INT,
OrderDate DATE,
TotalAmount DECIMAL(10, 2),
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));

INSERT INTO Orders
VALUES
(101, 1, '2023-12-20', 76.97),
(102, 2, '2023-12-12', 45.98);
```

SELECT * FROM Orders;



OrderDetails Table Creation:

```
CREATE TABLE OrderDetails (
OrderID INT,
ISBN VARCHAR(13),
Quantity INT,
Subtotal DECIMAL(10, 2),
PRIMARY KEY (OrderID, ISBN),
FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
FOREIGN KEY (ISBN) REFERENCES Books(ISBN)
);
INSERT INTO OrderDetails VALUES
(101, 'ISBN1234567890', 2, 51.98),
(102, 'ISBN2345678901', 1, 19.99),
(102, 'ISBN3456789012', 3, 24.99);
```



4. Write SQL queries to update customer information

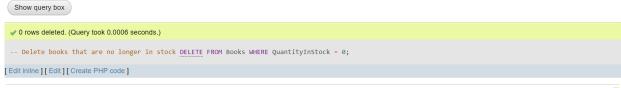
UPDATE Customers

SET PhoneNumber = '9769798448' WHERE CustomerID = 1;



5. Write SQL queries to delete books that are no longer in stock.

DELETE FROM Books WHERE QuantityInStock = 0;



6. Write SQL queries to generate a report showing the total revenue generated from orders.

SELECT SUM(TotalAmount) AS TotalRevenue FROM Orders;

The state of the s
Generate a report showing total revenue <u>SELECT</u> <u>SUM</u> (TotalAmount) AS TotalRevenue FROM Orders;
[Edit inline] [Edit] [Create PHP code]
☐ Show all Number of rows: 25 ➤ Filter rows: Search this table
Extra options
TotalRevenue 122.95
☐ Show all Number of rows: 25 ➤ Filter rows: Search this table