```sql
-- Table: Customer
CREATE TABLE project_Customer (
  Email VARCHAR(30) NOT NULL,
  CustomerID INT NOT NULL,
  Name VARCHAR(20) NOT NULL,
  Phone BIGINT NOT NULL,
  PRIMARY KEY (CustomerID)
);

-- Table: Orders (renamed from Order to avoid conflict with SQL reserved word)
CREATE TABLE project_Orders (
  OrderID INT NOT NULL,
  CustomerID INT NOT NULL,
  PRIMARY KEY (OrderID),
  FOREIGN KEY (CustomerID) REFERENCES project_Customer(CustomerID)
);

-- Table: Payment
CREATE TABLE project_Payment (
  PaymentID INT NOT NULL,
  OrderID INT NOT NULL,
  BookID INT NOT NULL,
  Payment_method VARCHAR(15) NOT NULL,
  PRIMARY KEY (PaymentID),
  FOREIGN KEY (OrderID) REFERENCES project_Orders(OrderID)
);

-- Table: Author
CREATE TABLE project_Author (
  AuthorID INT NOT NULL,
  First_name VARCHAR(15) NOT NULL,
  Last_name VARCHAR(15) NOT NULL,
  PRIMARY KEY (AuthorID)
);

-- Table: Book
CREATE TABLE project_Book (
  BookID INT NOT NULL,
  Title VARCHAR(100) NOT NULL,
  Price DECIMAL(10, 2) NOT NULL,
  AuthorID INT NOT NULL,
  PRIMARY KEY (BookID),
  FOREIGN KEY (AuthorID) REFERENCES project_Author(AuthorID)
);

-- Table: Relationship (Order–Book mapping)
CREATE TABLE Relationship (
  OrderID INT NOT NULL,
  BookID INT NOT NULL,
  PRIMARY KEY (OrderID, BookID),
  FOREIGN KEY (OrderID) REFERENCES project_Orders(OrderID),
  FOREIGN KEY (BookID) REFERENCES project_Book(BookID)
);

ALTER TABLE project_Customer MODIFY Phone BIGINT NOT NULL;

#Inster rows
INSERT INTO project_Customer (CustomerID, Email, Name, Phone) VALUES
(1, 'alice@gmail.com', 'Alice', 1234567890),
(2, 'bob@yahoo.com', 'Bob', 2345678901),
(3, 'carol@outlook.com', 'Carol', 3456789012),
(4, 'dave@aol.com', 'Dave', 4567890123),
(5, 'emma@gmail.com', 'Emma', 5678901234),
(6, 'frank@hotmail.com', 'Frank', 6789012345);

INSERT INTO project_Author (AuthorID, First_name, Last_name) VALUES
(101, 'George', 'Orwell'),
(102, 'Jane', 'Austen'),
(103, 'Mark', 'Twain'),
(104, 'Mary', 'Shelley'),
(105, 'Leo', 'Tolstoy'),
(106, 'Virginia', 'Woolf');

INSERT INTO project_Book (BookID, Title, Price, AuthorID) VALUES
(201, '1984', 15.99, 101),
(202, 'Pride & Prejudice', 12.50, 102),
(203, 'Tom Sawyer', 13.75, 103),
(204, 'Frankenstein', 10.00, 104),
(205, 'War and Peace', 20.99, 105),
(206, 'Mrs Dalloway', 14.25, 106);

INSERT INTO project_Orders (OrderID, CustomerID) VALUES
(301, 1),
(302, 2),
(303, 3),
(304, 4),
(305, 5),
(306, 6);

INSERT INTO project_Payment (PaymentID, BookID, Payment_method, OrderID) VALUES
(401, 201, 'Credit Card', 301),
(402, 202, 'PayPal', 301),
(403, 203, 'Debit Card', 302),
(404, 204, 'Cash', 303),
(405, 205, 'Credit Card', 304),
(406, 206, 'PayPal', 305);


#This query shows each customer's name and email along with their order IDs.
# It uses a cartesian product with a WHERE clause to join project_Customer and
project_Orders.
SELECT c.Name, c.Email, o.OrderID
FROM project_Customer c, project_Orders o
WHERE c.CustomerID = o.CustomerID;


#
# This query retrieves author names and the titles of the books they have written.
# It uses a cartesian product with a WHERE clause to join project_Author and
project_Book via AuthorID.
SELECT a.First_name, a.Last_name, b.Title
FROM project_Author a, project_Book b
WHERE a.AuthorID = b.AuthorID;

#
# This query returns the total amount paid per customer by summing up the book
prices.
# It uses INNER JOINs across project_Customer, project_Orders, project_Payment, and
```

```
project_Book,
# and groups results by customer name and email.
SELECT c.Name, c.Email, SUM(b.Price) AS TotalSpent
FROM project_Customer c
INNER JOIN project_Orders o ON c.CustomerID = o.CustomerID
INNER JOIN project_Payment p ON o.OrderID = p.OrderID
INNER JOIN project_Book b ON p.BookID = b.BookID
GROUP BY c.Name, c.Email;

# This query shows each book title along with the number of times it was purchased
and its total revenue.
# It joins project_Book with project_Payment and groups by book to calculate total
sales and revenue.
SELECT b.Title, COUNT(p.BookID) AS TimesPurchased, SUM(b.Price) AS TotalRevenue
FROM project_Book b
INNER JOIN project_Payment p ON b.BookID = p.BookID
GROUP BY b.Title;



# This query shows each customer's name along with the total amount they have spent
across all orders.
# It joins project_Customer, project_Orders, project_Payment, and project_Book,
# and groups by customer name to compute total spending.
SELECT c.Name, SUM(b.Price) AS TotalAmountSpent
FROM project_Customer c
INNER JOIN project_Orders o ON c.CustomerID = o.CustomerID
INNER JOIN project_Payment p ON o.OrderID = p.OrderID
INNER JOIN project_Book b ON p.BookID = b.BookID
GROUP BY c.Name;


CREATE VIEW project_Top3_BestSellingBooks AS
SELECT
  b.Title,
  COUNT(p.BookID) AS TimesSold,
  SUM(b.Price) AS TotalRevenue
FROM project_Book b
INNER JOIN project_Payment p ON b.BookID = p.BookID
GROUP BY b.Title
ORDER BY TimesSold DESC
LIMIT 3;
```