PROJECT SOLUTION

```
CREATE TABLE Books(
        Book_ID INT PRIMARY KEY,
        Title VARCHAR(100),
   Author VARCHAR(100),
        Genre VARCHAR(100),
        Published_Year INT,
        Price NUMERIC(10,2),
        Stock INT
);

CREATE TABLE Customer(
        Customer_ID INT PRIMARY KEY,
        Name VARCHAR(100),
        Email VARCHAR(100),
        Phone VARCHAR(100),
        City VARCHAR(100),
        Country VARCHAR(150)
);

CREATE TABLE Orders(
        Order_ID INT PRIMARY KEY,
        Customer_ID INT REFERENCES Customer(Customer_ID),
        Book_ID INT REFERENCES Books(Book_ID),
        Order_Date DATE,
        Quantity INT,
        Total_Amount NUMERIC(10,2)
);



--1) Retrieve all books in the "Fiction" genre
SELECT * FROM Books
WHERE Genre='Fiction';


--2) Find books published after the year 1950
SELECT * FROM Books
WHERE Published_Year>'1950';


--3) List all customers from the Canada
SELECT * FROM Customer
WHERE Country='Canada';


--4) Show orders placed in November 2023
SELECT * FROM Orders
WHERE Order_Date BETWEEN '2023-11-01' AND '2023-11-30';


--5) Retrieve the total stock of books available
SELECT SUM(Stock) AS total_stock
FROM Books;
```

--6) Find the details of the most expensive book

```
SELECT * FROM Books
ORDER BY Price DESC LIMIT '1';
```

--7) Show all customers who ordered more than 1 quantity of a book

```
SELECT * FROM Orders
WHERE quantity>1;
```

--8) Retrieve all orders where the total amount exceeds $20

```
SELECT * FROM Orders
WHERE total_amount>'20';
```

--9) List all genres available in the Books table

```
SELECT DISTINCT genre FROM Books;
```

--10) Find the book with the lowest stock

```
SELECT *FROM Books
ORDER BY stock ASC;
```

--11) Calculate the total revenue generated from all orders

```
SELECT SUM(Total_Amount) AS Total_Revenue
FROM Orders;
```

--Advance Queries

--1) Retrieve the total number of books sold for each genre

```
SELECT b.Genre, SUM (o.Quantity) AS Total_Books_sold
FROM Orders o
JOIN Books b ON o.Book_ID = b.Book_ID
GROUP BY b.Genre;
```

--2) Find the average price of books in the "Fantasy" genre

```
SELECT AVG(Price) AS Average_Price
FROM Books
WHERE Genre='Fantasy';
```

--3) List customers who have placed at least 2 orders

```
SELECT o.Customer_ID, COUNT(Order_id) AS Order_count
FROM Orders o
GROUP BY o.Customer_ID
HAVING COUNT(o.Order_ID) >= 2;
```

```
SELECT c.Customer_ID, c.Name
FROM Customers c
JOIN Orders o ON c.Customer_ID = o.Customer_ID
GROUP BY c.Customer_ID, c.Name
HAVING COUNT(o.Order_ID) >= 2;
```

--4) Find the most frequently ordered book

```
SELECT o.Book_ID, b.title, count (o.order_id) AS Order_Count
FROM orders o
JOIN Books b ON o.Book_ID=b.Book_ID
GROUP BY o.Book_ID, b.title
Order BY Order_Count DESC;
```

--5) Show the top 3 most expensive books of 'Fantasy' Genre

```
SELECT * FROM Books
WHERE Genre= 'Fantasy'
Order By Price DESC LIMIT '3';
```

--6) Retrieve the total quantity of books sold by each author

```
SELECT b.Author, SUM (o.quantity) AS Total_Books_Sold
FROM Orders o
JOIN Books b ON o.Book_id=b.Book_ID
GROUP BY b.Author;
```

--7) List the cities where customers who spent over $30 are locate

```
SELECT DISTINCT c.city, Total_amount
FROM Orders o
JOIN customer c ON o.customer_id=c.customer_id
WHERE o.total_amount>'30';
```

--8) Find the customer who spent the most on orders

```
SELECT c.customer_id, c.Name, SUM(o.total_amount) AS Total_spent
FROM orders o
JOIN customer c ON o.customer_id = c.customer_id
GROUP BY c.customer_id, c.Name
Order BY Total_spent DESC LIMIT '3';
```