SQL Project: Online Bookstore

Complete SQL Code

```
-- Create Database
CREATE DATABASE OnlineBookstore;

-- Switch to the database
\c OnlineBookstore;

-- Create Tables
DROP TABLE IF EXISTS Books;
CREATE TABLE Books (
    Book_ID SERIAL PRIMARY KEY,
    Title VARCHAR(100),
    Author VARCHAR(100),
    Genre VARCHAR(50),
    Published_Year INT,
    Price NUMERIC(10, 2),
    Stock INT
);
DROP TABLE IF EXISTS customers;
CREATE TABLE Customers (
    Customer_ID SERIAL PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Phone VARCHAR(15),
    City VARCHAR(50),
    Country VARCHAR(150)
);
DROP TABLE IF EXISTS orders;
CREATE TABLE Orders (
    Order_ID SERIAL PRIMARY KEY,
    Customer_ID INT REFERENCES Customers(Customer_ID),
    Book_ID INT REFERENCES Books(Book_ID),
    Order_Date DATE,
    Quantity INT,
    Total_Amount NUMERIC(10, 2)
);



-- Import Data into Books Table
COPY Books(Book_ID, Title, Author, Genre, Published_Year, Price, Stock)
FROM 'D:\Data analytics\SQL project\Books.csv'
CSV HEADER;

-- Import Data into Customers Table
COPY Customers(Customer_ID, Name, Email, Phone, City, Country)
FROM 'D:Course Updates\30 Day Series\SQL\CSV\Customers.csv'
CSV HEADER;

-- Import Data into Orders Table
```

SQL Project: Online Bookstore

```
COPY Orders(Order_ID, Customer_ID, Book_ID, Order_Date, Quantity, Total_Amount)
FROM 'D:\Course Updates\30 Day Series\SQL\CSV\Orders.csv'
CSV HEADER;


-- 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 customers
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)
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 c.name , o.quantity
from customers AS c
JOIN orders AS o
ON c.customer_id = o.customer_id
where o.quantity >1;

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


SELECT * FROM orders
WHERE total_amount > 20;
```

SQL Project: Online Bookstore

```
-- 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
limit 1;

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

select sum(total_amount)
from orders;


-- Advance Questions :

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

SELECT b.genre, sum(o.quantity) as total_quantity
from books as b
join orders as o
on b.book_id = o.book_id
group by b.genre;
-- 2) Find the average price of books in the "Fantasy" genre:

select avg(price) as avg_price
from books
where genre = 'Fantasy';


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

select c.name , o.customer_id, count(o.order_id) AS order_count
from customers AS c
JOIN orders AS o
ON c.customer_id = o.customer_id
group by o.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 as o
join books as b
on o.book_id = b.book_id
group by o.book_id,b.title
order by order_count desc limit 1;
```

SQL Project: Online Bookstore

```
-- 5) Show the top 3 most expensive books of 'Fantasy' Genre :
select genre , price
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_quantity
from books as b
join orders as o
on b.book_id = o.book_id
group by b.author;

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

select distinct c.city ,o.total_amount
from customers as c
join orders as o
on c.customer_id = o.customer_id
where o.total_amount >100;

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

select c.customer_id,c.name ,sum(o.total_amount) as total_spend
from customers as c
join orders as o
on c.customer_id = o.customer_id
group by c.customer_id , c.name
order by total_spend desc;

--9) Calculate the stock remaining after fulfilling all orders:
SELECT * FROM Books;
SELECT * FROM Customers;
SELECT * FROM Orders;

SELECT b.book_id , b.title , b.stock ,coalesce(sum(o.quantity),0)as order_quantity,
b.stock - coalesce(sum(o.quantity),0) as remaining_quantity
from books as b
left join orders as o
on b.book_id = o.book_id
group by b.book_id
order by b.book_id;
```

SQL Project: Online Bookstore