**📚 Online Bookstore SQL Project**

**🔹 Project Overview**

**This SQL project is based on an Online Bookstore dataset, where I performed various data analysis and querying tasks. The project was guided by Satish Dhawale, and instead of manually creating tables and databases, I imported a dataset and worked on it to generate meaningful insights.**

**Tools**

* **MYSQL**

**SCHEMA**

**The Star Schema is Designed to create a Relationship Between the Tables to Ensure Integrity and Facilitate Complex**

**Schema**

**Online\_Book\_Store**



Customers.CSV

Books.CSV

Orders.CSV

**In the next page, I’ve written step-by-step SQL script starting from basic level of queries to advance level of queries.**

**The first 11 queries are basic queries and the next 9 queries are advance level queries.**

**Basic Queries**

**1) Retrive all books in the 'Fiction' genre:**

**select \* from books**

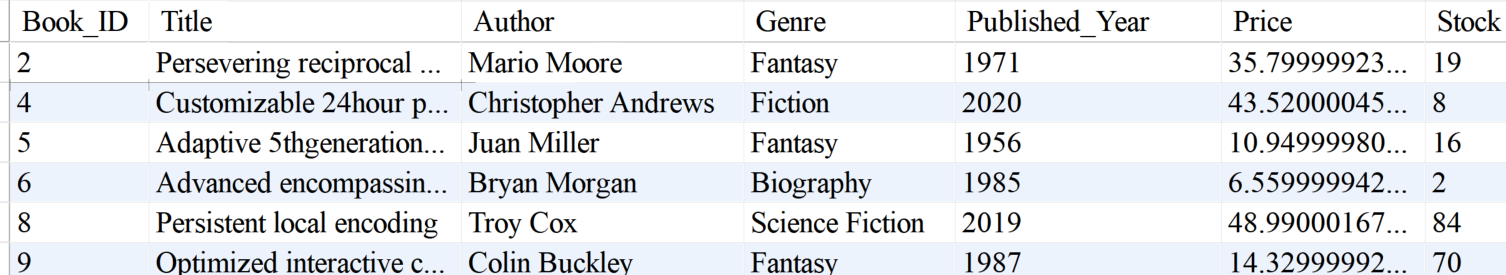
**where genre='Fiction';**

**result output: **

**2) Find books published after the year 1950**

**select \* from books**

**where Published\_Year>1950;**

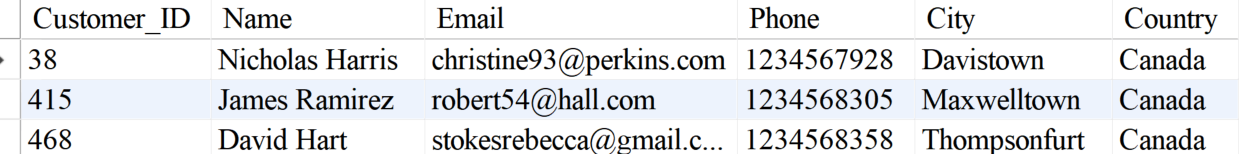
**Result Output:**

**3)List all customers from the Canada**

**select \* from customers**

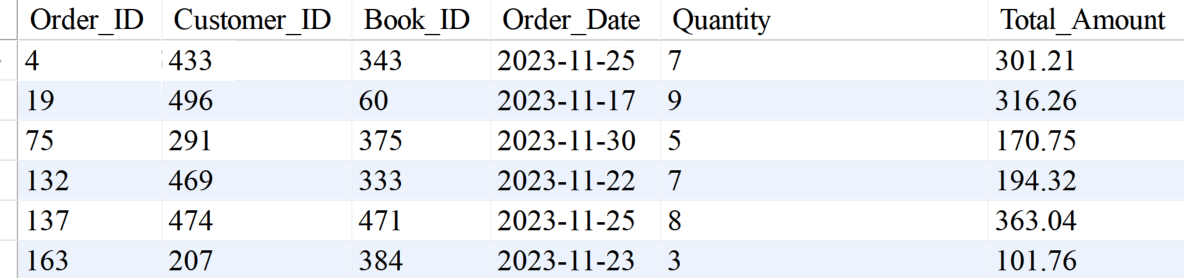
**where Country='Canada';**

**Result Output:**

****

**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**

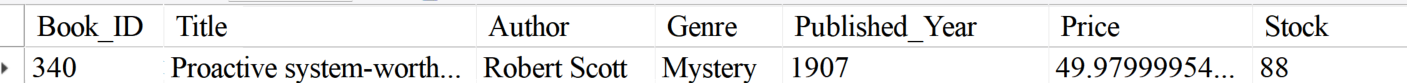
**select sum(stock)as Total\_Stock from books;**

**Result Output:**

****

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

**select \* from books order by Price desc**

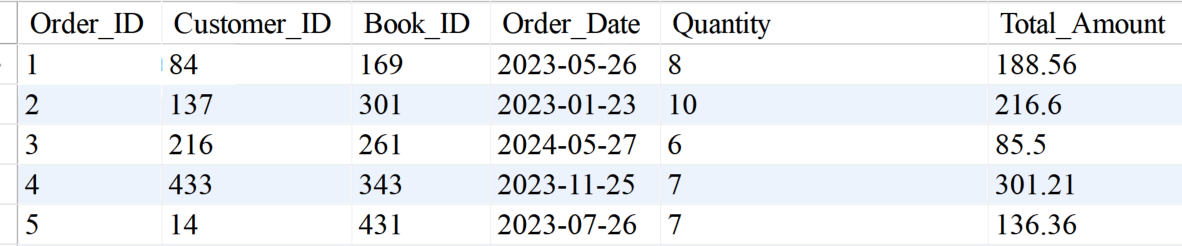
**limit 1;**

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

**select \* from orders**

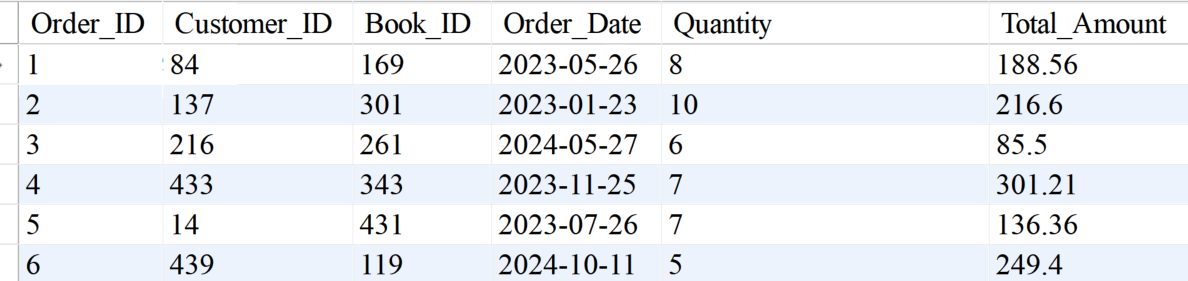
**where Quantity>1;**

**Result Output:**

****

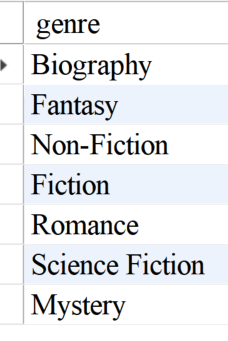
**8) Retrieve all orders where the total amount exceeds $20**

**select \* from orders**

**where Total\_Amount>20;**

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

**select distinct genre from books;**

**Result Output:  
**

**10) Find the book with the lowest stock**

**select \* from books**

**order by stock asc**

**limit 1;**

**Result Output:**

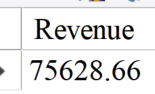
****

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

**select round(sum(Total\_Amount),2)as Revenue**

**from orders;**

**Result output:**

****

**Advanced Queries:**

**1) Retrieve the total number of books sold for each genres**

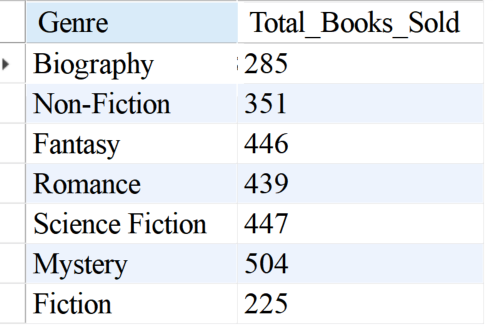
**select b.Genre, sum(o.Quantity) as Total\_Books\_Sold**

**from orders o**

**join books b on o.book\_id=b.book\_id**

**group by b.genre;**

**Result:**

****

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

**select round(avg(price),2)as Fantasy\_Avg\_Price**

**from books**

**where Genre='Fantasy';**

**Result:**

****

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

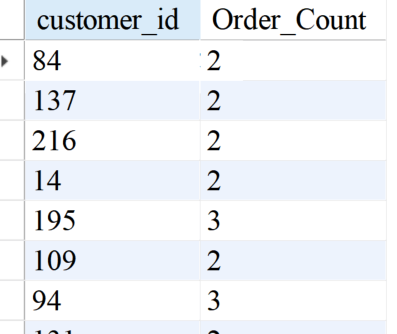
**select customer\_id, count(order\_id) as Order\_Count**

**from orders**

**group by Customer\_ID**

**having count(Order\_ID)>=2;**

**Result:**

****

**4) Find the most frequently ordered book**

**select o.book\_id, b.title, count(o.order\_id)as Order\_Count**

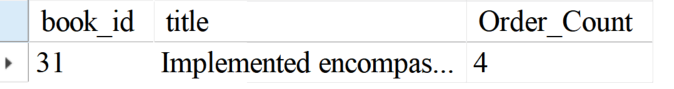
**from orders o**

**join books b on o.Book\_ID=b.Book\_ID**

**group by o.book\_id, b.Title**

**order by Order\_Count desc limit 1;**

**Result:**

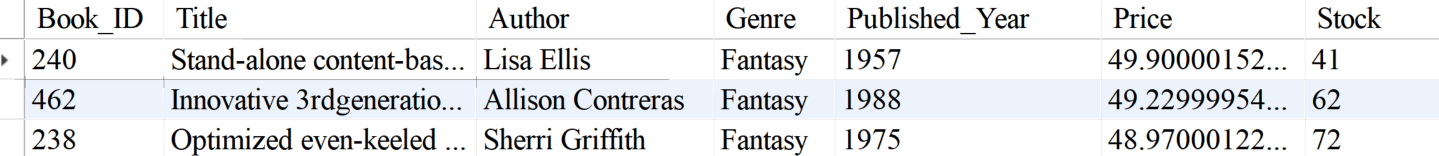
****

**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**

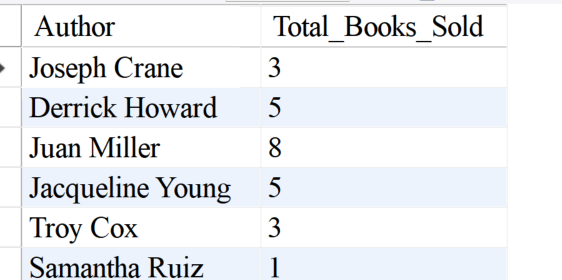
**select b.Author, sum(o.quantity)as Total\_Books\_Sold**

**from orders o**

**join books b on b.Book\_ID=o.book\_id**

**group by b.author;**

**Results:**

****

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

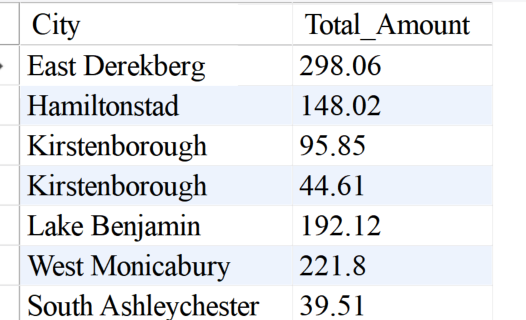
**select distinct c.City, o.Total\_Amount**

**from orders o**

**join customers c on o.Customer\_ID=c.Customer\_ID**

**where o.Total\_Amount>30;**

**Result:**

****

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

**select c.customer\_id, c.name, round(sum(o.total\_amount),2)as Total\_Spent**

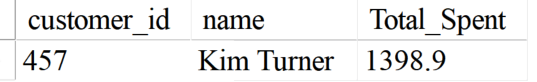
**from orders o**

**join customers c on o.Customer\_ID=c.Customer\_ID**

**group by c.Customer\_ID, c.Name**

**order by Total\_Spent desc limit 1;**

**Result:**

****

**9) Calculate the stock remaining after fulfilling all orders**

**select b.book\_id, b.title, b.stock, coalesce(sum(quantity),0)as Order\_Quantity,**

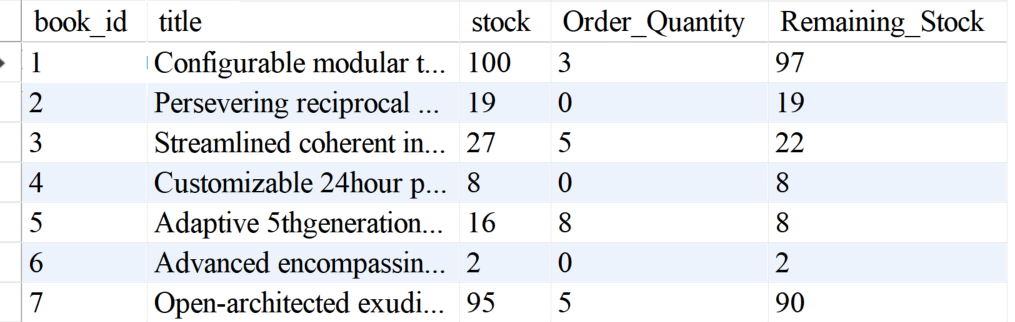
**b.Stock-coalesce(sum(quantity),0)as Remaining\_Stock**

**from books b**

**left join orders o on b.Book\_ID=o.Book\_ID**

**group by b.Book\_ID, b.Title,b.Stock;**

**Result:**

****

**#Without Coalesce And Using Only Join Function, So That It Will Print Only Matched And Non-Null Values. This Result Can Be Obtained Using These 2 Methods.**

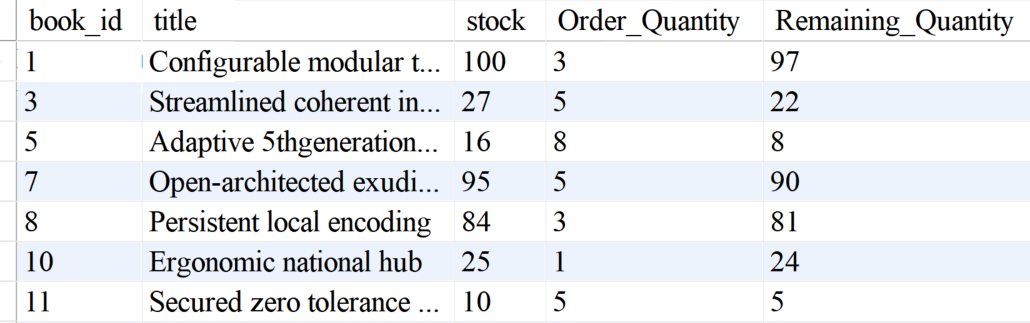
**select b.book\_id, b.title, b.stock, sum(o.quantity)as Order\_Quantity,**

**b.stock-sum(o.quantity)as Remaining\_Quantity**

**from books b**

**join orders o on b.Book\_ID=o.Book\_ID**

**group by b.Book\_ID, b.Title, b.Stock;**

**Result:  
**