ONLINE BOOKSTORE DATA ANALYSIS USING SQL

By P.Ramanjaneyulu

INTRODUCTION

This project focuses on analyzing data from an Online Bookstore using SQL.

1.Books.csv – Contains details about books such as title, genre, author, price, and stock availability.

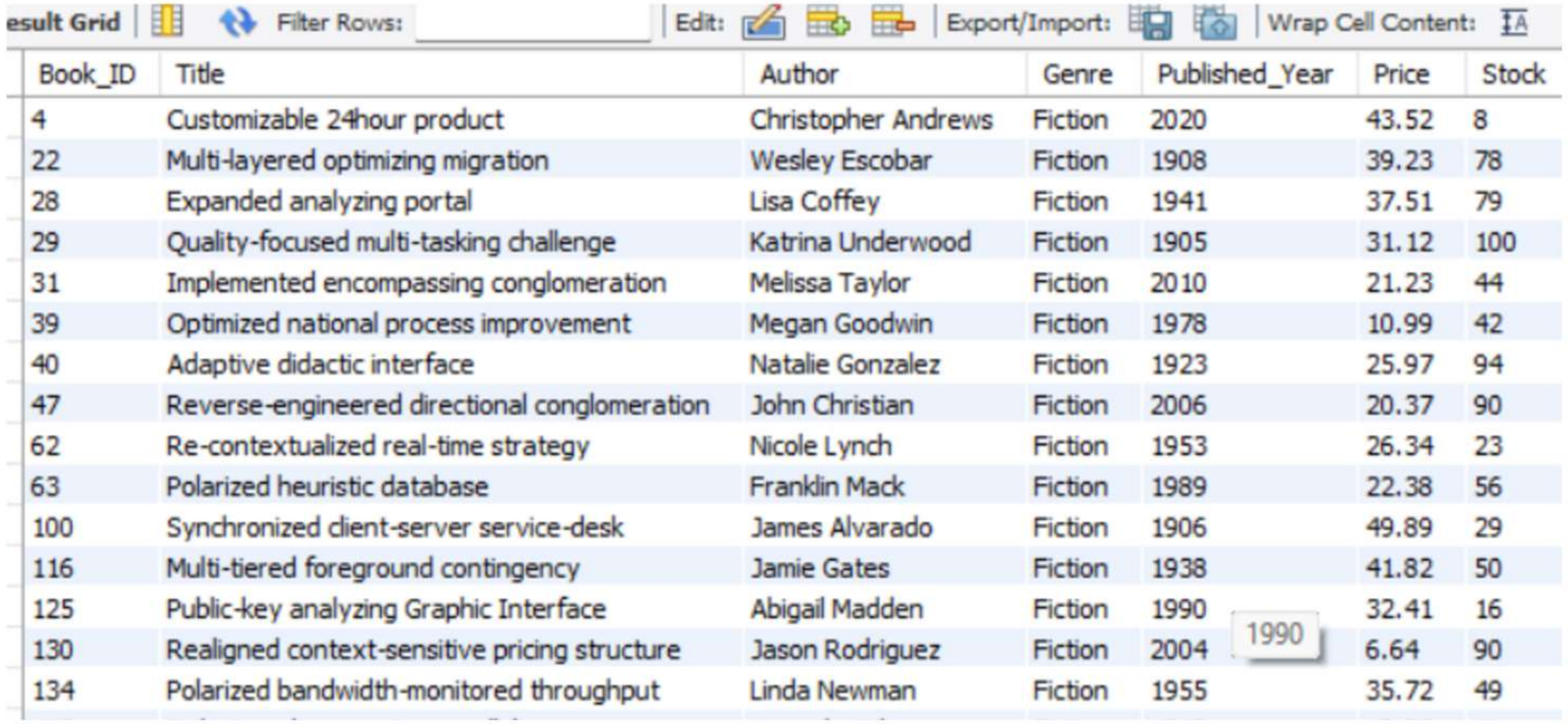
2.Customers.csv – Stores customer information, including Customer_ID, name, location, and purchasing patterns.

3.Orders.csv – Includes order details such as Order_ID, Customer_ID, Book_ID, quantity purchased, total price, and order date.

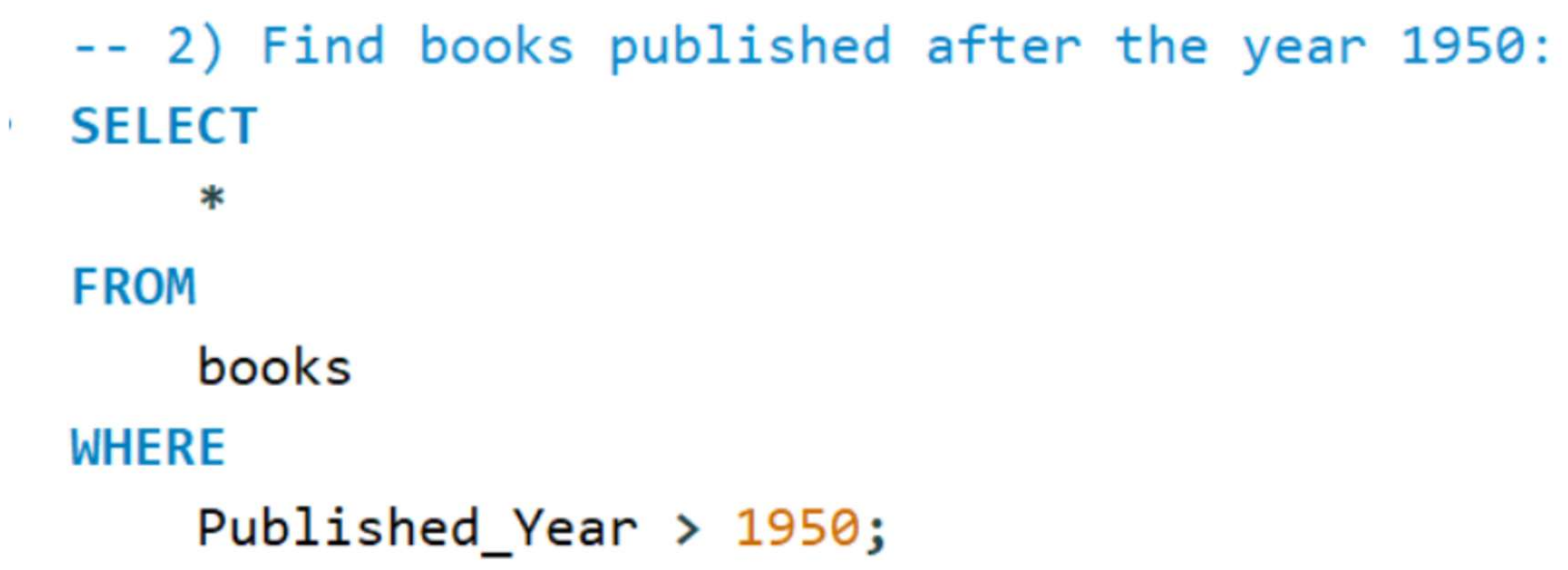
PROJECT OBJECTIVES:

- Book Inventory Analysis: Track available stock, identify bestsellers, and find books with low inventory.
- Customer Insights: Segment customers based on purchase behavior and total spending.
- Order Trends & Revenue Analysis: Analyze sales patterns, total revenue, and frequently ordered books.
- Advanced Data Insights: Identify top-selling genres, high-value customers, and stock management strategies.

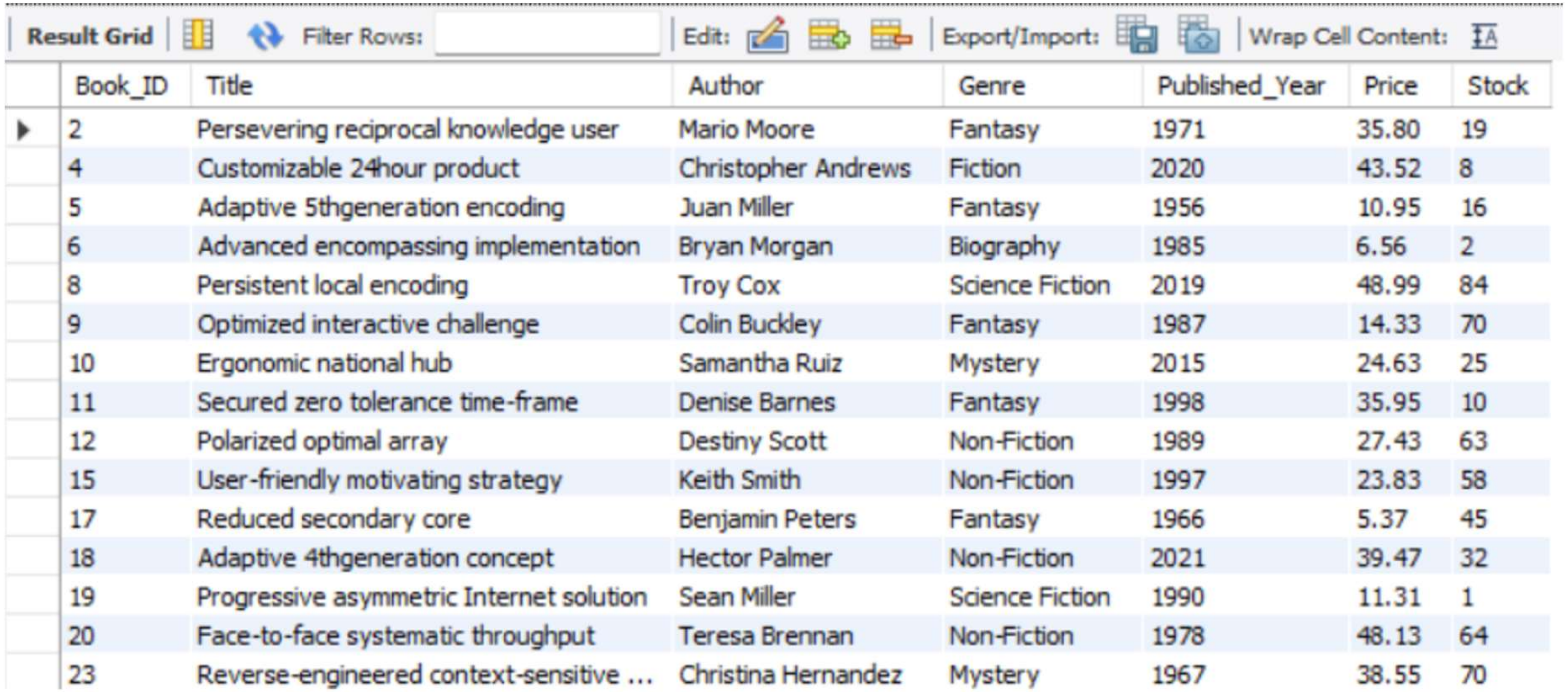
```
-- 1) Retrieve all books in the "Fiction" genre:
• SELECT
      *
  FROM
      books
  WHERE
      Genre = 'Fiction';
```



Book_ID	Title	Author	Genre	Published_Year	Price	Stock
4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
22	Multi-layered optimizing migration	Wesley Escobar	Fiction	1908	39.23	78
28	Expanded analyzing portal	Lisa Coffey	Fiction	1941	37.51	79
29	Quality-focused multi-tasking challenge	Katrina Underwood	Fiction	1905	31.12	100
31	Implemented encompassing conglomeration	Melissa Taylor	Fiction	2010	21.23	44
39	Optimized national process improvement	Megan Goodwin	Fiction	1978	10.99	42
40	Adaptive didactic interface	Natalie Gonzalez	Fiction	1923	25.97	94
47	Reverse-engineered directional conglomeration	John Christian	Fiction	2006	20.37	90
62	Re-contextualized real-time strategy	Nicole Lynch	Fiction	1953	26.34	23
63	Polarized heuristic database	Franklin Mack	Fiction	1989	22.38	56
100	Synchronized client-server service-desk	James Alvarado	Fiction	1906	49.89	29
116	Multi-tiered foreground contingency	Jamie Gates	Fiction	1938	41.82	50
125	Public-key analyzing Graphic Interface	Abigail Madden	Fiction	1990	32.41	16
130	Realigned context-sensitive pricing structure	Jason Rodriguez	Fiction	2004	6.64	90
134	Polarized bandwidth-monitored throughput	Linda Newman	Fiction	1955	35.72	49

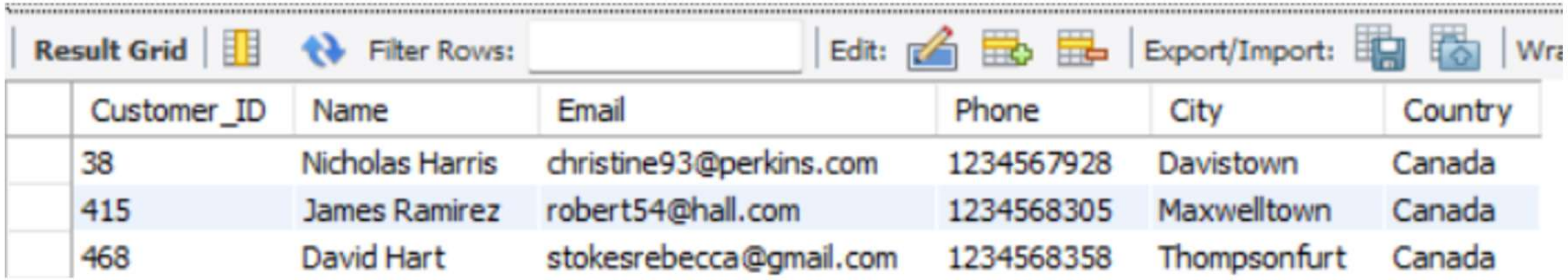


```
-- 2) Find books published after the year 1950:
SELECT
    *
FROM
    books
WHERE
    Published_Year > 1950;
```



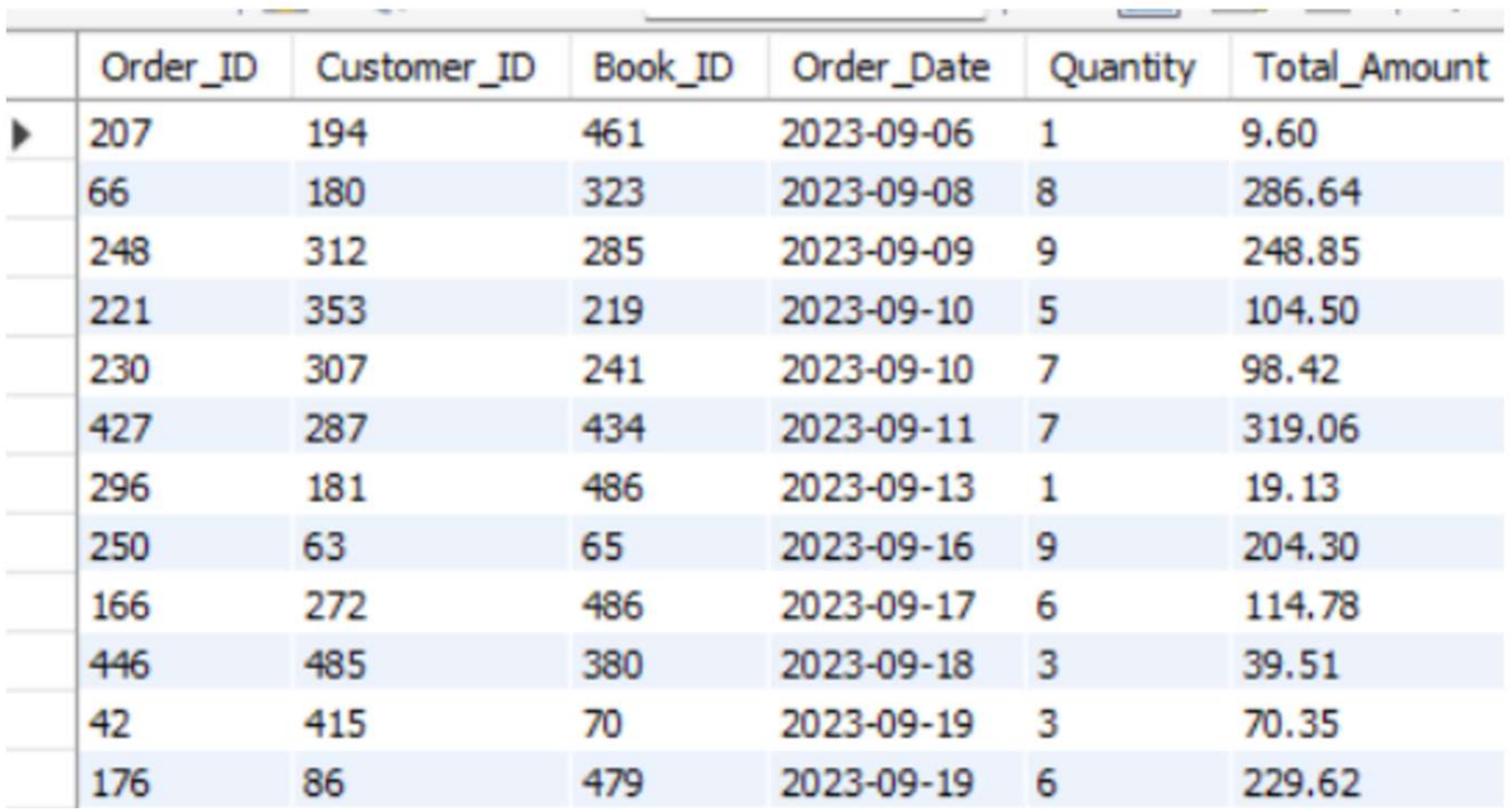
Book_ID	Title	Author	Genre	Published_Year	Price	Stock
2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19
4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	10.95	16
6	Advanced encompassing implementation	Bryan Morgan	Biography	1985	6.56	2
8	Persistent local encoding	Troy Cox	Science Fiction	2019	48.99	84
9	Optimized interactive challenge	Colin Buckley	Fantasy	1987	14.33	70
10	Ergonomic national hub	Samantha Ruiz	Mystery	2015	24.63	25
11	Secured zero tolerance time-frame	Denise Barnes	Fantasy	1998	35.95	10
12	Polarized optimal array	Destiny Scott	Non-Fiction	1989	27.43	63
15	User-friendly motivating strategy	Keith Smith	Non-Fiction	1997	23.83	58
17	Reduced secondary core	Benjamin Peters	Fantasy	1966	5.37	45
18	Adaptive 4thgeneration concept	Hector Palmer	Non-Fiction	2021	39.47	32
19	Progressive asymmetric Internet solution	Sean Miller	Science Fiction	1990	11.31	1
20	Face-to-face systematic throughput	Teresa Brennan	Non-Fiction	1978	48.13	64
23	Reverse-engineered context-sensitive ...	Christina Hernandez	Mystery	1967	38.55	70

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

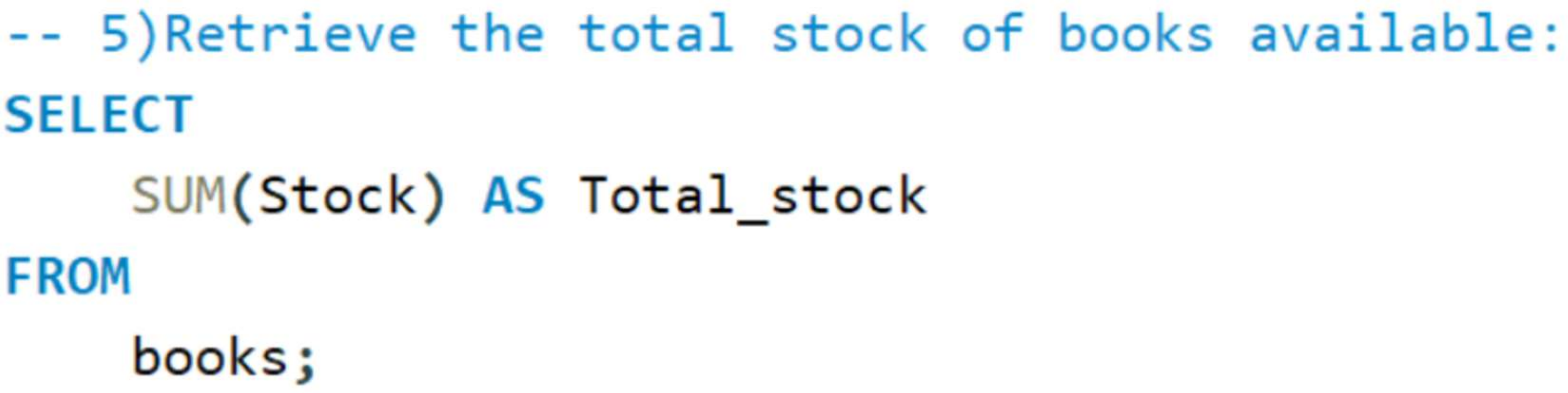


Customer_ID	Name	Email	Phone	City	Country
38	Nicholas Harris	christine93@perkins.com	1234567928	Davistown	Canada
415	James Ramirez	robert54@hall.com	1234568305	Maxwelltown	Canada
468	David Hart	stokesrebecca@gmail.com	1234568358	Thompsonfurt	Canada

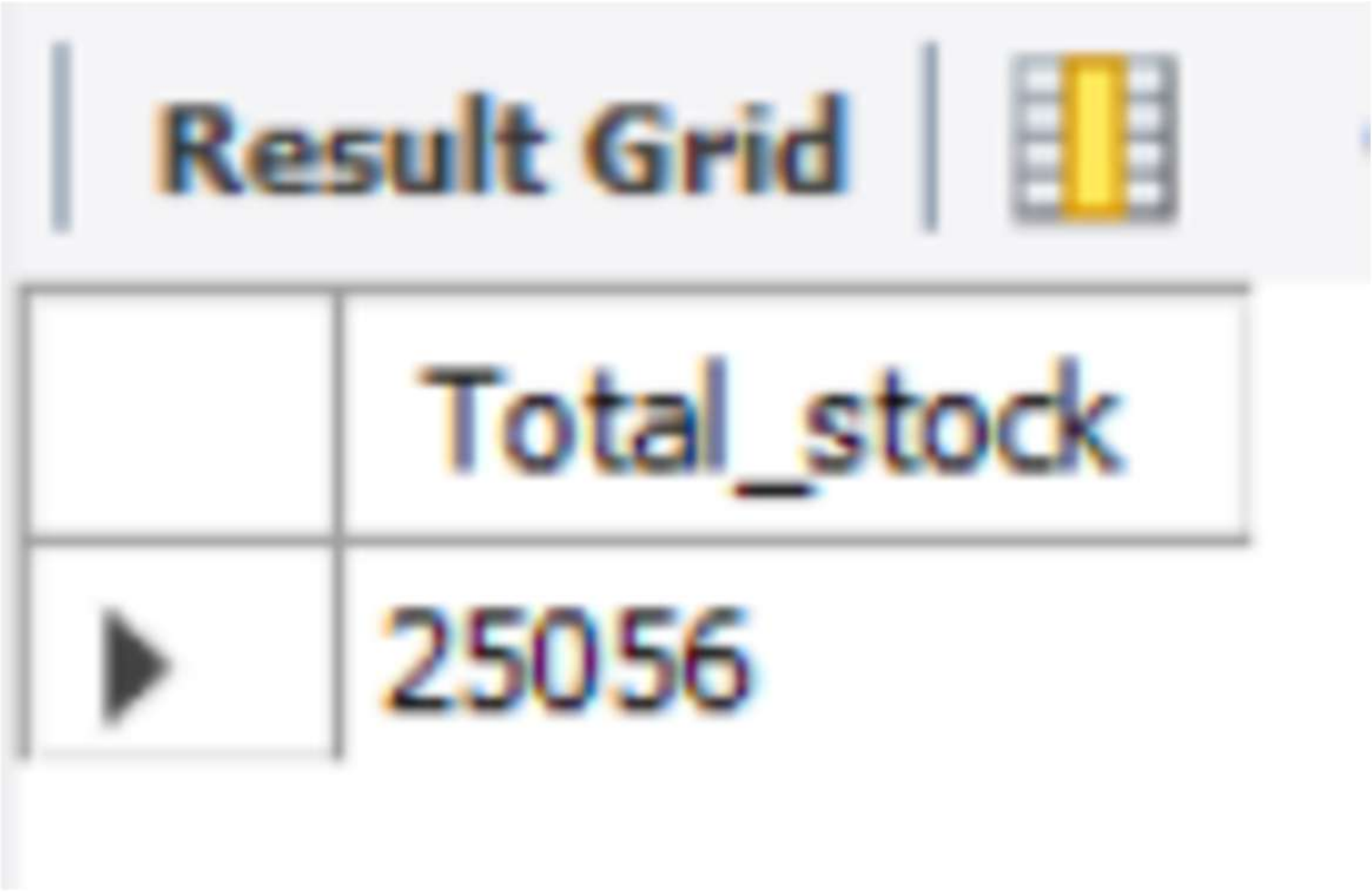
```
-- 4) Show orders placed in November 2023:

SELECT
    *
FROM
    orders
WHERE
    Order_Date BETWEEN '2023-09-01' AND '2023-09-30'
ORDER BY Order_Date;
```



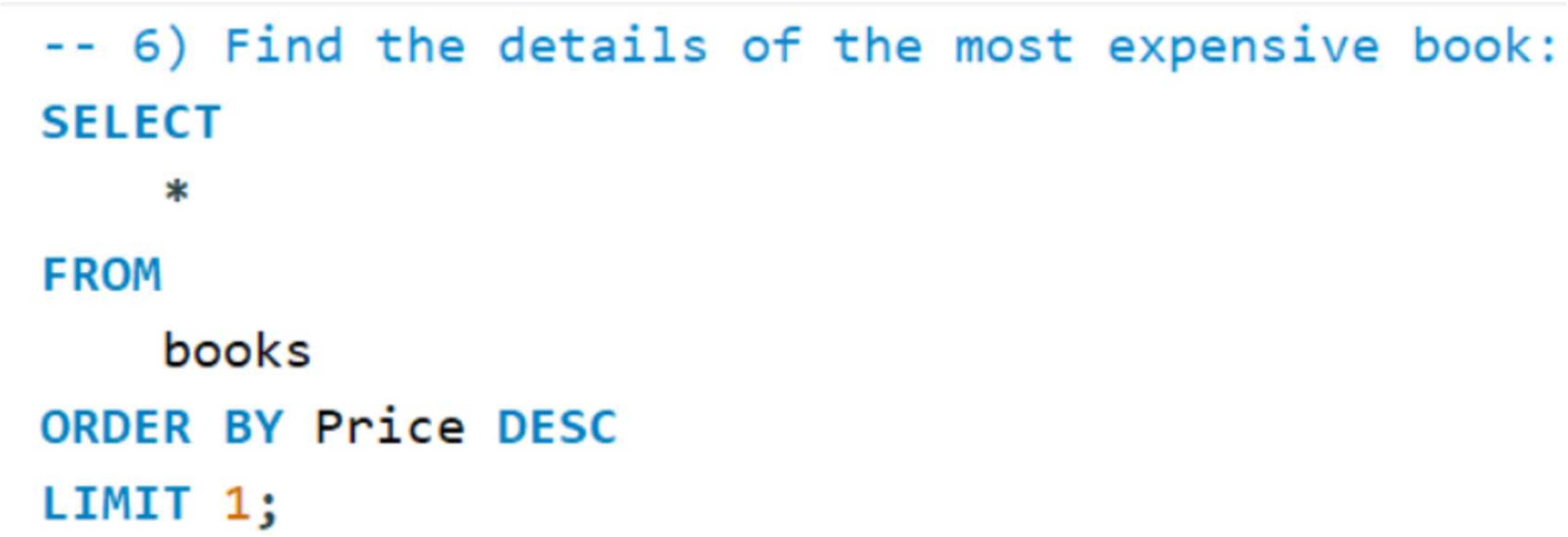
Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
207	194	461	2023-09-06	1	9.60
66	180	323	2023-09-08	8	286.64
248	312	285	2023-09-09	9	248.85
221	353	219	2023-09-10	5	104.50
230	307	241	2023-09-10	7	98.42
427	287	434	2023-09-11	7	319.06
296	181	486	2023-09-13	1	19.13
250	63	65	2023-09-16	9	204.30
166	272	486	2023-09-17	6	114.78
446	485	380	2023-09-18	3	39.51
42	415	70	2023-09-19	3	70.35
176	86	479	2023-09-19	6	229.62



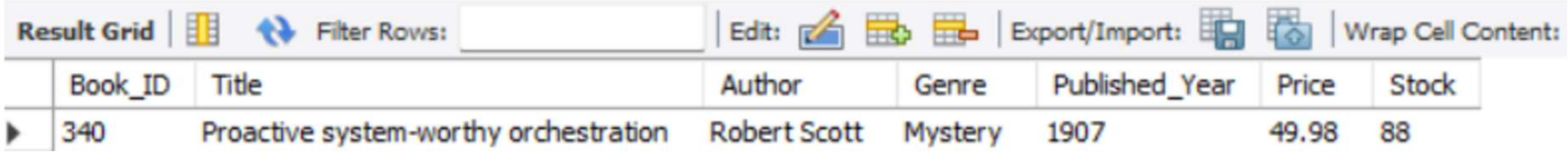
```sql
-- 5)Retrieve the total stock of books available:
SELECT
    SUM(Stock) AS Total_stock
FROM
    books;
```



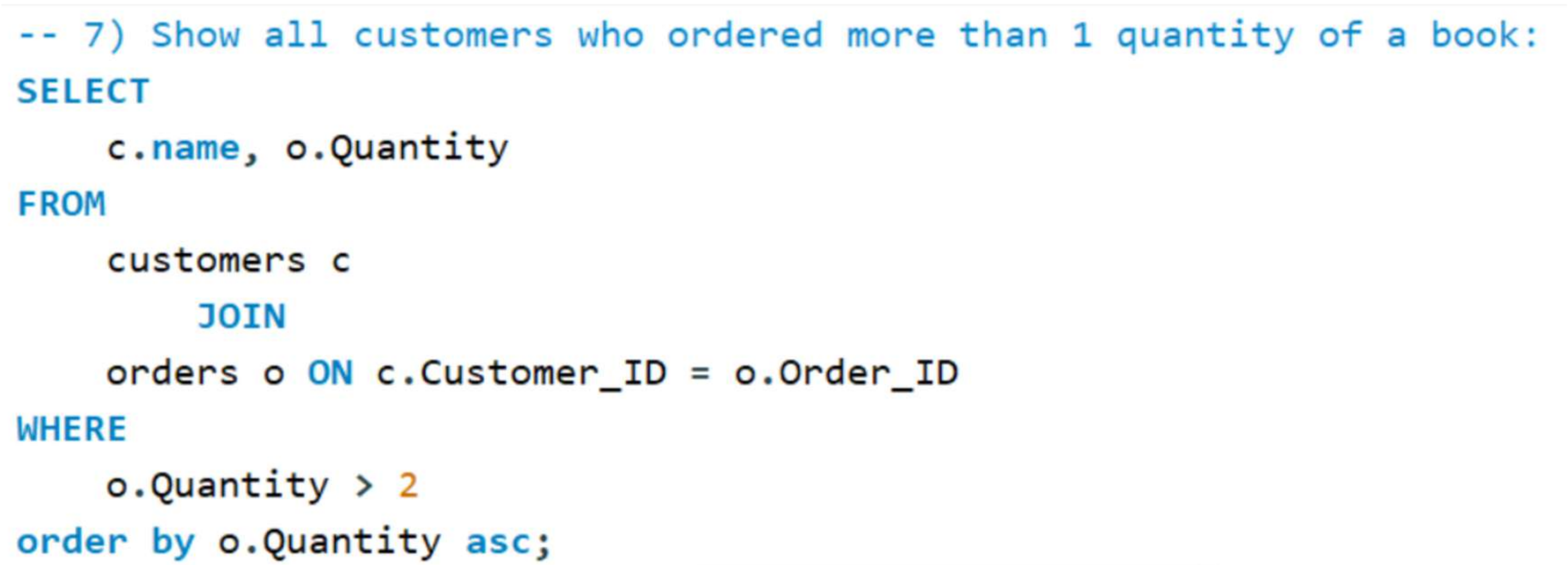
	Total_stock
▶	25056



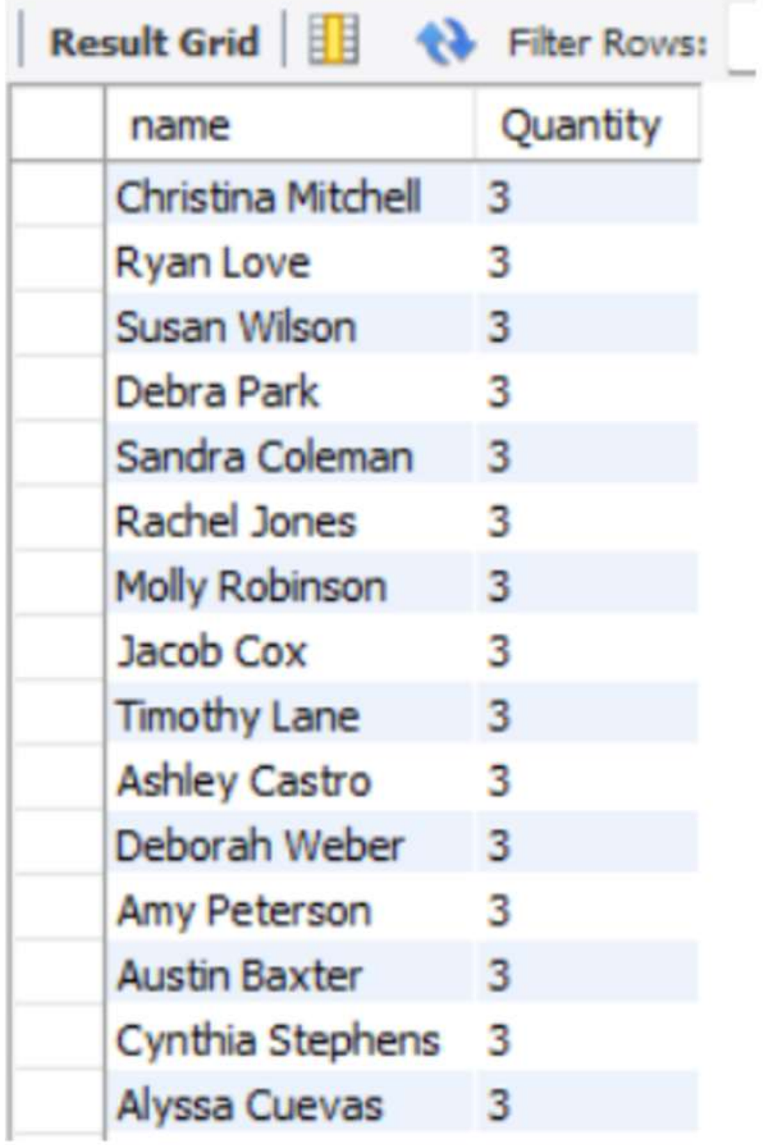
```sql
-- 6) Find the details of the most expensive book:
SELECT
    *
FROM
    books
ORDER BY Price DESC
LIMIT 1;
```



Book_ID	Title	Author	Genre	Published_Year	Price	Stock
340	Proactive system-worthy orchestration	Robert Scott	Mystery	1907	49.98	88

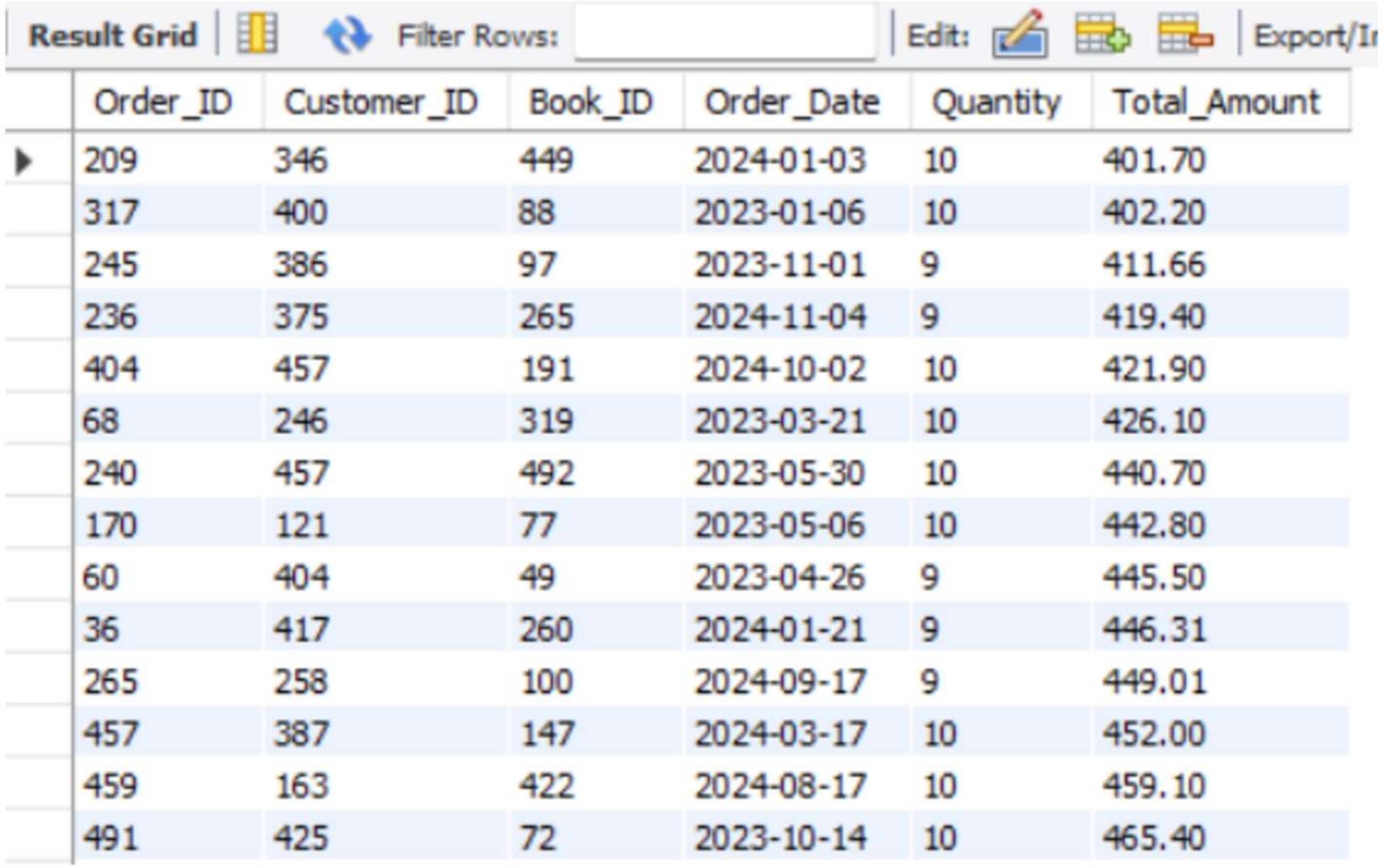


```
-- 7) Show all customers who ordered more than 1 quantity of a book:
SELECT
    c.name, o.Quantity
FROM
    customers c
        JOIN
    orders o ON c.Customer_ID = o.Order_ID
WHERE
    o.Quantity > 2
order by o.Quantity asc;
```

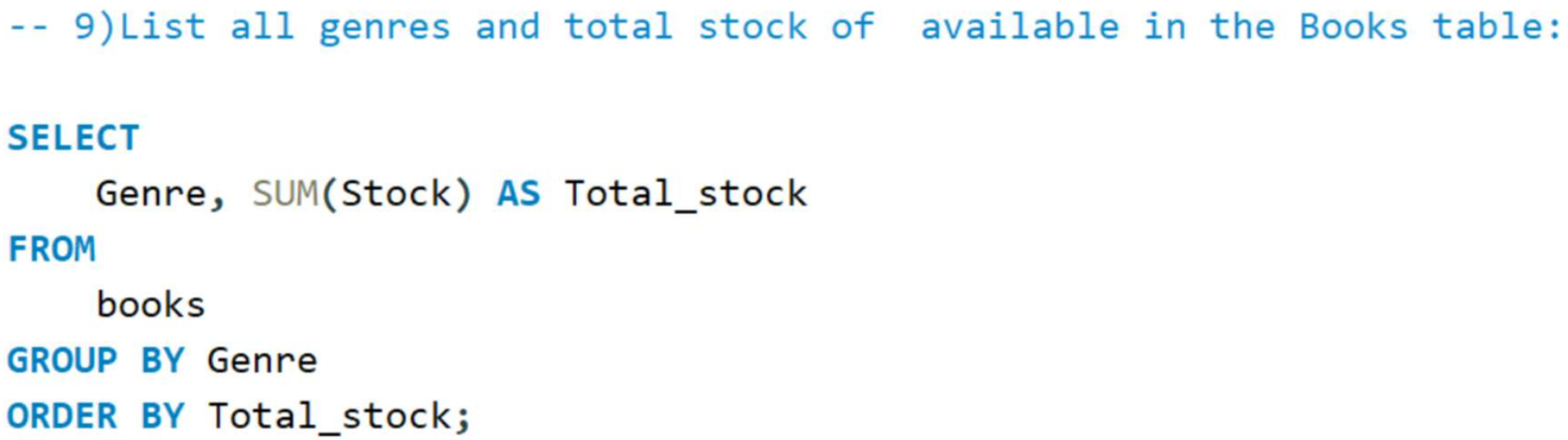


name	Quantity
Christina Mitchell	3
Ryan Love	3
Susan Wilson	3
Debra Park	3
Sandra Coleman	3
Rachel Jones	3
Molly Robinson	3
Jacob Cox	3
Timothy Lane	3
Ashley Castro	3
Deborah Weber	3
Amy Peterson	3
Austin Baxter	3
Cynthia Stephens	3
Alyssa Cuevas	3

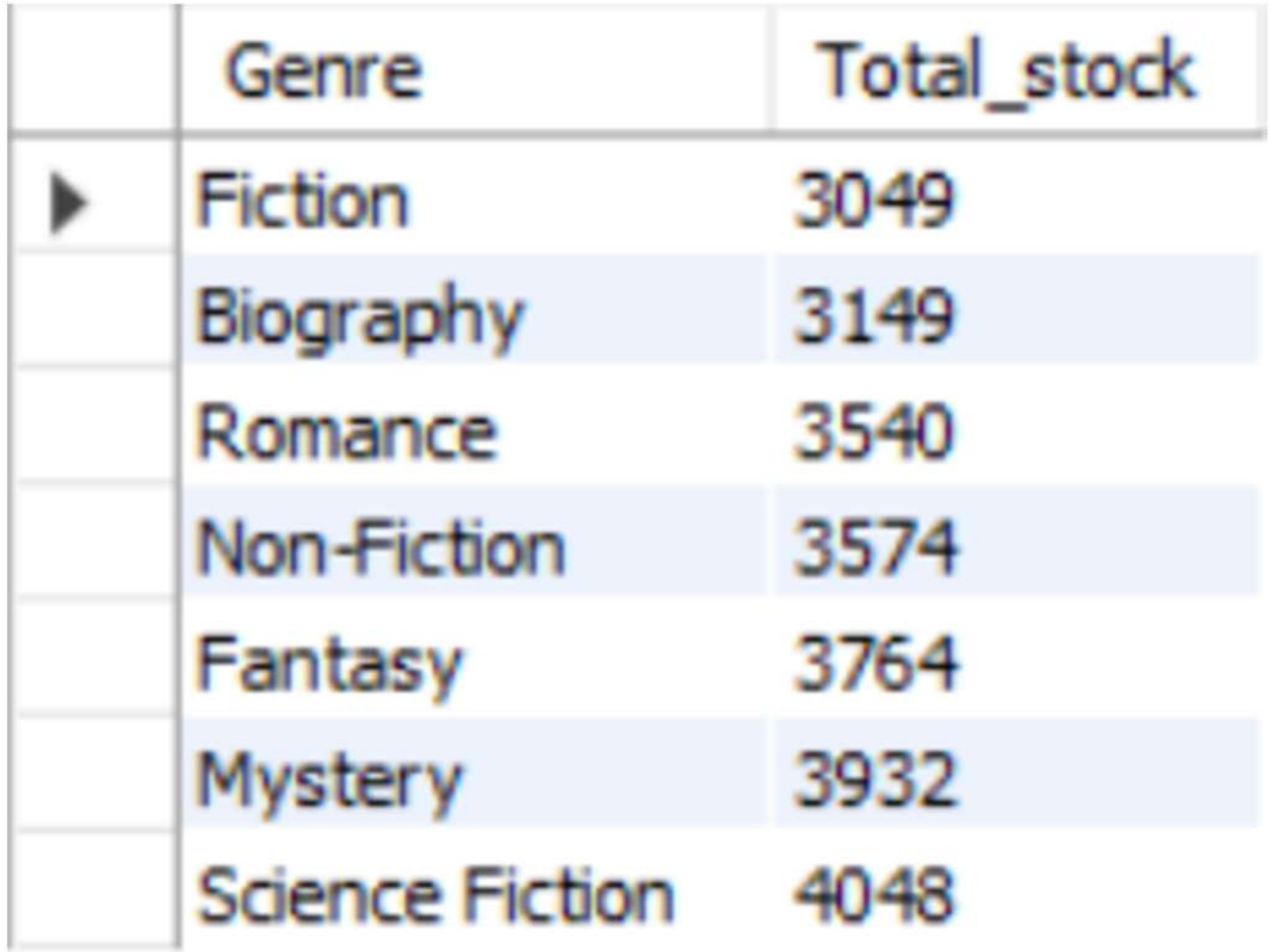
```sql
-- 8) Retrieve all orders where the total amount exceeds $400:
SELECT
    *
FROM
    orders
WHERE
    Total_Amount > 400
order by Total_Amount ;
```



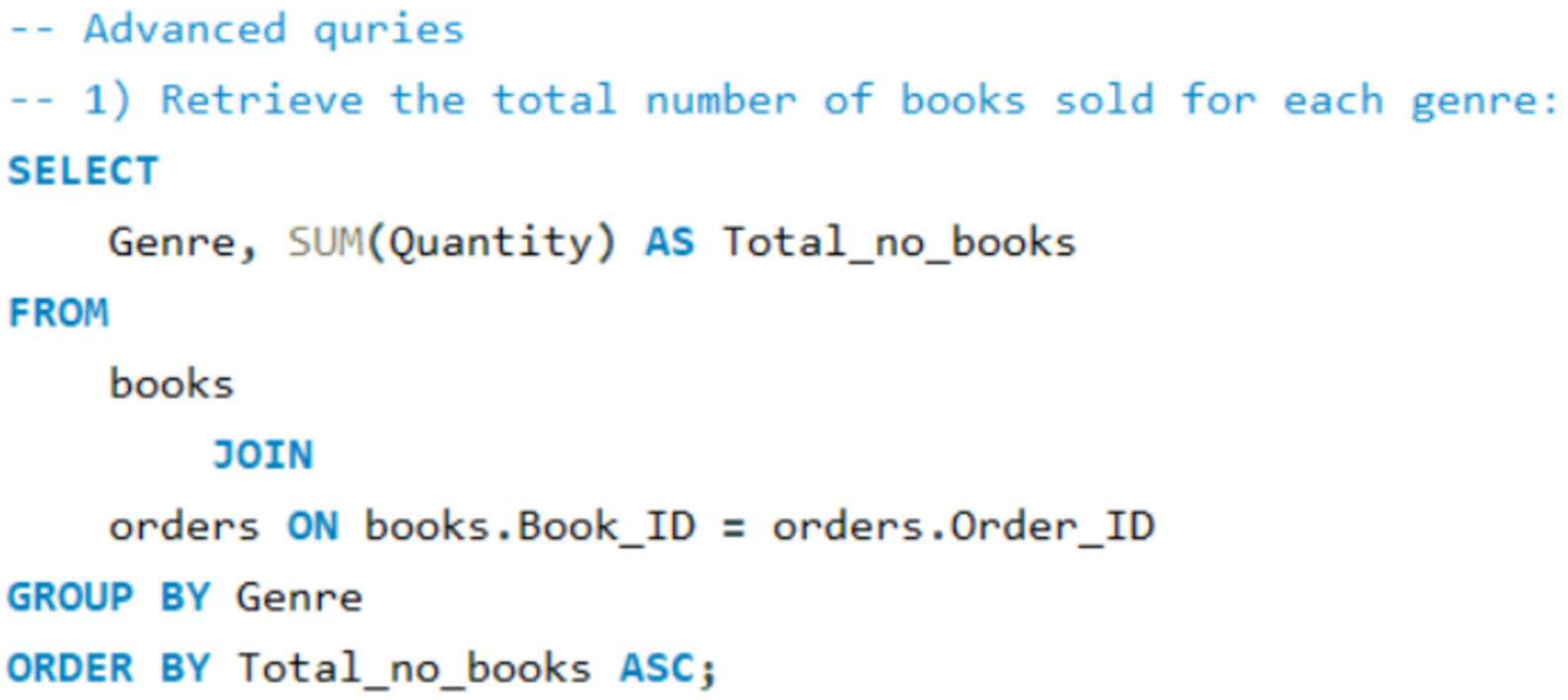
Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
209	346	449	2024-01-03	10	401.70
317	400	88	2023-01-06	10	402.20
245	386	97	2023-11-01	9	411.66
236	375	265	2024-11-04	9	419.40
404	457	191	2024-10-02	10	421.90
68	246	319	2023-03-21	10	426.10
240	457	492	2023-05-30	10	440.70
170	121	77	2023-05-06	10	442.80
60	404	49	2023-04-26	9	445.50
36	417	260	2024-01-21	9	446.31
265	258	100	2024-09-17	9	449.01
457	387	147	2024-03-17	10	452.00
459	163	422	2024-08-17	10	459.10
491	425	72	2023-10-14	10	465.40



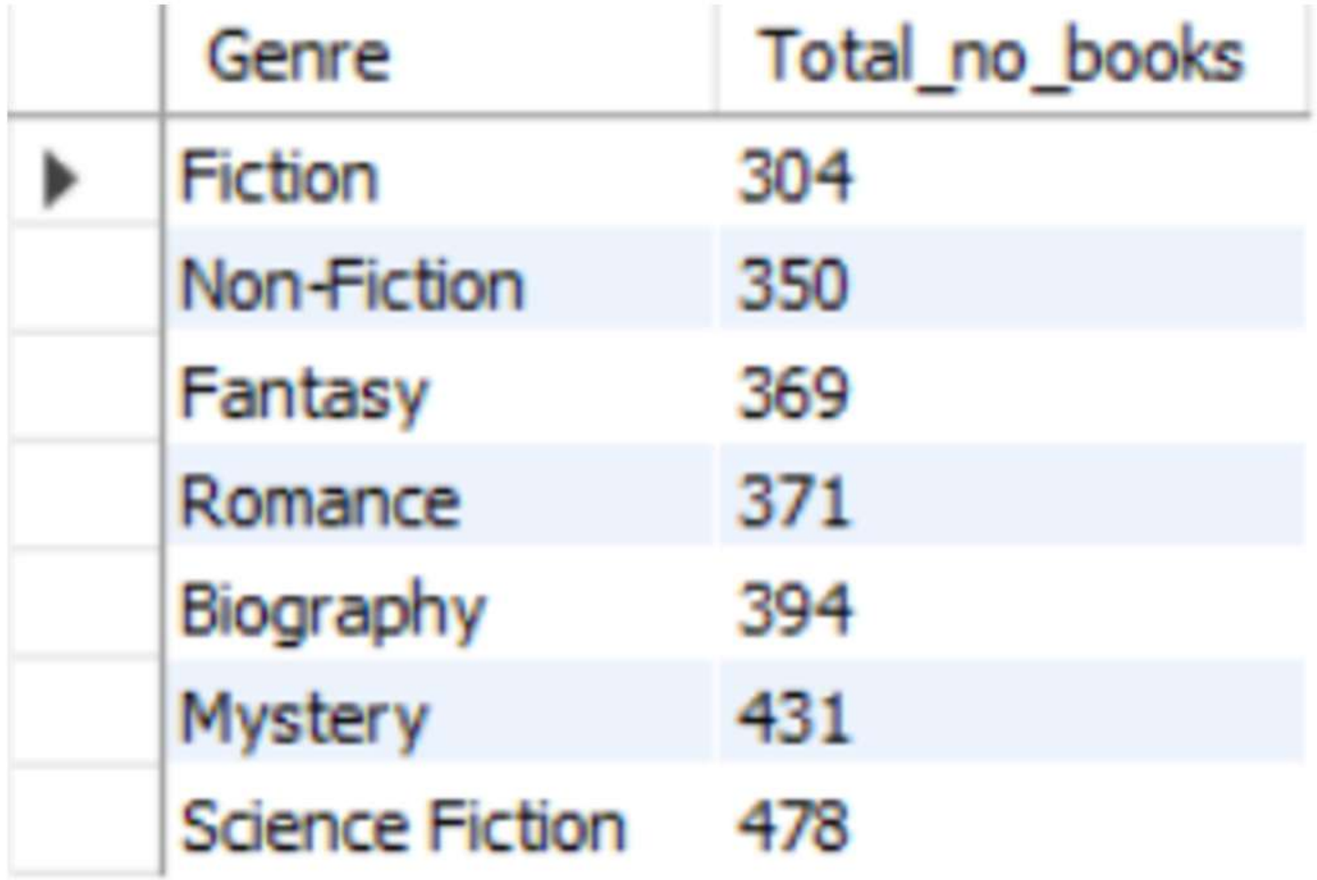
```
-- 9)List all genres and total stock of  available in the Books table:

SELECT
    Genre, SUM(Stock) AS Total_stock
FROM
    books
GROUP BY Genre
ORDER BY Total_stock;
```



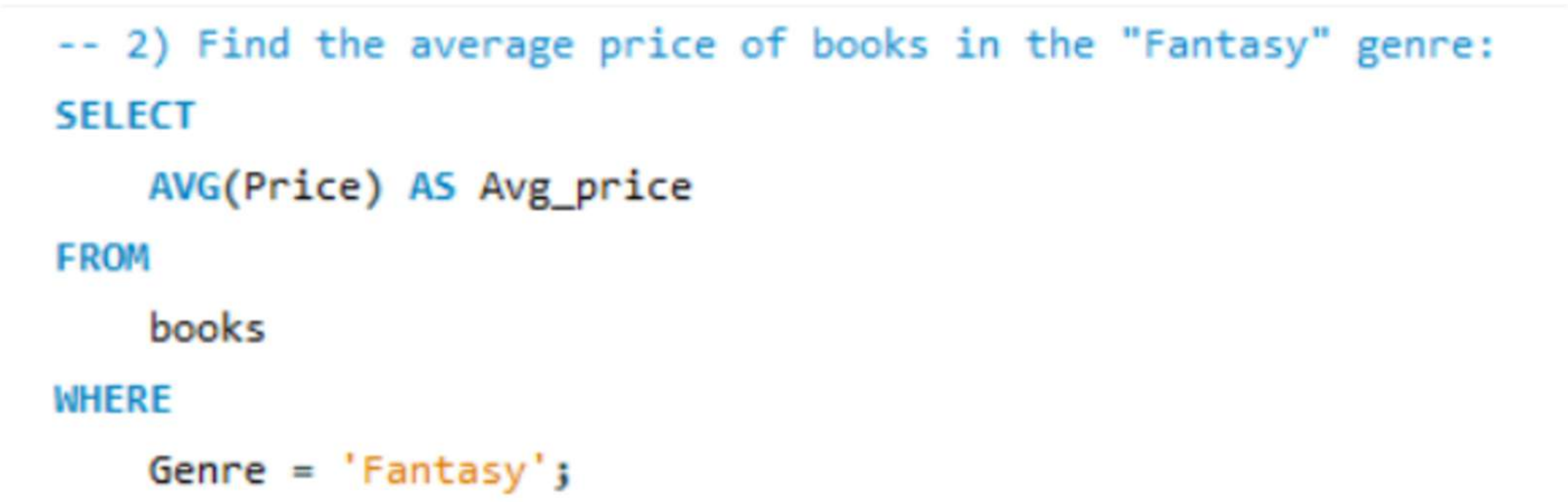
	Genre	Total_stock
▶	Fiction	3049
	Biography	3149
	Romance	3540
	Non-Fiction	3574
	Fantasy	3764
	Mystery	3932
	Science Fiction	4048



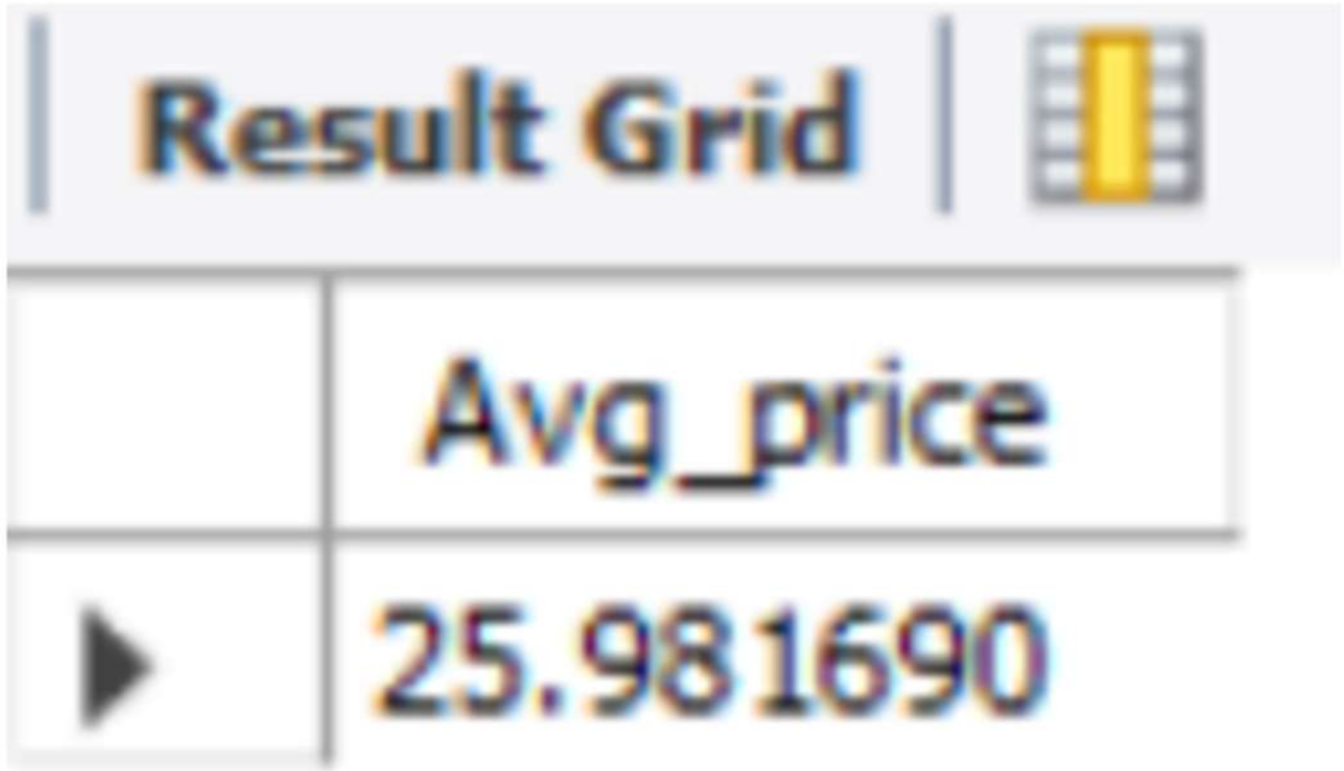
```sql
-- Advanced quries
-- 1) Retrieve the total number of books sold for each genre:
SELECT
    Genre, SUM(Quantity) AS Total_no_books
FROM
    books
        JOIN
    orders ON books.Book_ID = orders.Order_ID
GROUP BY Genre
ORDER BY Total_no_books ASC;
```



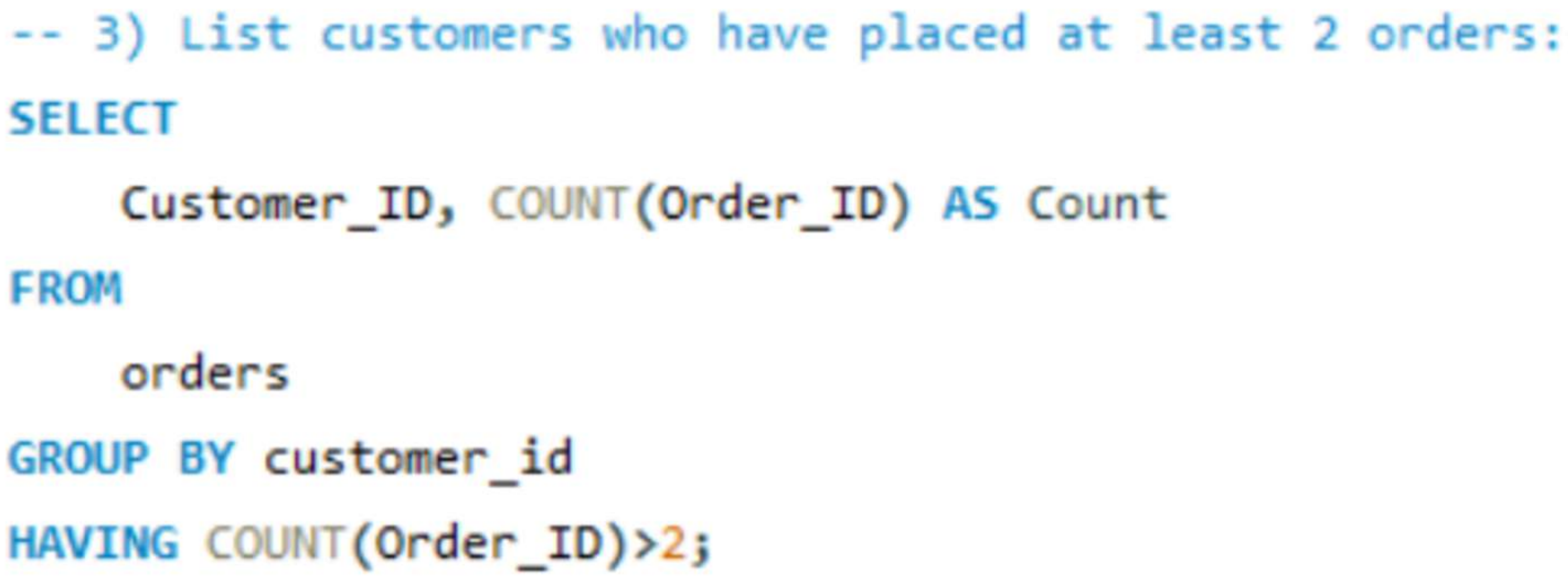
Genre	Total_no_books
Fiction	304
Non-Fiction	350
Fantasy	369
Romance	371
Biography	394
Mystery	431
Science Fiction	478



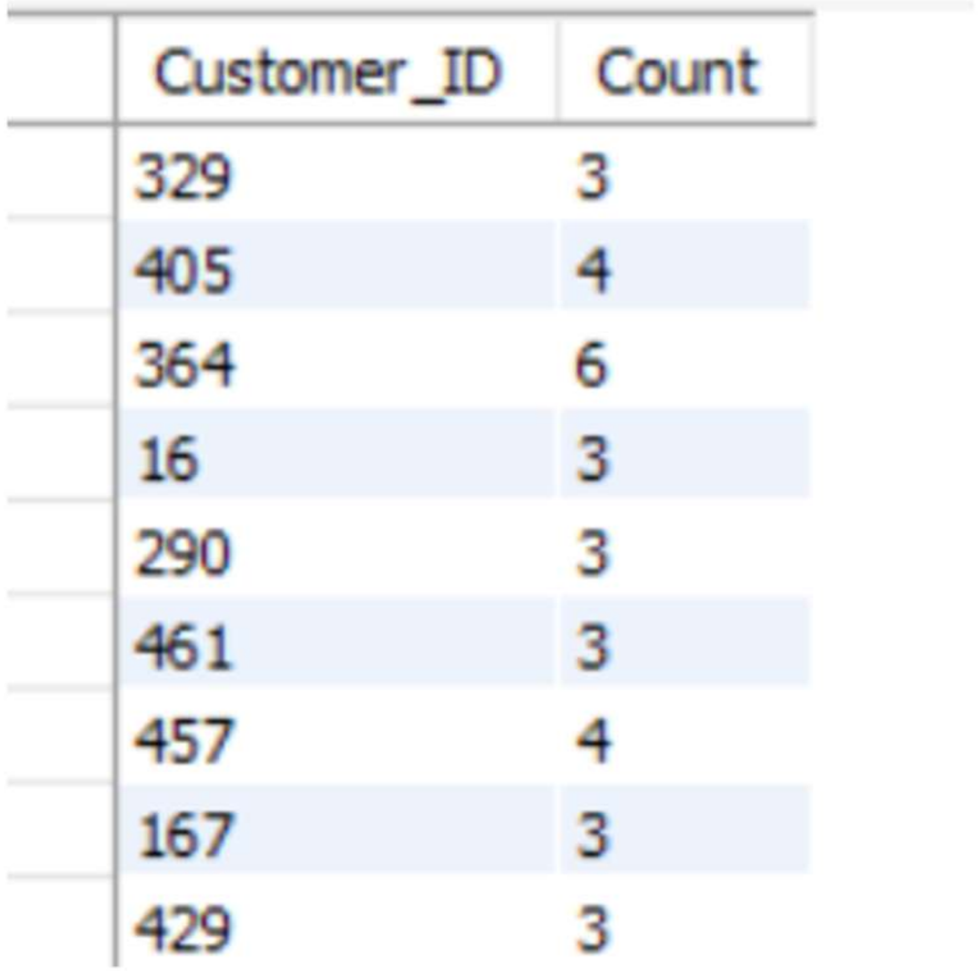
```
-- 2) Find the average price of books in the "Fantasy" genre:
SELECT
    AVG(Price) AS Avg_price
FROM
    books
WHERE
    Genre = 'Fantasy';
```



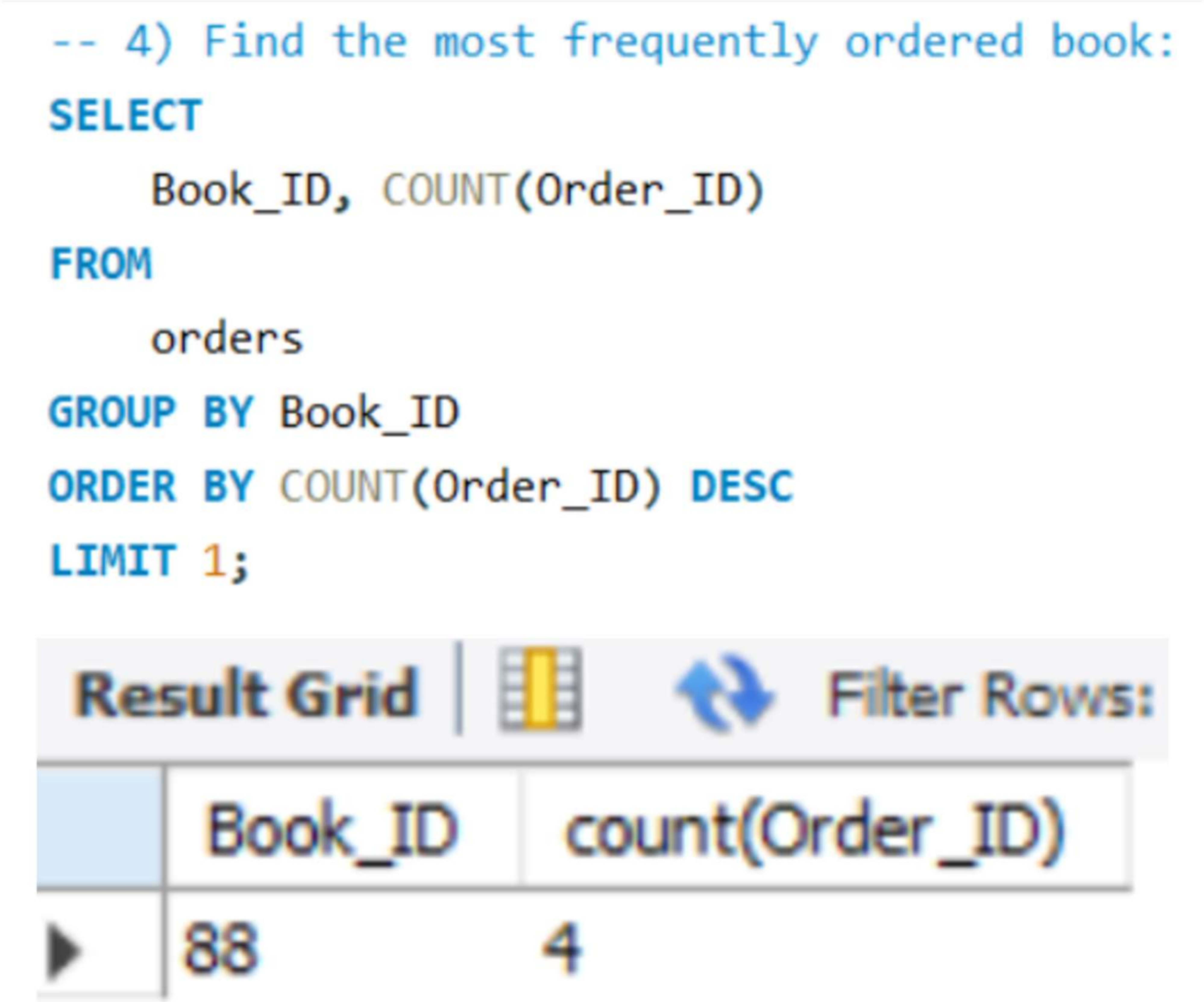
	Avg_price
▶	25.981690



```
-- 3) List customers who have placed at least 2 orders:
SELECT
    Customer_ID, COUNT(Order_ID) AS Count
FROM
    orders
GROUP BY customer_id
HAVING COUNT(Order_ID)>2;
```

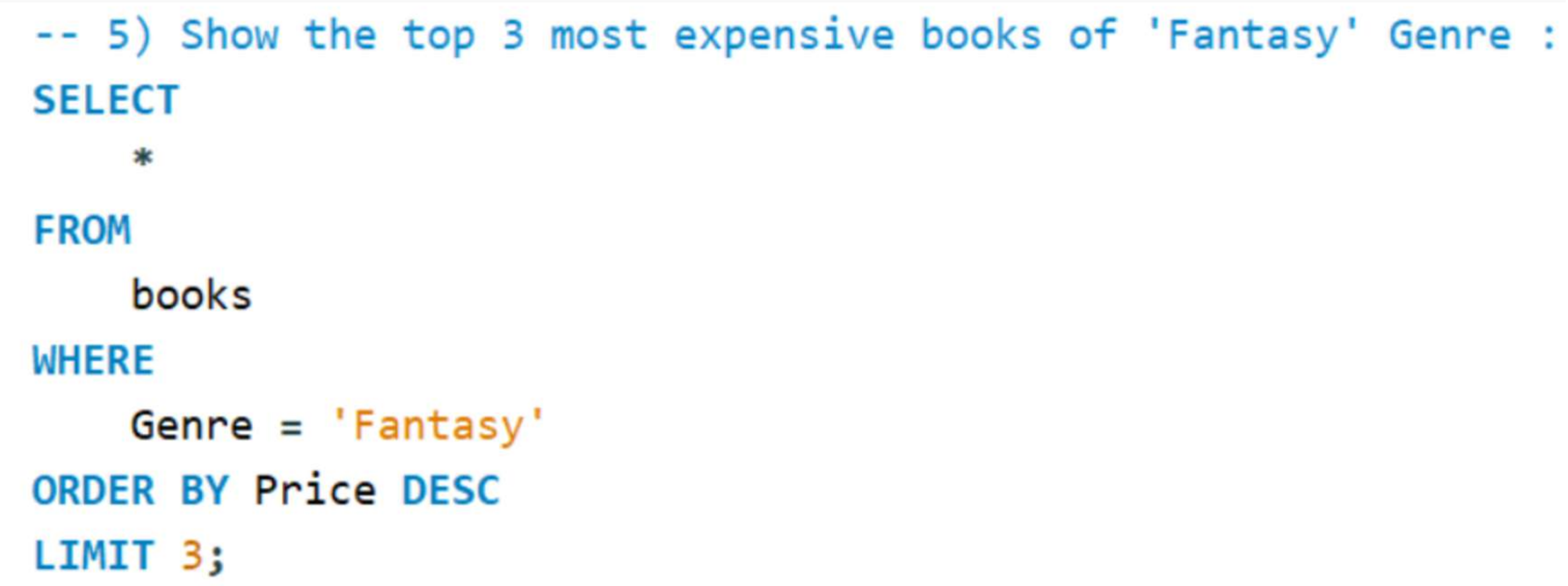


Customer_ID	Count
329	3
405	4
364	6
16	3
290	3
461	3
457	4
167	3
429	3

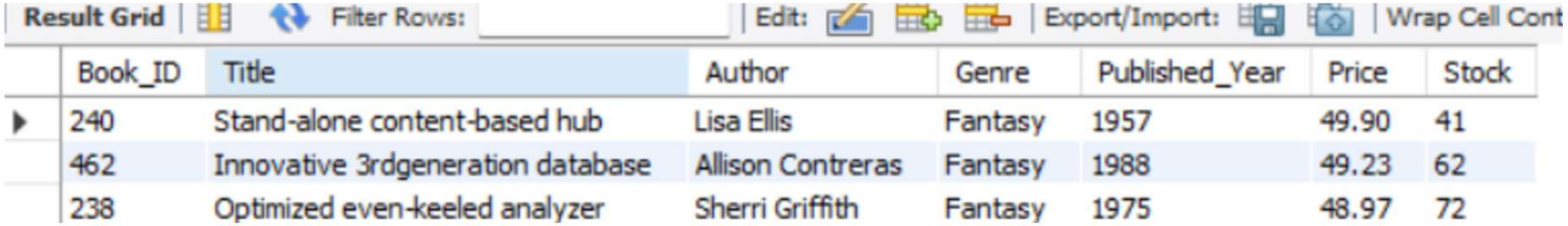


```
-- 4) Find the most frequently ordered book:
SELECT
    Book_ID, COUNT(Order_ID)
FROM
    orders
GROUP BY Book_ID
ORDER BY COUNT(Order_ID) DESC
LIMIT 1;
```

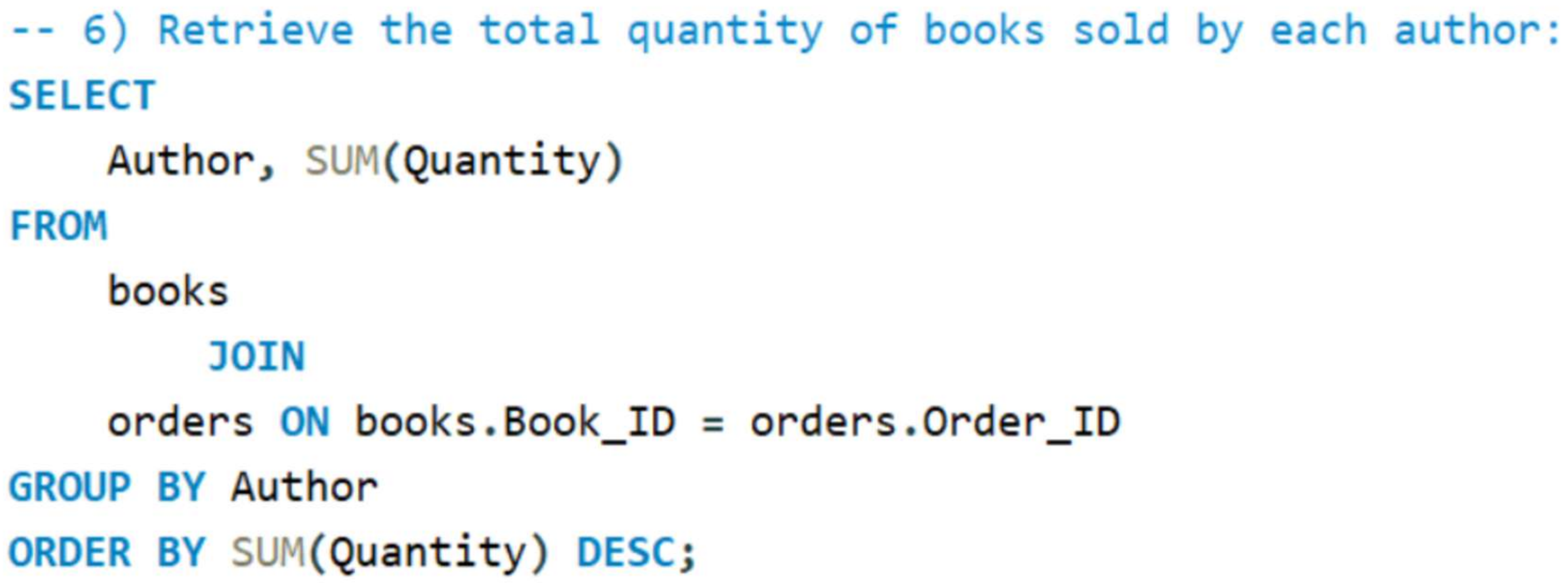
Book_ID	count(Order_ID)
88	4



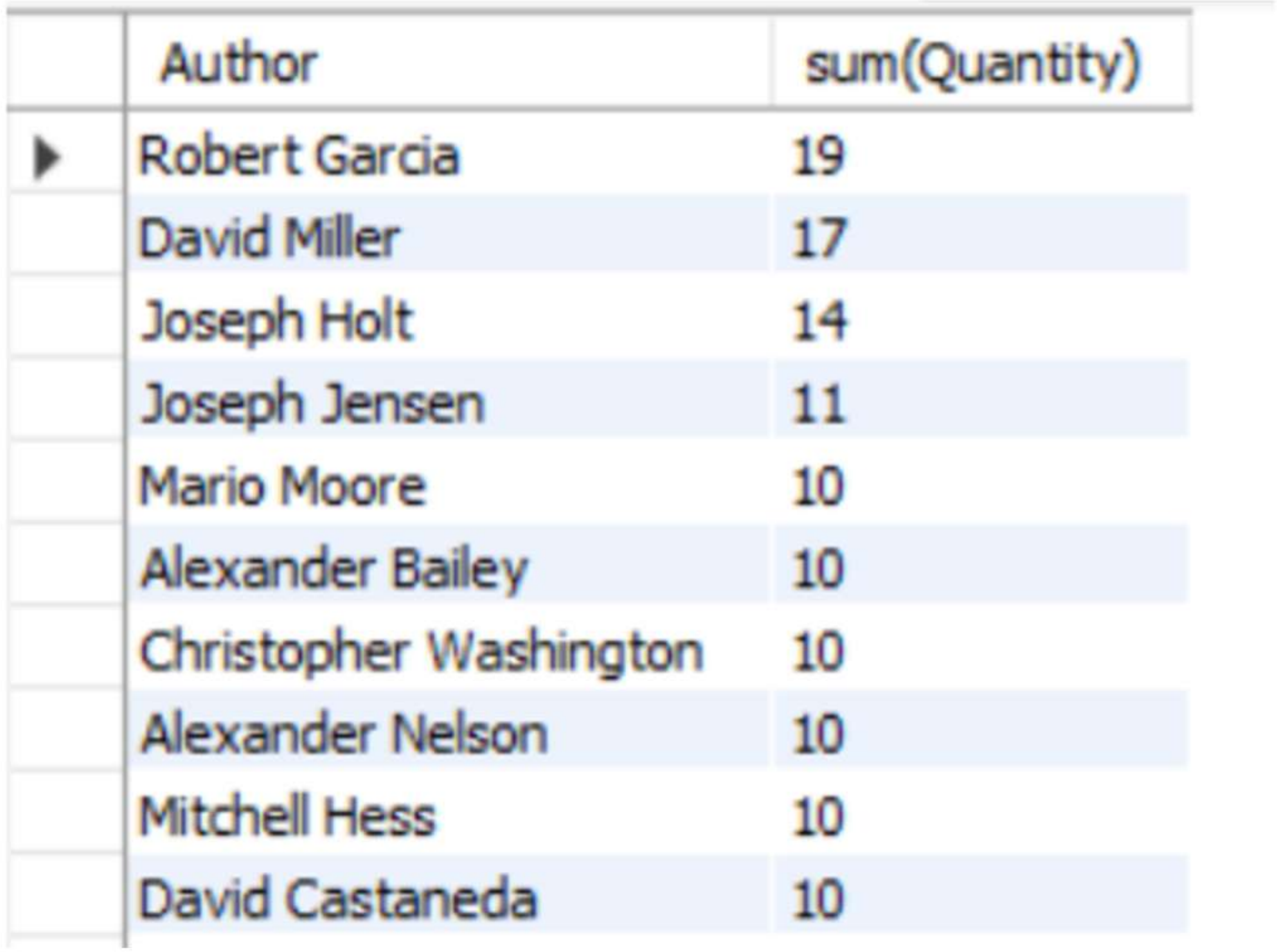
```sql
-- 5) Show the top 3 most expensive books of 'Fantasy' Genre :
SELECT
    *
FROM
    books
WHERE
    Genre = 'Fantasy'
ORDER BY Price DESC
LIMIT 3;
```



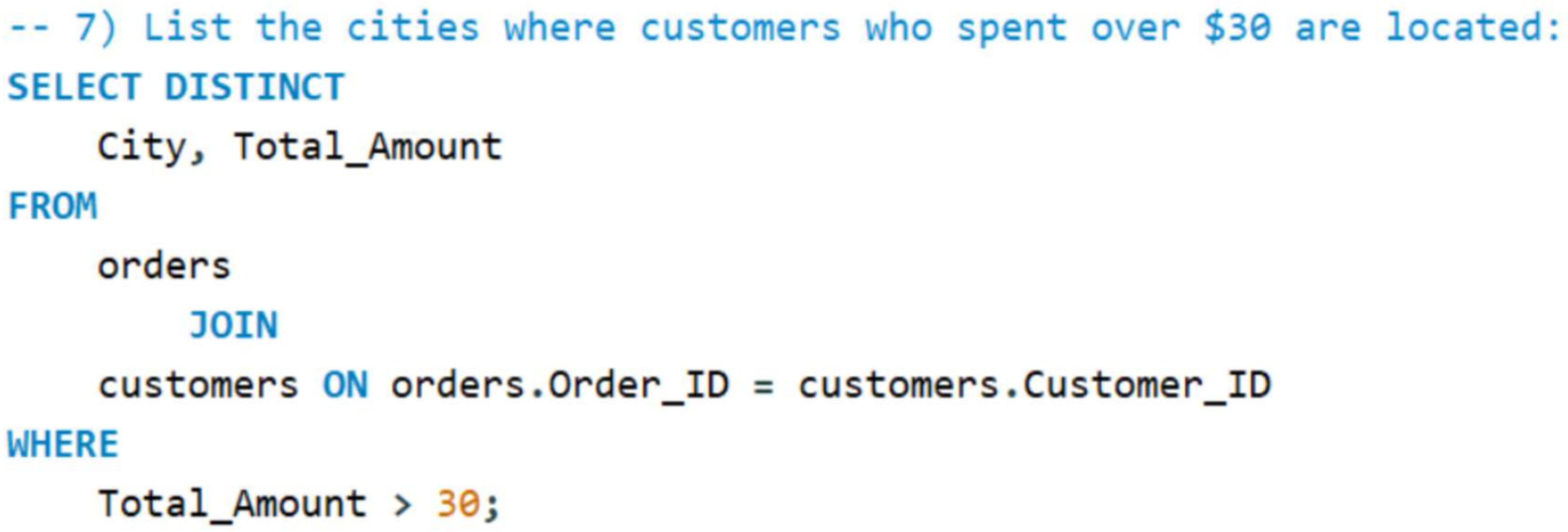
Book_ID	Title	Author	Genre	Published_Year	Price	Stock
240	Stand-alone content-based hub	Lisa Ellis	Fantasy	1957	49.90	41
462	Innovative 3rdgeneration database	Allison Contreras	Fantasy	1988	49.23	62
238	Optimized even-keeled analyzer	Sherri Griffith	Fantasy	1975	48.97	72



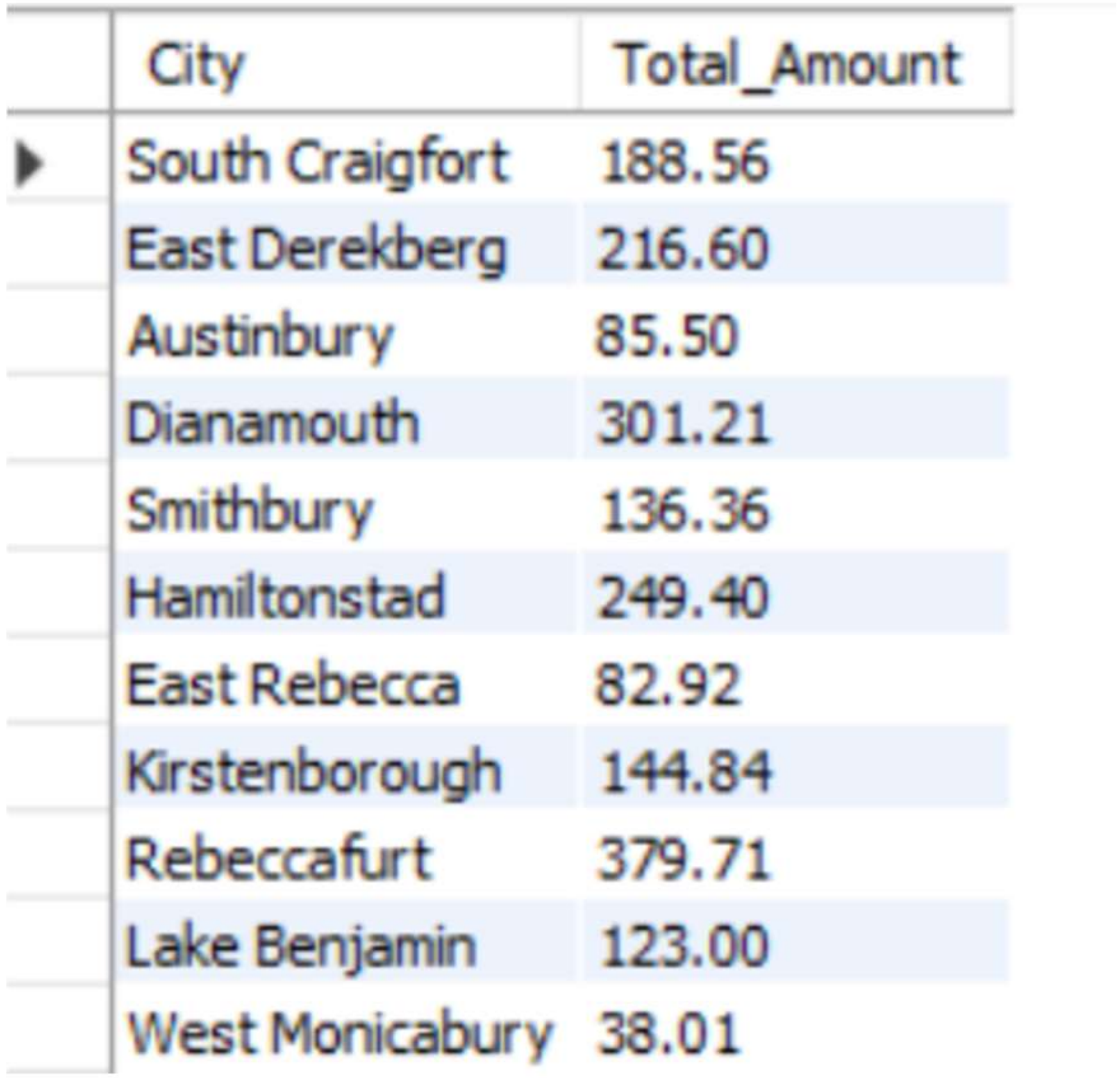
```
-- 6) Retrieve the total quantity of books sold by each author:
SELECT
    Author, SUM(Quantity)
FROM
    books
        JOIN
    orders ON books.Book_ID = orders.Order_ID
GROUP BY Author
ORDER BY SUM(Quantity) DESC;
```



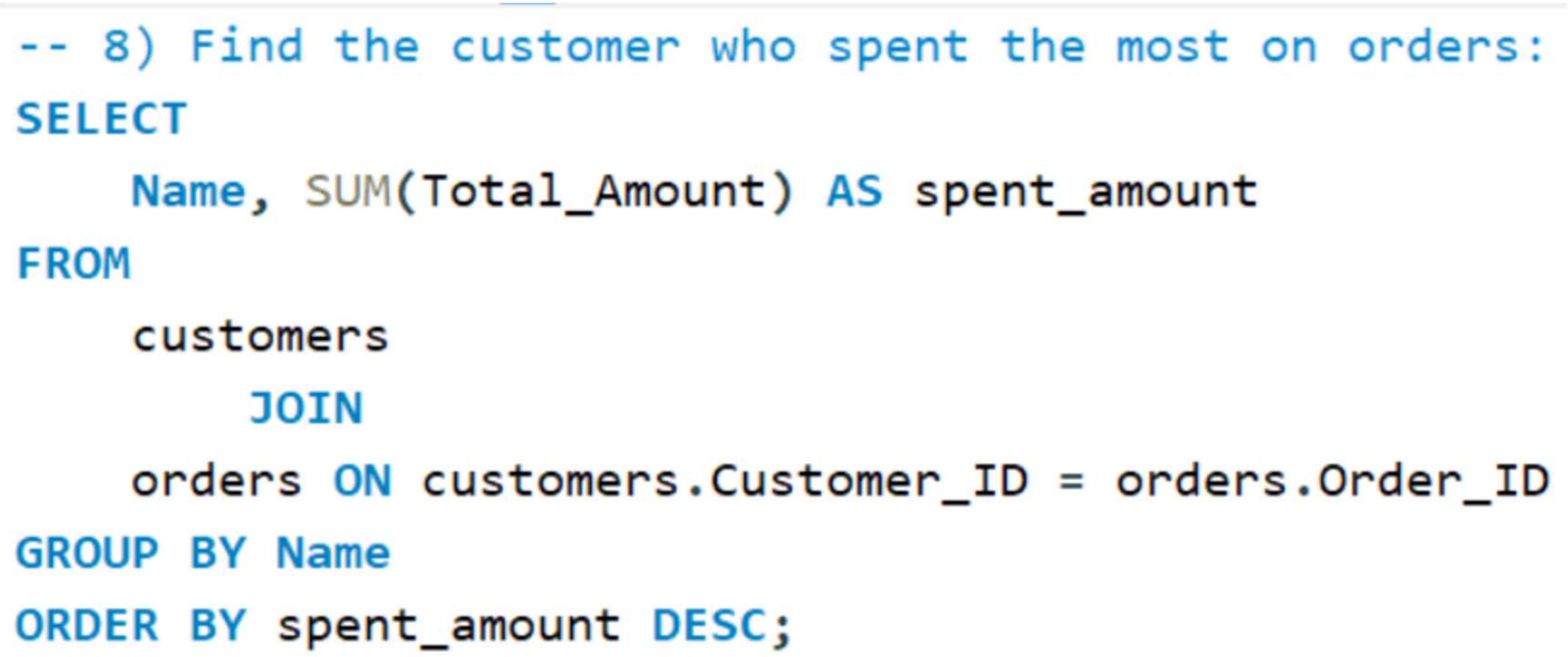
Author	sum(Quantity)
Robert Garcia	19
David Miller	17
Joseph Holt	14
Joseph Jensen	11
Mario Moore	10
Alexander Bailey	10
Christopher Washington	10
Alexander Nelson	10
Mitchell Hess	10
David Castaneda	10



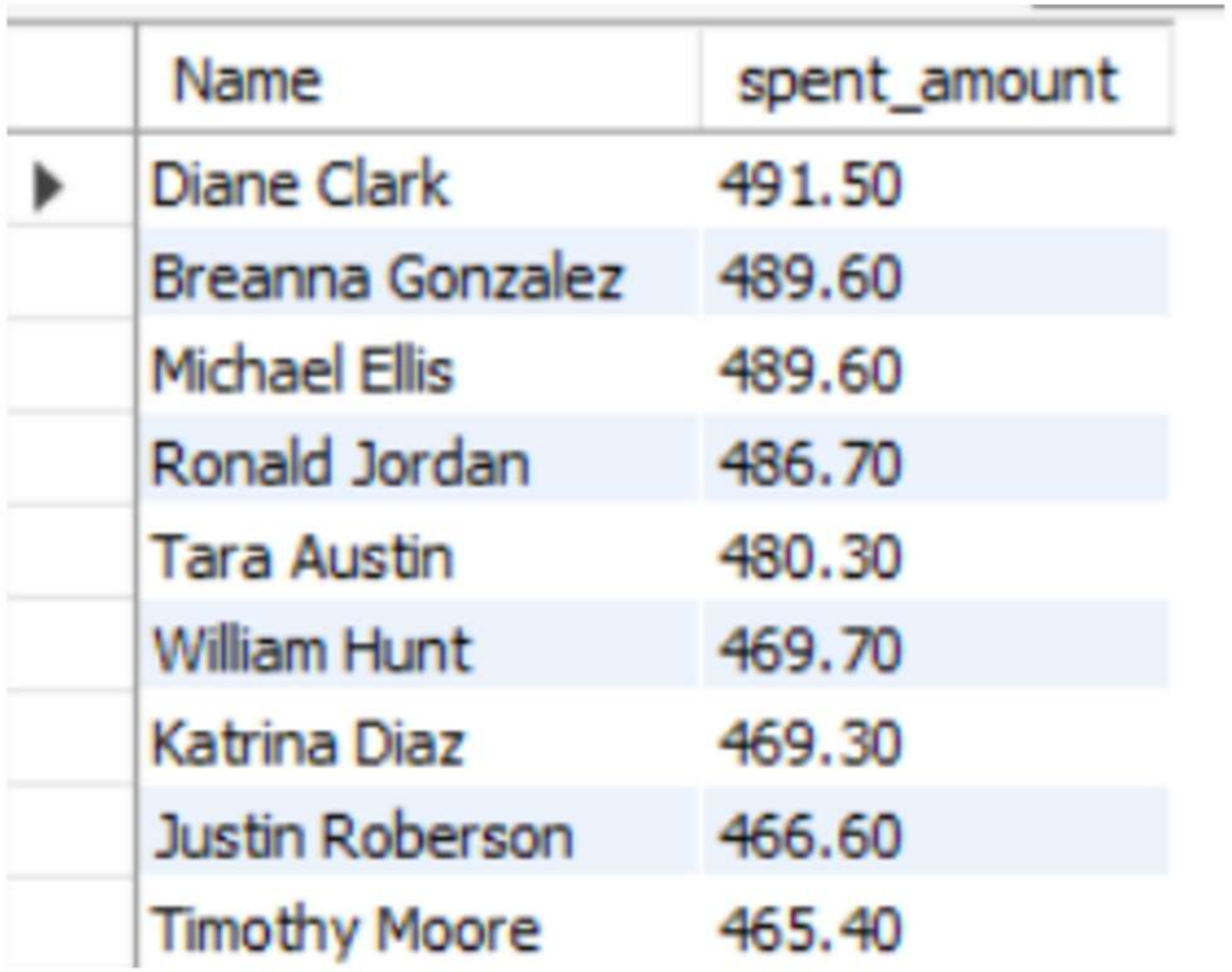
```sql
-- 7) List the cities where customers who spent over $30 are located:
SELECT DISTINCT
    City, Total_Amount
FROM
    orders
        JOIN
    customers ON orders.Order_ID = customers.Customer_ID
WHERE
    Total_Amount > 30;
```



City	Total_Amount
South Craigfort	188.56
East Derekberg	216.60
Austinbury	85.50
Dianamouth	301.21
Smithbury	136.36
Hamiltonstad	249.40
East Rebecca	82.92
Kirstenborough	144.84
Rebeccafurt	379.71
Lake Benjamin	123.00
West Monicabury	38.01

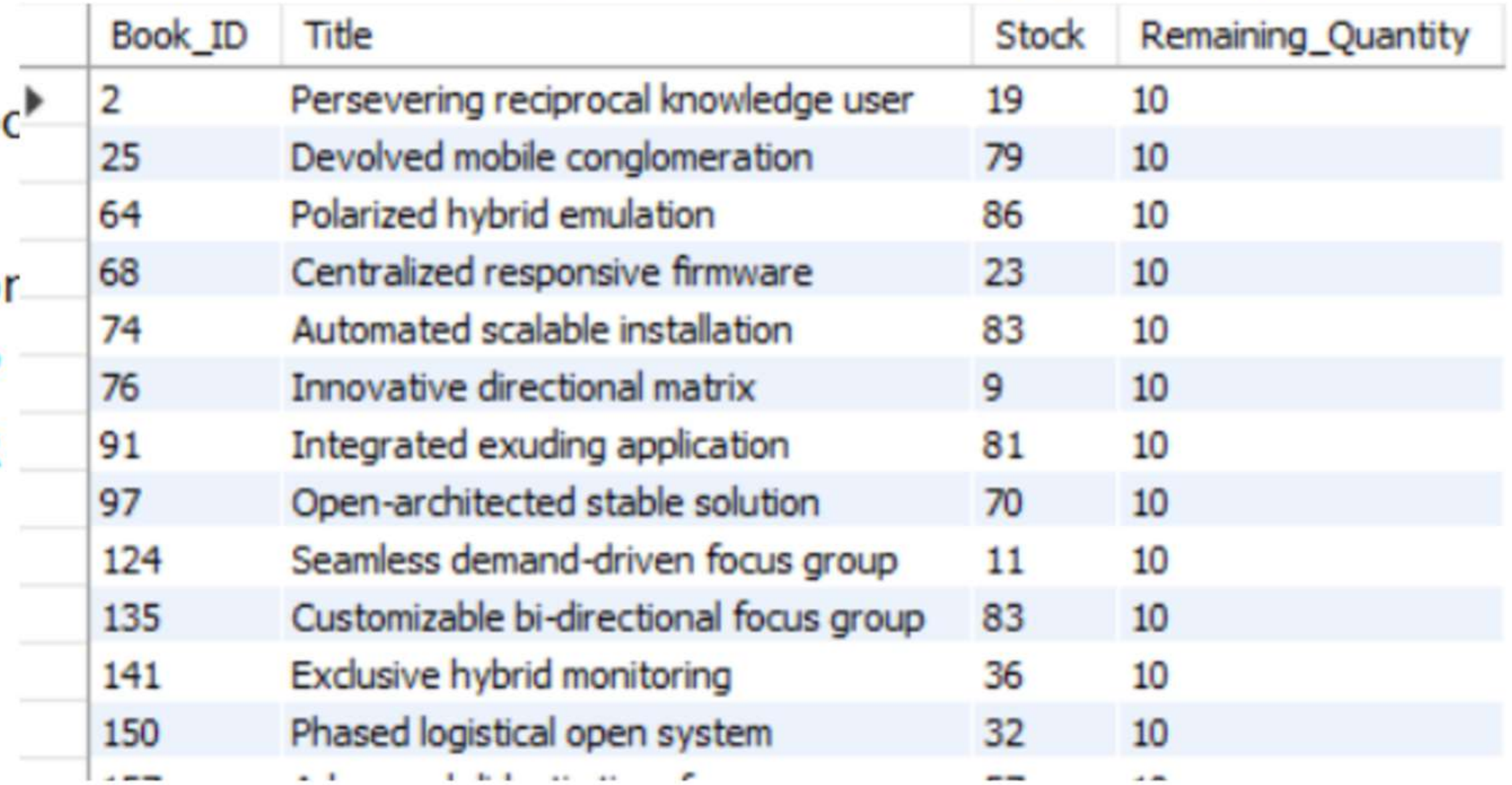


```
-- 8) Find the customer who spent the most on orders:
SELECT
    Name, SUM(Total_Amount) AS spent_amount
FROM
    customers
        JOIN
    orders ON customers.Customer_ID = orders.Order_ID
GROUP BY Name
ORDER BY spent_amount DESC;
```



Name	spent_amount
Diane Clark	491.50
Breanna Gonzalez	489.60
Michael Ellis	489.60
Ronald Jordan	486.70
Tara Austin	480.30
William Hunt	469.70
Katrina Diaz	469.30
Justin Roberson	466.60
Timothy Moore	465.40

```
-- 9) Calculate the stock remaining after fulfilling all orders:
SELECT
    b.Book_ID,
    b.Title,
    b.Stock,
    COALESCE(SUM(Quantity)) AS Remaining_Quantity
FROM
    bc

    or
GROUP
ORDER
```



Book_ID	Title	Stock	Remaining_Quantity
2	Persevering reciprocal knowledge user	19	10
25	Devolved mobile conglomeration	79	10
64	Polarized hybrid emulation	86	10
68	Centralized responsive firmware	23	10
74	Automated scalable installation	83	10
76	Innovative directional matrix	9	10
91	Integrated exuding application	81	10
97	Open-architected stable solution	70	10
124	Seamless demand-driven focus group	11	10
135	Customizable bi-directional focus group	83	10
141	Exclusive hybrid monitoring	36	10
150	Phased logistical open system	32	10

CONCLUSION

The Online Bookstore Data Analysis using SQL helps optimize inventory management, customer segmentation, and revenue analysis. By extracting insights from book sales, customer behavior, and order trends, businesses can make data-driven decisions to enhance profitability. This project showcases SQL skills in data extraction, filtering, and aggregation, making it a valuable addition to any data analyst’s portfolio.

THANK YOU

www.linkedin.com/in/putuka-ramanjaneyulu-2128r