

# ONLINE BOOK STORE DATA ANALYSIS USING **MYSQL**

By - Oyekan Oluwole

oyekanwale





# CONTENT

1. INTRODUCTION
2. PROJECT GOALS
3. DATA DESCRIPTION
4. DATA SCHEMA
5. KPI'S
6. DATA ANALYSIS
7. FINDING INSIGHTS
9. RECOMANDATIONS
10. CONCLUSION



# INTRODUCTION

This project harnessed MySQL's analytical power and SQL techniques (like joins), along with critical thinking, to uncover valuable insights for tackling business challenges within an online bookstore's stock and sales data. The main focus of the analysis was to understand how sales performed based on category, the revenue generated, what customers preferred, and when orders were placed.

## KEY FEATURES:

1. TIME BASED ORDER ANALYSIS.
2. ORDERS MANAGEMENT.
3. REVENUE ANALYSIS.
4. MOST POPULAR BOOKS BASED ON PRICE, ORDER.



# PROJECT GOALS

- 1. **Understand Customer Preferences:** Pinpoint popular book genres, preferred formats (e.g., hardcover, paperback, ebook), and peak browsing/purchase times to better align our offerings with customer interests.
- 2. **Optimize Sales Strategies:** Provide data-backed suggestions to improve sales during slower periods and increase the sales of popular and high-earning titles and genres.
- 3. **Improve Operational Efficiency:** Gain insights into order patterns to optimize our warehouse operations, inventory levels, and potentially even content acquisition planning.
- 4. **Drive Revenue Growth:** Identify opportunities to increase overall revenue through targeted recommendations, adjustments to our book selection and pricing, and improvements to the overall customer experience.



# DATA DESCRIPTION

This document outlines the schema for the tables used in a Book Sales ordering system database. The database includes 3 csv format datasheets- **Customers, Orders, Books.**

## 1. CUSTOMERS :

- This table stores information about each customer data.
- Columns:
  - **Customer\_id (INT, PRIMARY KEY):** A unique identifier assigned to each individual customer within the database. This serves as the primary key for this table.
  - **Name (VARCHAR):** each customer name has been recorded here.
  - **Email and Phone :** Each customer email id and Phone no has been noted here which is unique.
  - **City (VARCHAR):** this explain customers are from which city.
  - **Country (VARCHAR):** customers belong to which country, that has been noted.



## 2. ORDERS:

- . This table stores header-level information for each customer order placed. Each row represents a single, complete order.
- . Columns:
  - **Order\_id (INT, PRIMARY KEY):** A unique identifier assigned to each customer order. This serves as the primary key for this table.
  - **Order\_Date (DATE):** The date on which the order was placed.
  - **Customer\_id (INT, FOREIGN KEY):** this serves as a foreign key to the table referencing customers table.
  - **Book\_id (INT, FOREIGN KEY):** this serves as a foreign key to the table referencing books table.
  - **Quantity (INT):** the quantity of orders from each customers.
  - **Total amount (INT):** the total amount by sales is written here.

## 3. BOOKS:

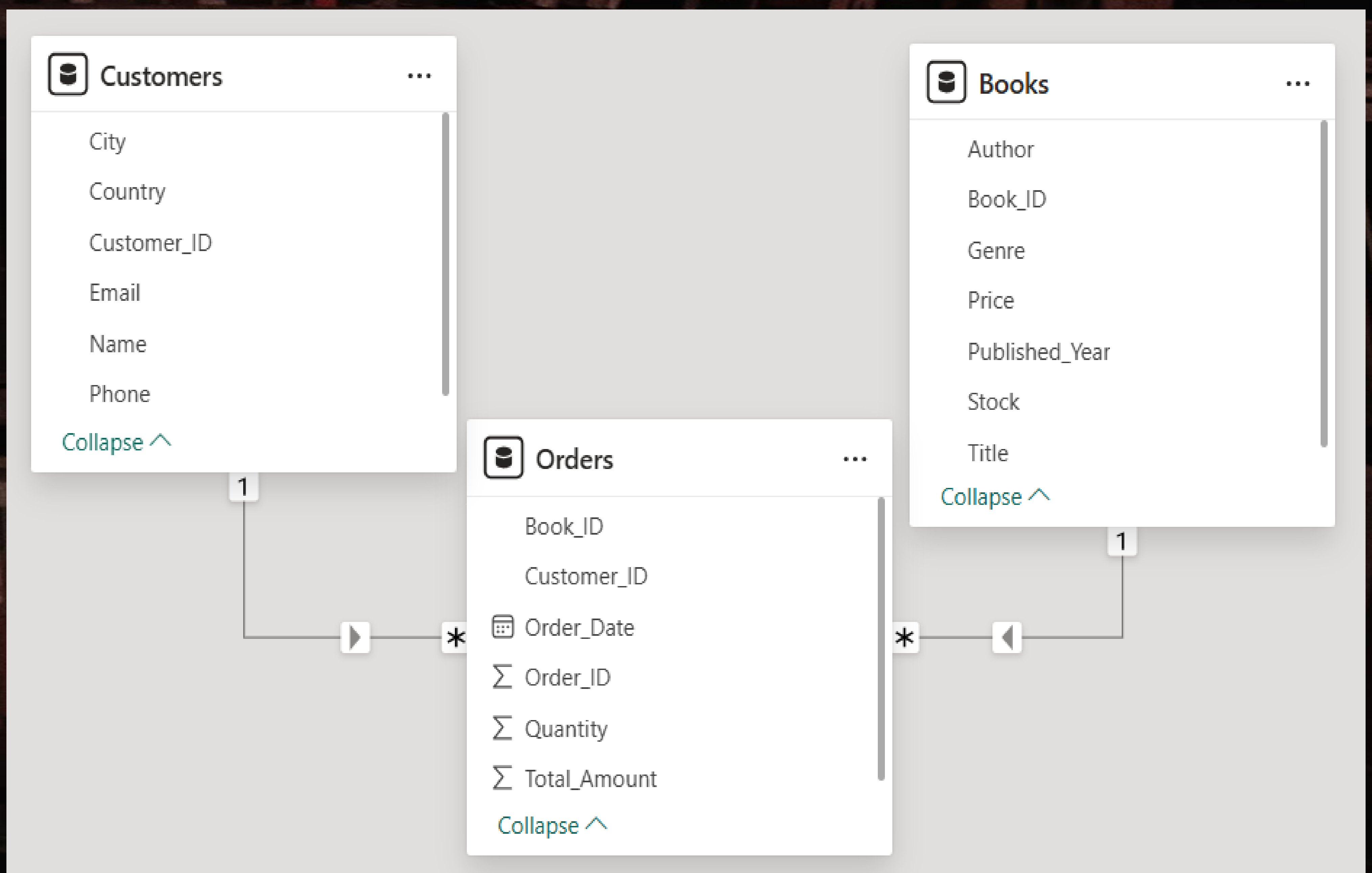
- . This table stores header-level information for each books stock
- . Columns:
  - **Book\_id (INT, PRIMARY KEY):** A unique identifier for each book .
  - **Author name (VARCHAR):** All author names have been mentioned here.
  - **Genre (VARCHAR):** Each author's genre has been mentioned here.
  - **Published year (DATE):** The year on which book was written.
  - **Price (INT):** the price of each book from each authors.
  - **Stock (INT):** the book stocks for individual authors is here.



# DATA SCHEMA

A relational database with 3 tables managing book orders, order types, genres.

**Relationships:** **Customer\_id**, **Book\_id** are primary keys of one table and foreign keys of **ORDERS table**. Foreign keys ensure data links and integrity. Data schema is prepared in data model view of Power BI





# KEY PERFORMANCE INDICATORS

- ❑ **Fiction Sales Percentage:** Track the proportion of total revenue generated by fiction books.
- ❑ **November Sales Growth:** Measure the percentage increase in revenue during November compared to the previous month or year.
- ❑ **Canadian Customer Revenue Contribution:** Monitor the percentage of total revenue originating from Canadian customers.
- ❑ **Loyalty Program Enrollment Rate:** Track the number of customers signing up for the loyalty program.
- ❑ **Repeat Customer Rate:** Measure the percentage of customers who have placed more than one order.
- ❑ **Stockout Rate (Bestsellers):** Monitor how frequently best-selling books are out of stock.
- ❑ **Top Author/Book Sales Contribution:** Track the percentage of total revenue generated by the top-performing authors and individual book titles.
- ❑ **Overall Revenue Growth:** Measure the percentage increase in total revenue over specific periods.
- ❑ **Customer Acquisition Cost (Canada):** Analyze the cost of acquiring new customers in the Canadian market.
- ❑ **Average Order Value (Loyal Customers):** Track the average amount spent per order by loyalty program members or repeat customers.
- ❑ **Inventory Turnover Rate (Bestsellers):** Measure how quickly best-selling books are sold and replaced.



# DATA ANALYSIS

## 1. Retrieve all books in the fiction genre:

```
1  -- 1) Retrieve all books in the "Fiction" genre:
2  •  SELECT * FROM Books
3     WHERE Genre='Fiction';
4
5
```

Result Grid

Filter Rows:

Edit:

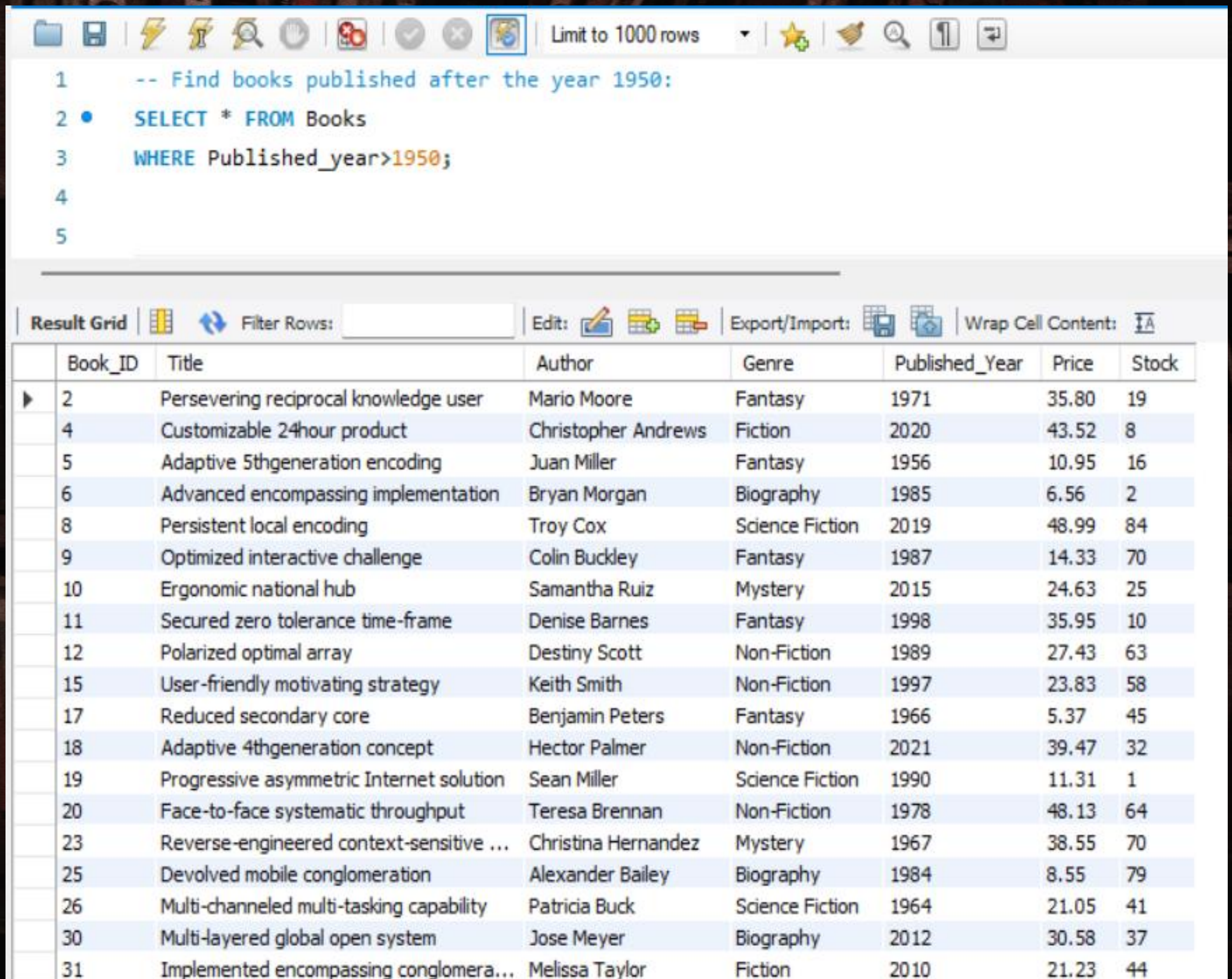
Export/Import:

Wrap Cell Content:

	Book_ID	Title	Author	Genre	Published_Year	Price	Stock
	327	Implemented zero-defect product	Kim Lee	Fiction	1984	21.40	29
	329	Multi-tiered dedicated encryption	Jacob Fischer	Fiction	1973	38.49	27
	331	Profit-focused systematic project	Dennis Hooper	Fiction	1936	15.36	89
	342	Digitized regional monitoring	Kenneth Finley	Fiction	1969	17.11	84
	362	Profit-focused multimedia throughput	Andrea Lucas	Fiction	1933	29.43	64
	366	Re-contextualized national secured line	Robert Wilson	Fiction	1922	13.10	4
	367	Universal analyzing support	Eric Taylor	Fiction	1940	29.32	35
	372	Realigned mobile focus group	Michael Jackson	Fiction	1905	17.73	42
	373	Exclusive dynamic approach	Michelle Mercado	Fiction	1911	39.09	49
	403	Persevering incremental strategy	David Miller	Fiction	1928	15.01	82
	406	De-engineered methodical capacity	James Gonzalez	Fiction	2006	35.79	50
	408	Decentralized tertiary customer loyalty	William Roberts	Fiction	1946	10.07	2



## 2. Retrieve all books published after 1950 :



The screenshot shows a database query interface. At the top, there is a toolbar with various icons and a text input field containing "Limit to 1000 rows". Below the toolbar, a SQL query is displayed in a text editor:

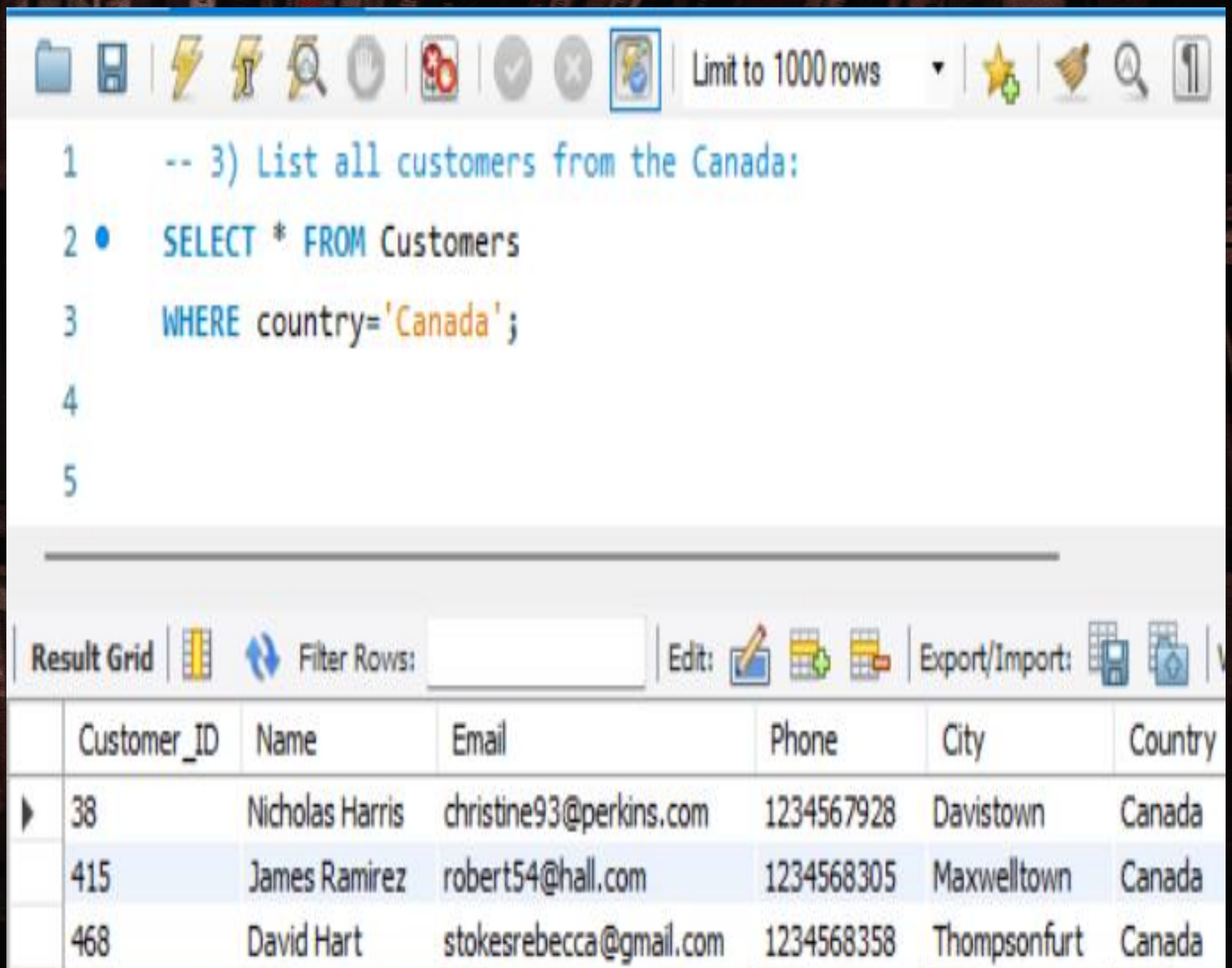
```
1 -- Find books published after the year 1950:
2 • SELECT * FROM Books
3 WHERE Published_year>1950;
4
5
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, and an "Export/Import:" button. The "Result Grid" displays a table of book records. The table has the following columns: Book\_ID, Title, Author, Genre, Published\_Year, Price, and Stock. The data is as follows:

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
25	Devolved mobile conglomeration	Alexander Bailey	Biography	1984	8.55	79
26	Multi-channeled multi-tasking capability	Patricia Buck	Science Fiction	1964	21.05	41
30	Multi-layered global open system	Jose Meyer	Biography	2012	30.58	37
31	Implemented encompassing conglomer...	Melissa Taylor	Fiction	2010	21.23	44



### 3. Retrieve all customers from Canada :



The screenshot shows a SQL query editor interface. The query is as follows:

```
1  -- 3) List all customers from the Canada:
2  •  SELECT * FROM Customers
3     WHERE country='Canada';
4
5
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field and "Edit:" and "Export/Import:" buttons. The result grid displays the following data:

	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. Retrieve all orders placed in November 2023 :

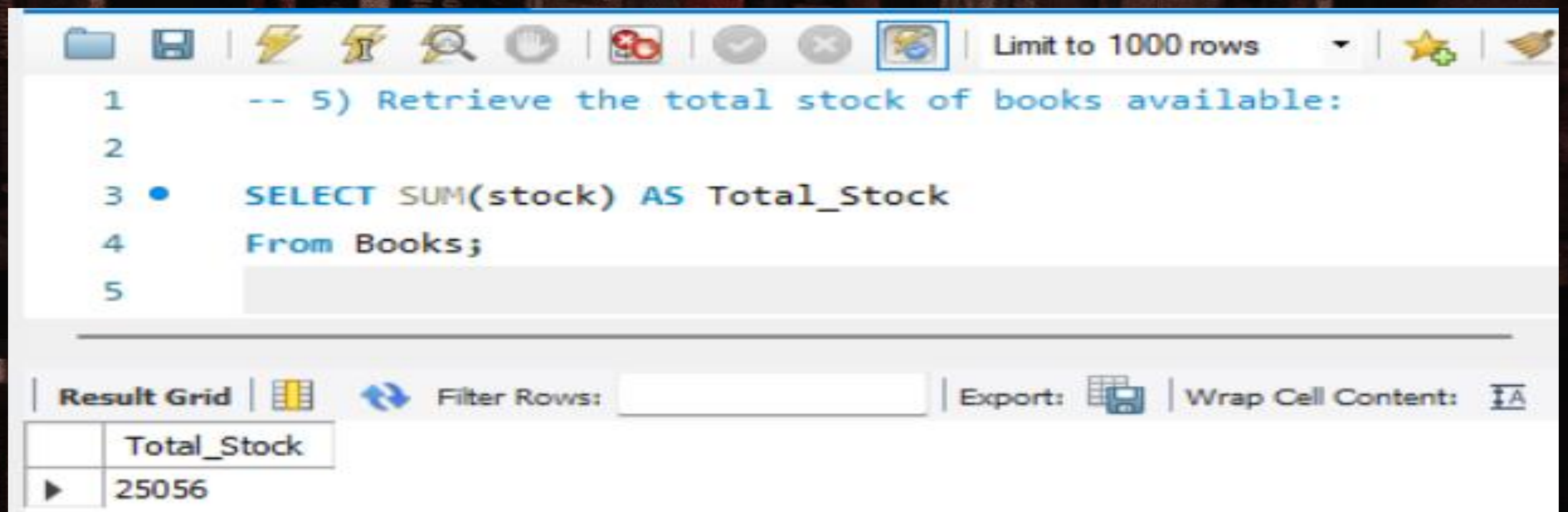
```
-- 4) Show orders placed in November 2023:

SELECT * FROM Orders
WHERE order_date BETWEEN '2023-11-01' AND '2023-11-30';
```

Result Grid						
Filter Rows:						
Edit:						
Export/Imp						
	Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
▶	4	433	343	2023-11-25	7	301.21
	19	496	60	2023-11-17	9	316.26
	75	291	375	2023-11-30	5	170.75
	132	469	333	2023-11-22	7	194.32
	137	474	471	2023-11-25	8	363.04
	163	207	384	2023-11-23	3	101.76
	182	129	293	2023-11-01	7	125.51
	200	313	303	2023-11-23	1	6.57
	213	325	447	2023-11-17	7	253.75
	231	22	384	2023-11-11	1	33.92
	245	386	97	2023-11-01	9	411.66
	252	405	387	2023-11-15	5	237.10
	257	123	403	2023-11-06	1	15.01
	288	6	128	2023-11-13	1	24.04
	307	368	133	2023-11-17	1	20.96
	322	270	112	2023-11-08	2	16.04
	344	385	218	2023-11-25	5	26.80
	389	485	391	2023-11-18	2	66.84
	414	23	234	2023-11-10	1	7.15
	429	449	146	2023-11-01	7	101.50
	432	420	168	2023-11-04	3	42.39
	449	490	222	2023-11-18	1	29.59



## 5. Retrieve total stock of books available :



The screenshot shows a SQL query editor with a toolbar at the top. The query text is as follows:

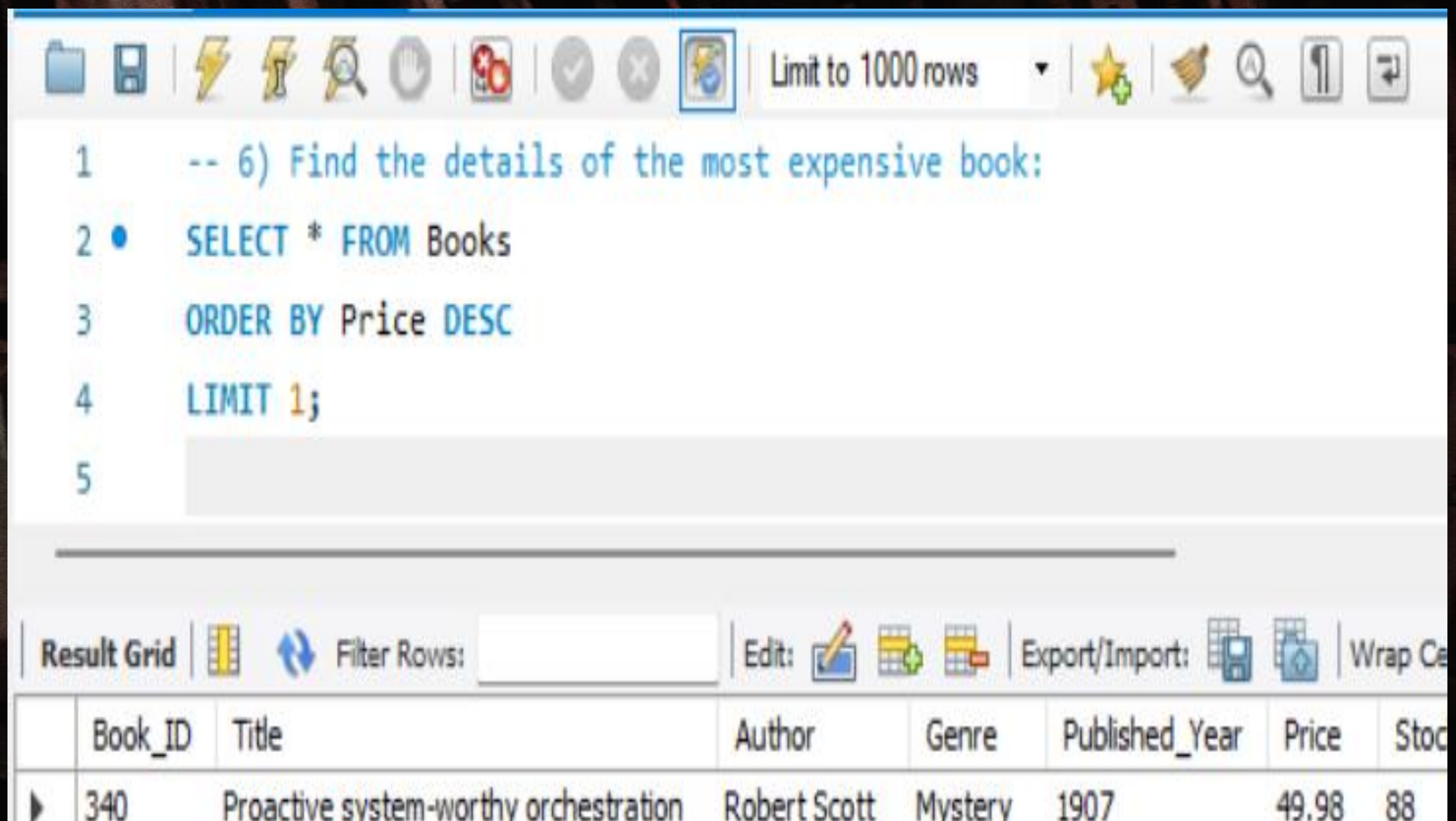
```
1  -- 5) Retrieve the total stock of books available:
2
3  •  SELECT SUM(stock) AS Total_Stock
4     From Books;
5
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid displays a single row with the following data:

Total_Stock
25056

## 6. Retrieve the details of most expensive book

:



The screenshot shows a SQL query editor with a toolbar at the top. The query text is as follows:

```
1  -- 6) Find the details of the most expensive book:
2  •  SELECT * FROM Books
3     ORDER BY Price DESC
4     LIMIT 1;
5
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, an "Export/Import:" button, and a "Wrap Cell Content:" checkbox. The result grid displays a single row with the following data:

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



## 7. Retrieve all customers who ordered more than 1 quantity of a book :

```
1  -- 7) Show all customers who ordered more than 1 quantity of a book:
2  •  SELECT * FROM Orders
3     WHERE quantity>1;
4
5
```

	Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
▶	1	84	169	2023-05-26	8	188.56
	2	137	301	2023-01-23	10	216.60
	3	216	261	2024-05-27	6	85.50
	4	433	343	2023-11-25	7	301.21
	5	14	431	2023-07-26	7	136.36
	6	439	119	2024-10-11	5	249.40
	7	195	467	2023-10-23	6	82.92
	8	32	159	2024-05-07	4	144.84
	9	109	407	2024-01-04	9	379.71
	10	94	122	2024-07-09	4	123.00
	12	454	3	2024-06-17	2	31.50
	13	420	180	2023-06-08	5	125.45
	14	454	319	2023-08-24	2	85.22
	15	127	479	2023-01-10	6	229.62
	16	412	196	2023-10-06	8	53.52
	17	462	481	2023-03-20	5	52.75



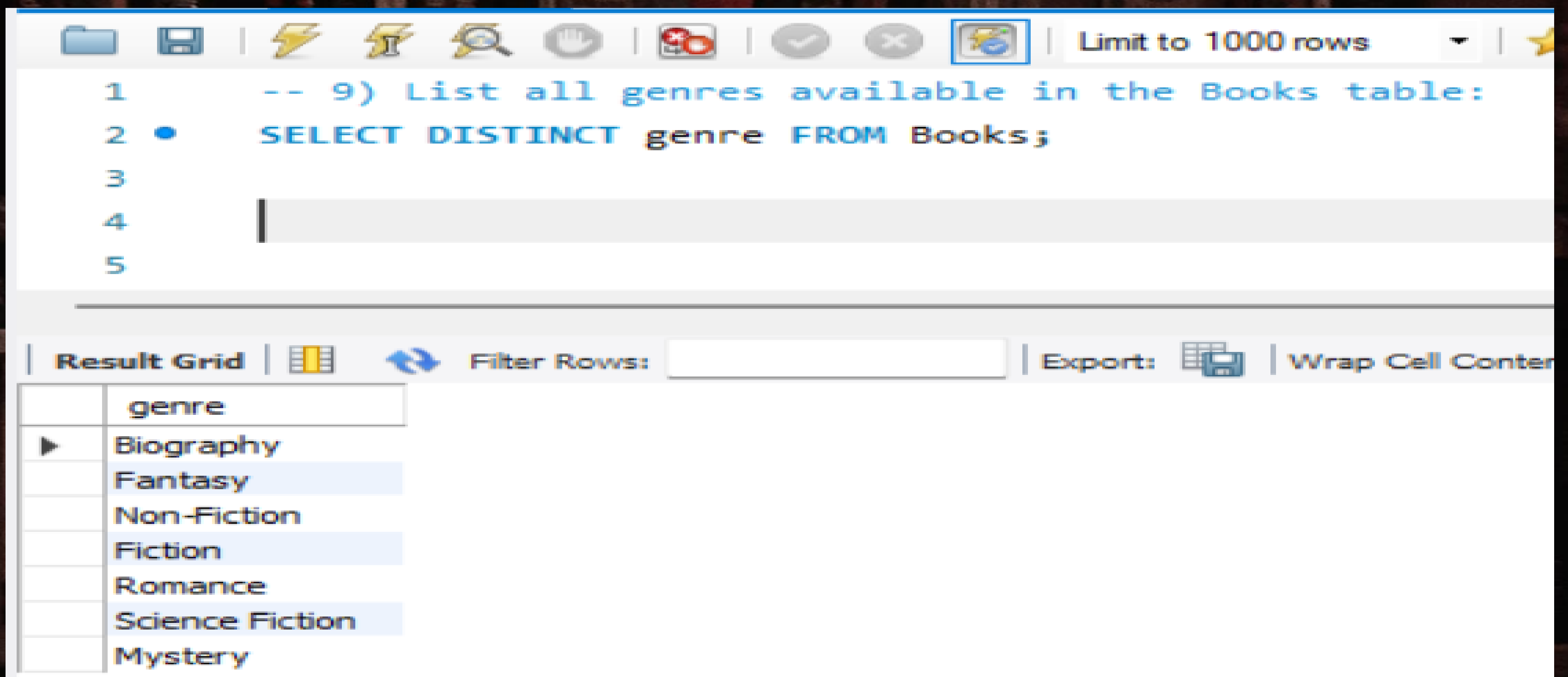
## 8. Retrieve all orders where the total amount exceeds \$20 :

```
1  -- 8) Retrieve all orders where the total amount exceeds $20:
2  •  SELECT * FROM Orders
3     WHERE total_amount>20;
4
5
```

Result Grid						
Filter Rows:						
	Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
▶	1	84	169	2023-05-26	8	188.56
	2	137	301	2023-01-23	10	216.60
	3	216	261	2024-05-27	6	85.50
	4	433	343	2023-11-25	7	301.21
	5	14	431	2023-07-26	7	136.36
	6	439	119	2024-10-11	5	249.40
	7	195	467	2023-10-23	6	82.92
	8	32	159	2024-05-07	4	144.84
	9	109	407	2024-01-04	9	379.71
	10	94	122	2024-07-09	4	123.00
	11	131	206	2023-10-16	1	38.01
	12	454	3	2024-06-17	2	31.50
	13	420	180	2023-06-08	5	125.45
	14	454	319	2023-08-24	2	85.22
	15	127	479	2023-01-10	6	229.62
	16	412	196	2023-10-06	8	53.52
	17	462	481	2023-03-20	5	52.75
	18	377	101	2024-08-07	4	193.96
	19	496	60	2023-11-17	9	316.26
	21	356	287	2024-12-05	3	120.57
	22	177	427	2024-06-10	3	54.00
	23	119	301	2023-08-04	3	64.98



## 9. List all genre available in books table :



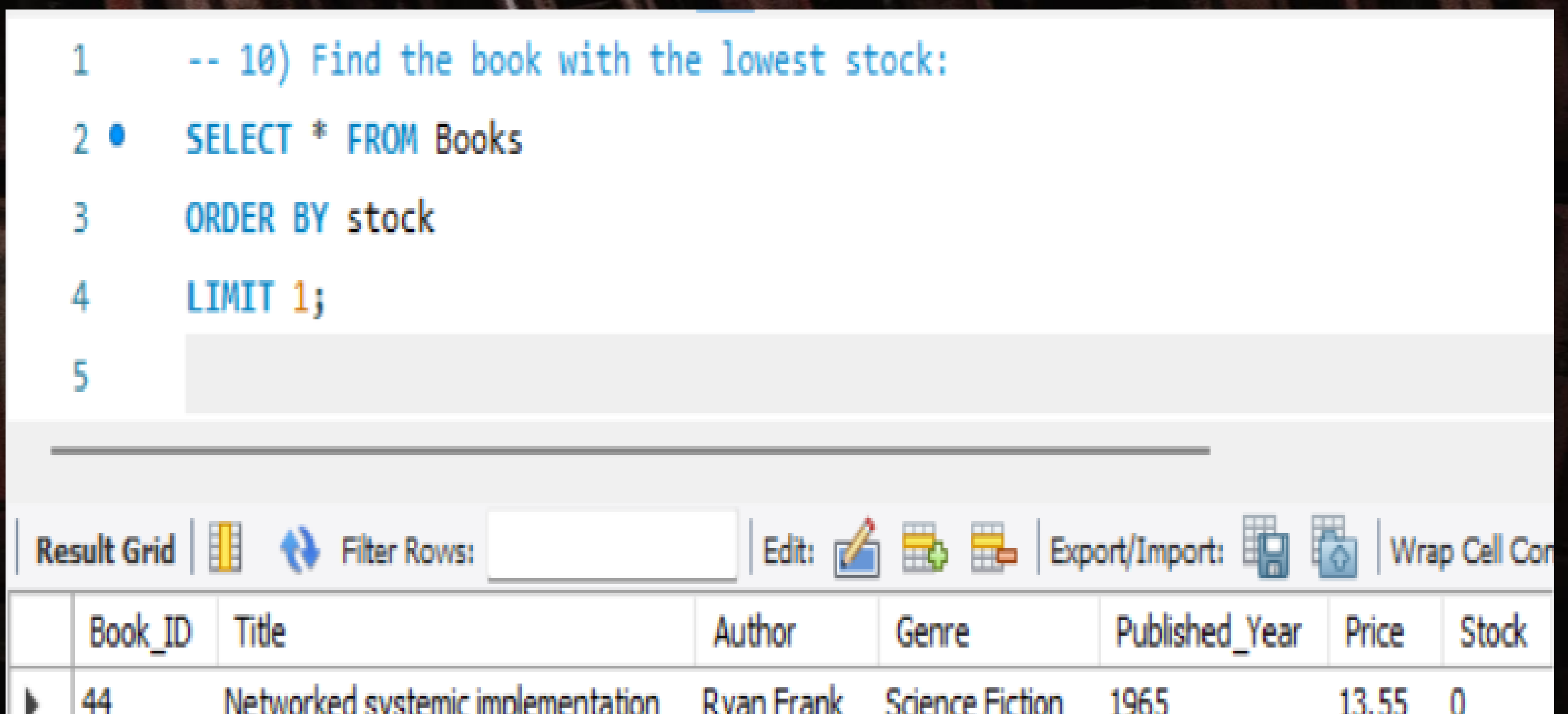
The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1  -- 9) List all genres available in the Books table:
2  • SELECT DISTINCT genre FROM Books;
3
4
5
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows" input field and an "Export" button. The result grid displays a list of genres:

genre
Biography
Fantasy
Non-Fiction
Fiction
Romance
Science Fiction
Mystery

## 10. Find the book with lowest stock :



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

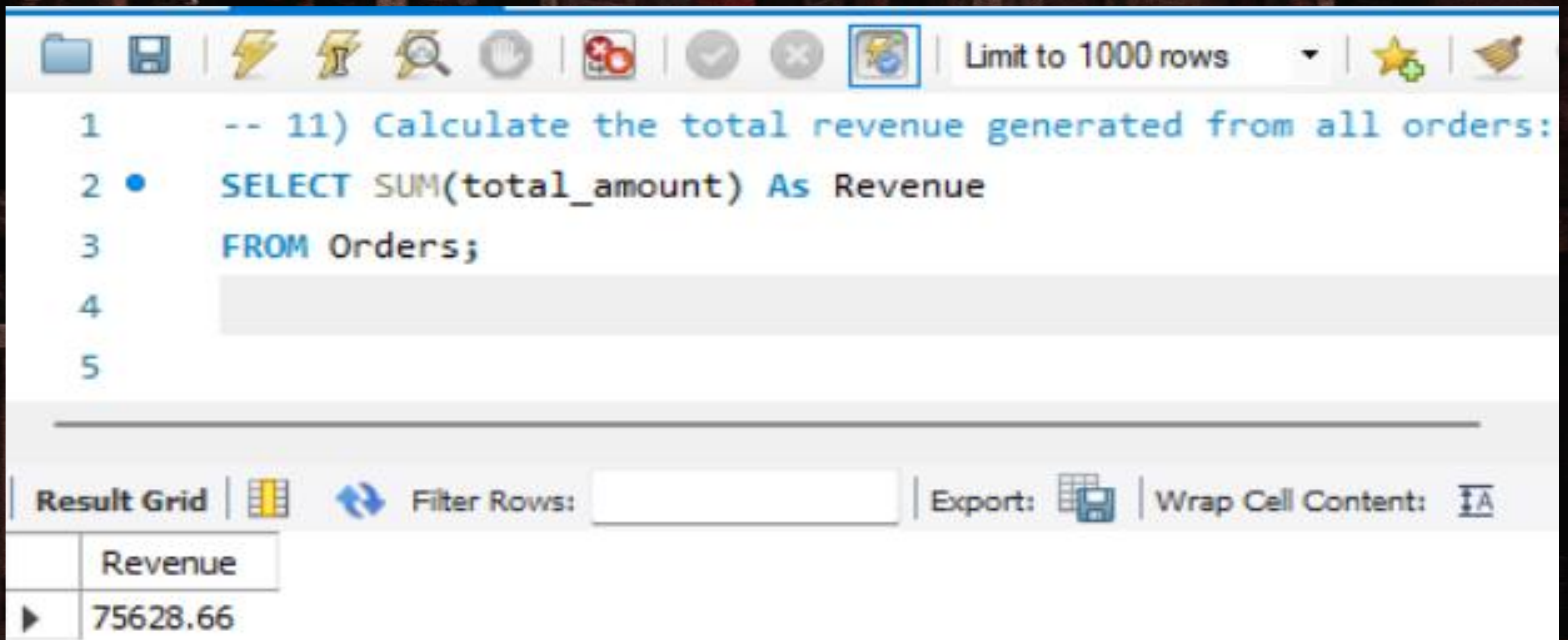
```
1  -- 10) Find the book with the lowest stock:
2  • SELECT * FROM Books
3    ORDER BY stock
4    LIMIT 1;
5
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows" input field, an "Edit" button, and an "Export/Import" button. The result grid displays a table with the following data:

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
44	Networked systemic implementation	Ryan Frank	Science Fiction	1965	13.55	0



## 11. Total revenue generated from all orders :



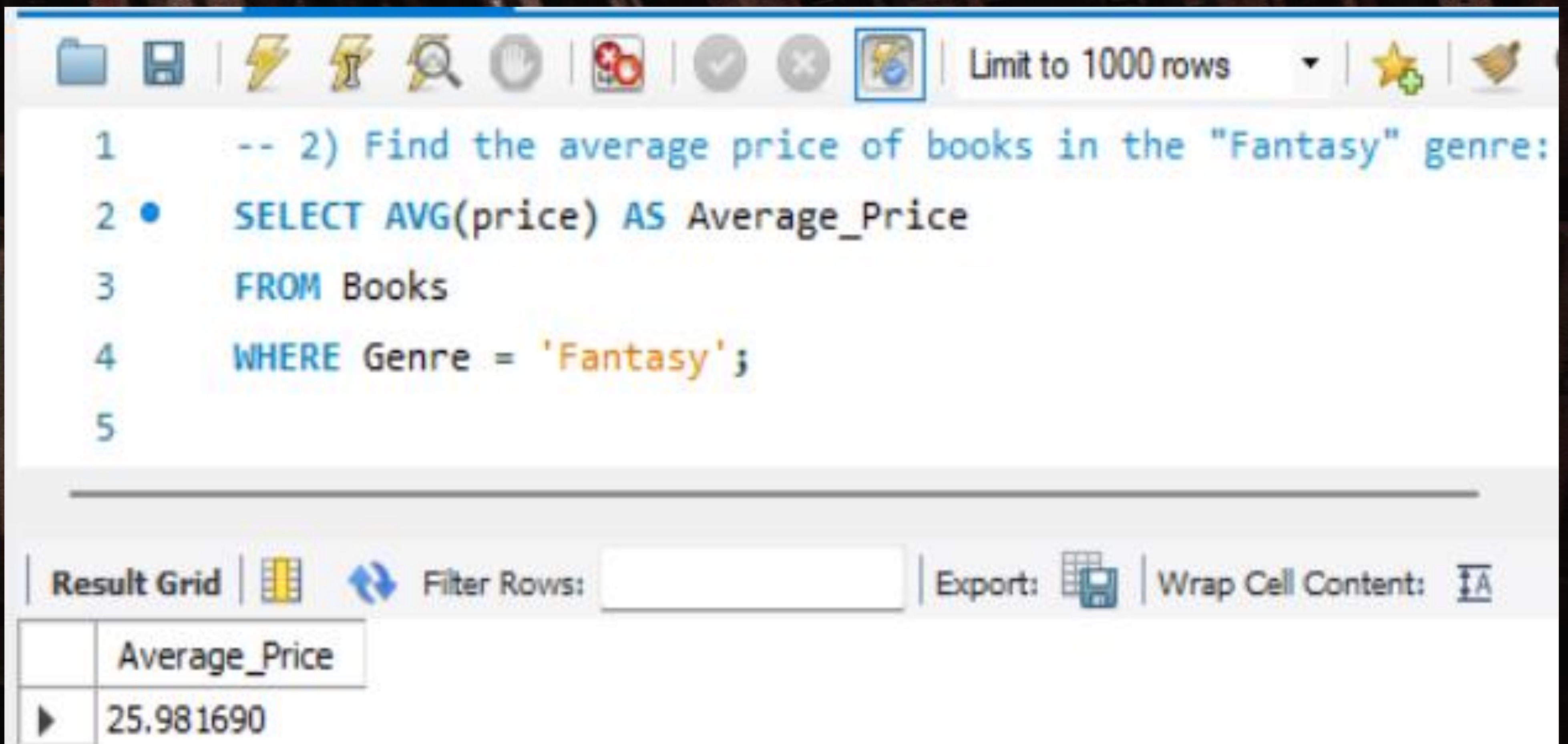
The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, execution, and settings. The query text is as follows:

```
-- 11) Calculate the total revenue generated from all orders:
SELECT SUM(total_amount) As Revenue
FROM Orders;
```

Below the query editor, there is a control bar with a 'Result Grid' button, a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays a single row with the column header 'Revenue' and the value '75628.66'.

Revenue
75628.66

## 12. Avg price of books in Fantasy genre :



The screenshot shows a SQL query editor with a toolbar at the top. The query text is as follows:

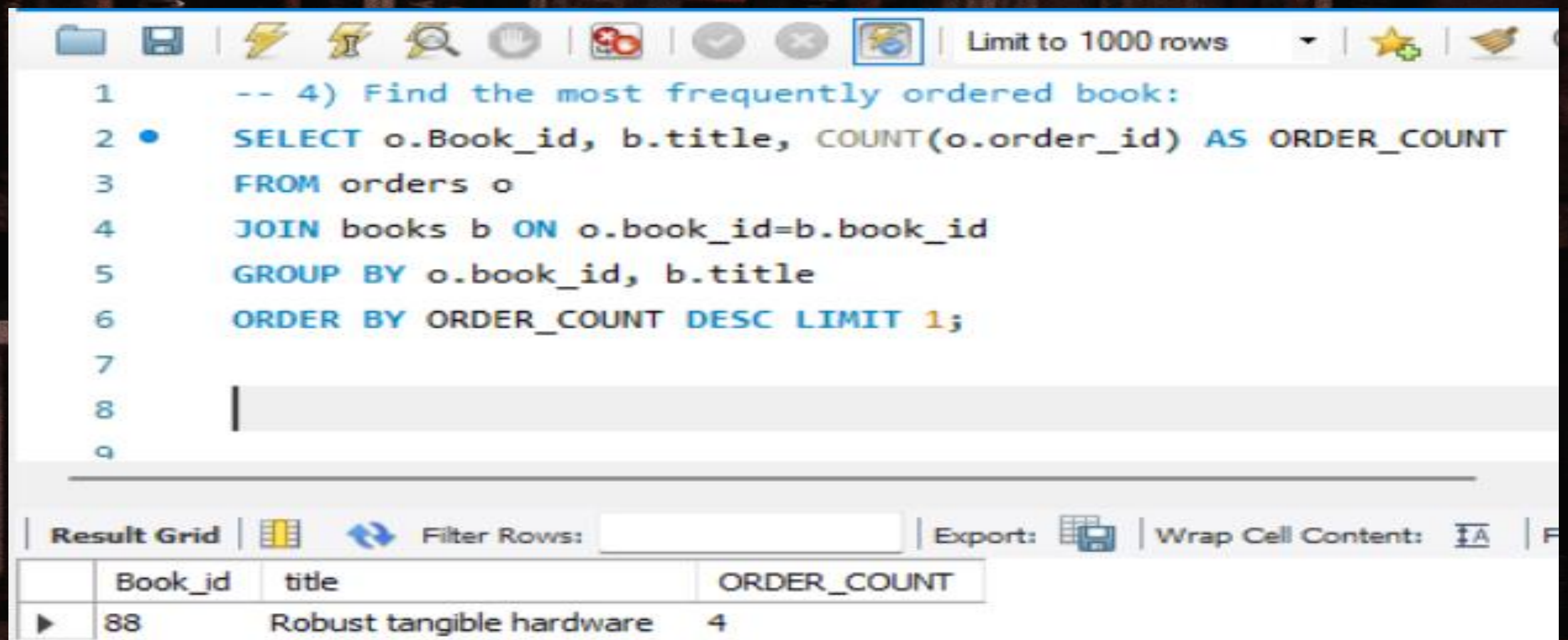
```
-- 2) Find the average price of books in the "Fantasy" genre:
SELECT AVG(price) AS Average_Price
FROM Books
WHERE Genre = 'Fantasy';
```

Below the query editor, there is a control bar with a 'Result Grid' button, a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays a single row with the column header 'Average\_Price' and the value '25.981690'.

Average_Price
25.981690



## 13. Find the most frequently ordered book :



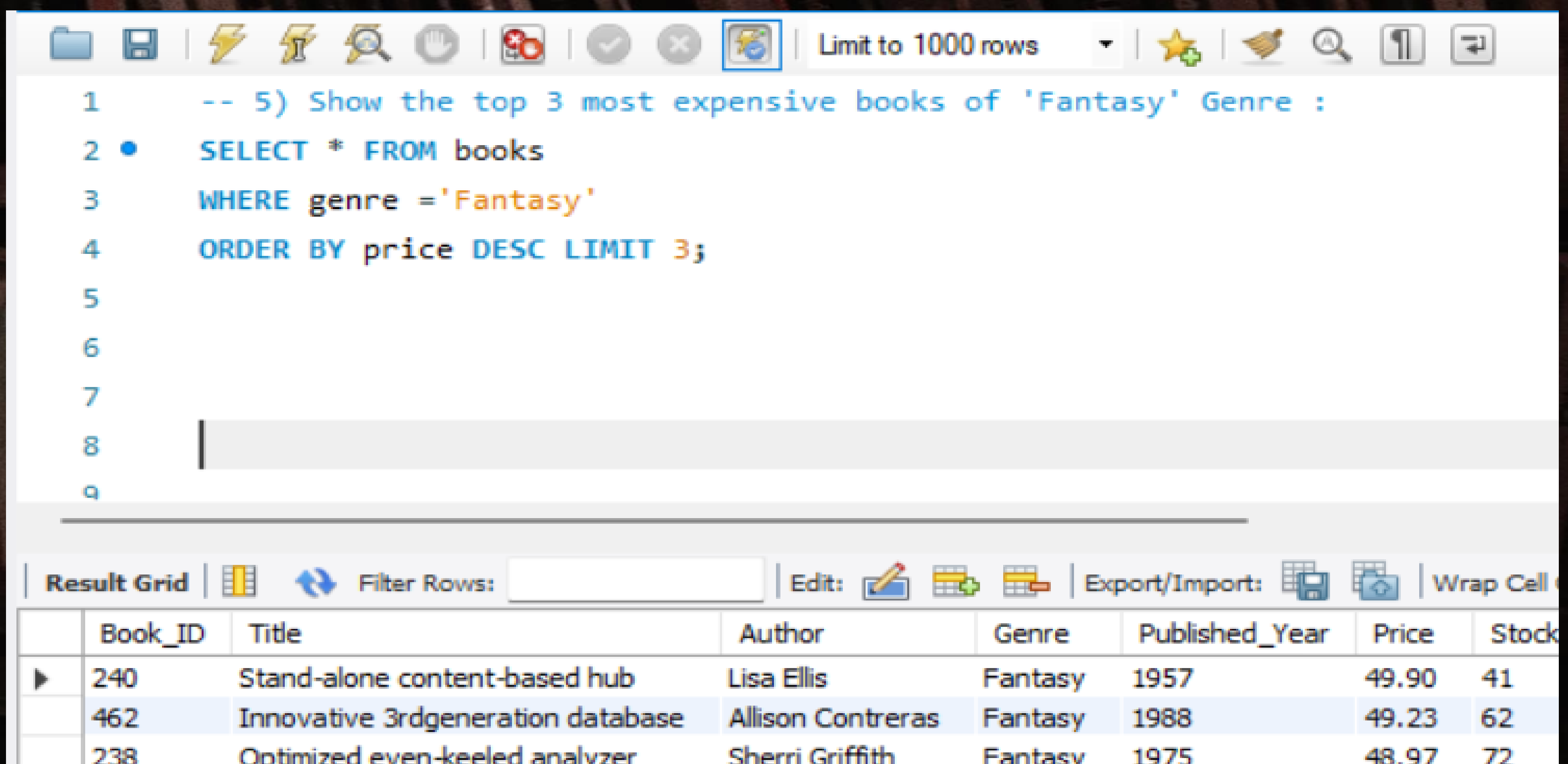
The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
-- 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;
```

Below the query editor, the 'Result Grid' is displayed with the following data:

Book_id	title	ORDER_COUNT
88	Robust tangible hardware	4

## 14. Top 3 expensive books of Fantasy genre :



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

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

Below the query editor, the 'Result Grid' is displayed with the following data:

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



## 15. Calculate the stock remaining after fulfilling all orders :

```
-- 9) Calculate the stock remaining after fulfilling all 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 b
LEFT JOIN orders o ON b.book_id=o.book_id
GROUP BY b.book_id ORDER BY b.book_id;
```

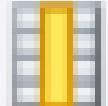


Result Grid | Filter Rows: | Export: | Wrap Cell Content: [A](#)

	book_id	title	stock	Order_quantity	Remaining_Quantity
4	4	Customizable 24hour product	8	0	8
5	5	Adaptive 5thgeneration encoding	16	8	8
6	6	Advanced encompassing implementation	2	0	2
7	7	Open-architected exuding structure	95	5	90
8	8	Persistent local encoding	84	3	81
9	9	Optimized interactive challenge	70	0	70
10	10	Ergonomic national hub	25	1	24
11	11	Secured zero tolerance time-frame	10	5	5
12	12	Polarized optimal array	63	0	63
13	13	Adaptive 5thgeneration orchestration	99	9	90
14	14	Re-engineered demand-driven parallelism	95	0	95
15	15	User-friendly motivating strategy	58	0	58



16. List the cities where customers who spent over \$30 are located :

```
1  -- 7) List the cities where customers who spent over $30 are located:
2
3  •  SELECT DISTINCT c.city, total_amount
4     FROM orders o
5     JOIN customers c ON o.customer_id=c.customer_id
6     WHERE o.total_amount > 30;
7
8
9
```

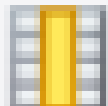


Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	city	total_amount
▶	Lake Paul	188.56
	North Keith	216.60
	Kelseyfort	85.50
	East David	301.21
	Richardsonville	136.36
	Ramosstad	249.40
	Rogersborough	82.92
	New Carlosbury	144.84
	Ravenberg	379.71
	West Anthony	123.00



# 17. Retrieve the total quantity of books sold by each author :

```
1  -- 6) Retrieve the total quantity of books sold by each author:
2
3  •  SELECT b.author, SUM(o.quantity) AS Total_Books_Sold
4     FROM orders o
5     JOIN books b ON o.book_id=b.book_id
6     GROUP BY b.Author;
7
8
9
```

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	author	Total_Books_Sold
▶	Margaret Moore	8
	John Davidson	13
	Christopher Fuentes	6
	Marissa Smith	16
	Christopher Dixon	15
	Tonya Saunders	21
	Larry Hunt	6
	Brandon Foster	4
	Michelle Bell	11
	Mary French	14



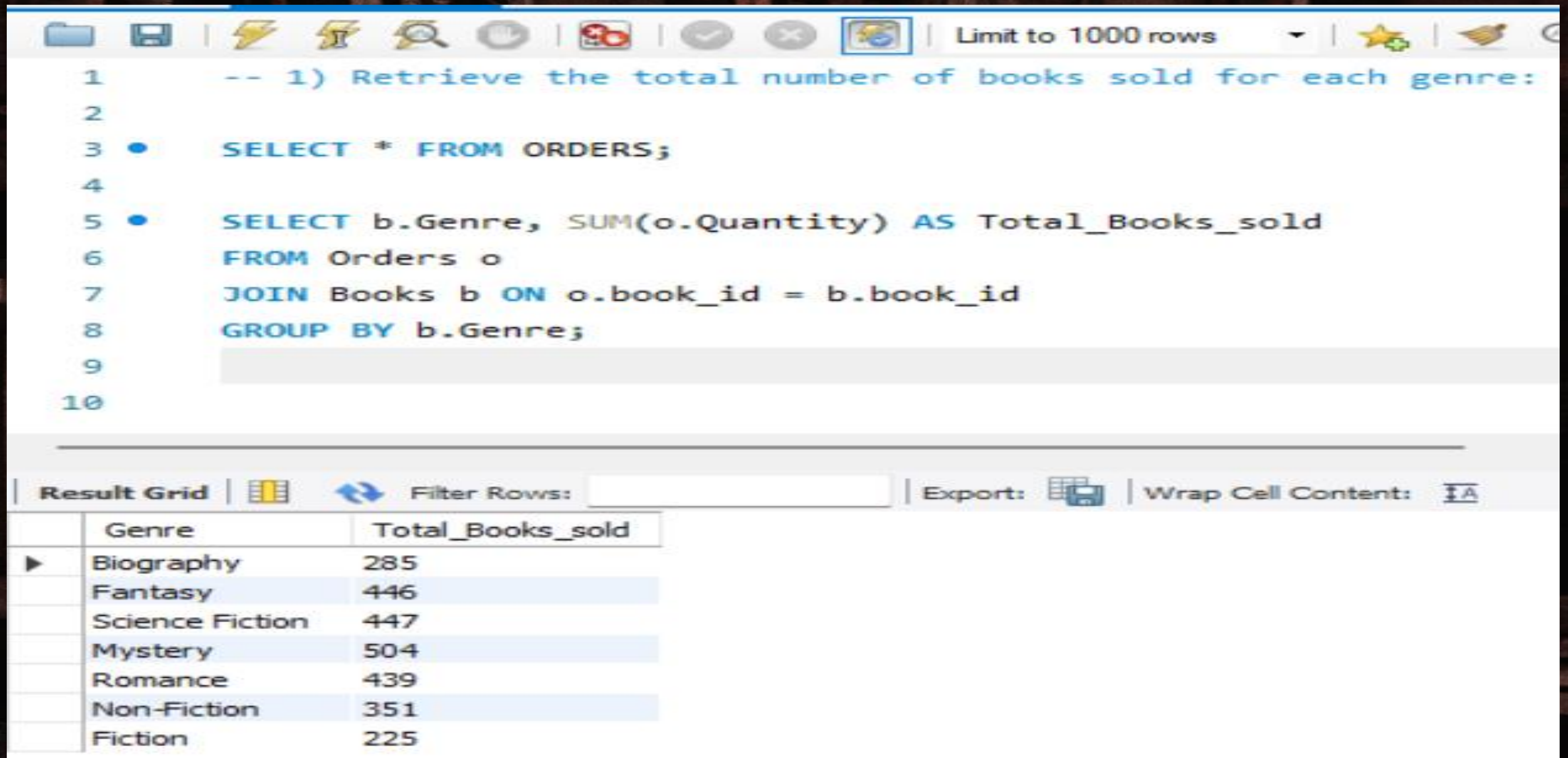
## 18. List of customers who have placed at least 2 orders :

```
1  -- 3) List customers who have placed at least 2 orders:
2  • SELECT o.customer_id, c.name, COUNT(o.Order_id) AS ORDER_COUNT
3  FROM orders o
4  JOIN customers c ON o.customer_id=c.customer_id
5  GROUP BY o.customer_id, c.name
6  HAVING COUNT(Order_id) >=2;
7
8
9
```

Result Grid		 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	customer_id	name	ORDER_COUNT	
▶	84	Gary Blair	2	
	137	Steven Miller	2	
	216	Phillip Allen	2	
	14	John Wood	2	
	195	Dominique Turner	3	
	109	Jacob Kelley	2	
	94	Mr. David Cox	3	
	131	Peter Smith	2	
	454	April Anderson	2	
	420	Andrew Murray	3	
	462	James Brewer	3	
	377	Darrell Khan	2	
	177	Sarah Powell	2	



## 19. Retrieve total no. of books sold for each genre :



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

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

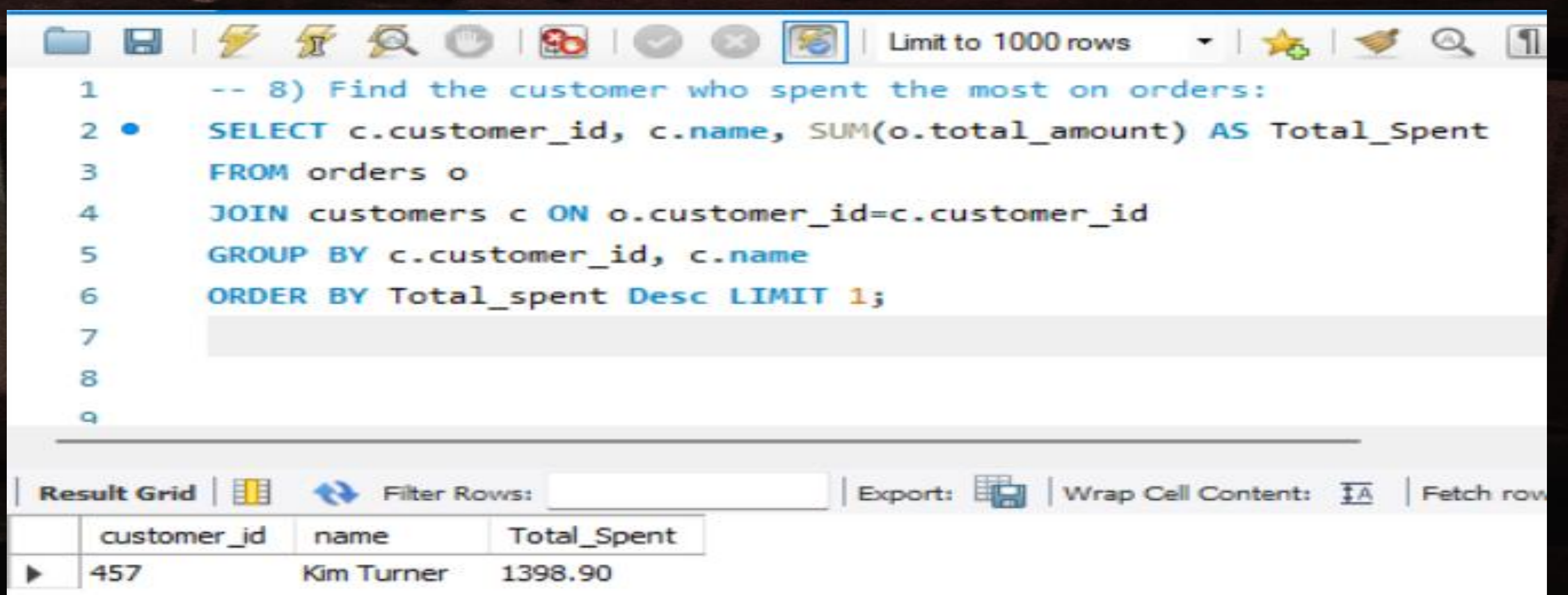
SELECT * FROM ORDERS;

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;
```

The result grid displays the following data:

Genre	Total_Books_sold
Biography	285
Fantasy	446
Science Fiction	447
Mystery	504
Romance	439
Non-Fiction	351
Fiction	225

## 20. Find the customer who spent the most on orders :



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

```
-- 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 customers c ON o.customer_id=c.customer_id
GROUP BY c.customer_id, c.name
ORDER BY Total_spent Desc LIMIT 1;
```

The result grid displays the following data:

customer_id	name	Total_Spent
457	Kim Turner	1398.90



# KEY INSIGHTS

- ❖ Fiction is the most popular genre.
- ❖ November 2023 saw a peak in sales.
- ❖ Canada represents a significant customer base.
- ❖ High-value and bulk orders suggest opportunities for loyalty programs.
- ❖ Identifying low-stock and best-selling books can improve inventory management.
- ❖ Certain authors and books perform exceptionally well.
- ❖ Analyzing financial data supports strategic decisions.
- ❖ Total books sold per genre and the most frequently ordered book were identified.
- ❖ Loyal customers (multiple orders) and top spenders were identified.
- ❖ Cities with the highest-spending customers were revealed.
- ❖ The most expensive book, lowest-stock book, and top 3 priciest Fantasy books were found.



# RECOMMENDATIONS

- ✓ **Focus on Fiction:** Increase inventory and promotional efforts for fiction titles.
- ✓ **Capitalize on November:** Plan marketing campaigns and promotions for the November sales spike.
- ✓ **Target Canadian Market:** Explore localized marketing strategies for customers in Canada.
- ✓ **Implement Loyalty Programs:** Reward repeat customers and high-value purchases.
- ✓ **Optimize Inventory:** Prioritize restocking best-selling and low-stock items.
- ✓ **Highlight Top Performers:** Feature popular authors and books in promotions and displays.
- ✓ **Leverage Financial Data:** Use revenue and profitability analysis for strategic planning.
- ✓ **Genre-Specific Promotions:** Tailor promotions based on genre popularity.
- ✓ **Engage Loyal Customers:** Offer exclusive deals or early access to frequent buyers.
- ✓ **Location-Based Marketing:** Consider targeted campaigns for high-spending cities.
- ✓ **Monitor Book Performance:** Regularly track sales and stock levels of individual titles.



# CONCLUSION

These data-driven insights, derived through rigorous SQL analysis provide a solid foundation for informed decision-making and strategic initiatives aimed at enhancing Online books sales performance. By analyzing key metrics and relationships between orders, Genre, order details, price, and quantity purchased the project seeks to optimize sales strategies, improve operational efficiency, and enhance understanding of customer preferences within the online books selling business.

# THANK YOU

[oyekan.wole@gmail.com](mailto:oyekan.wole@gmail.com)