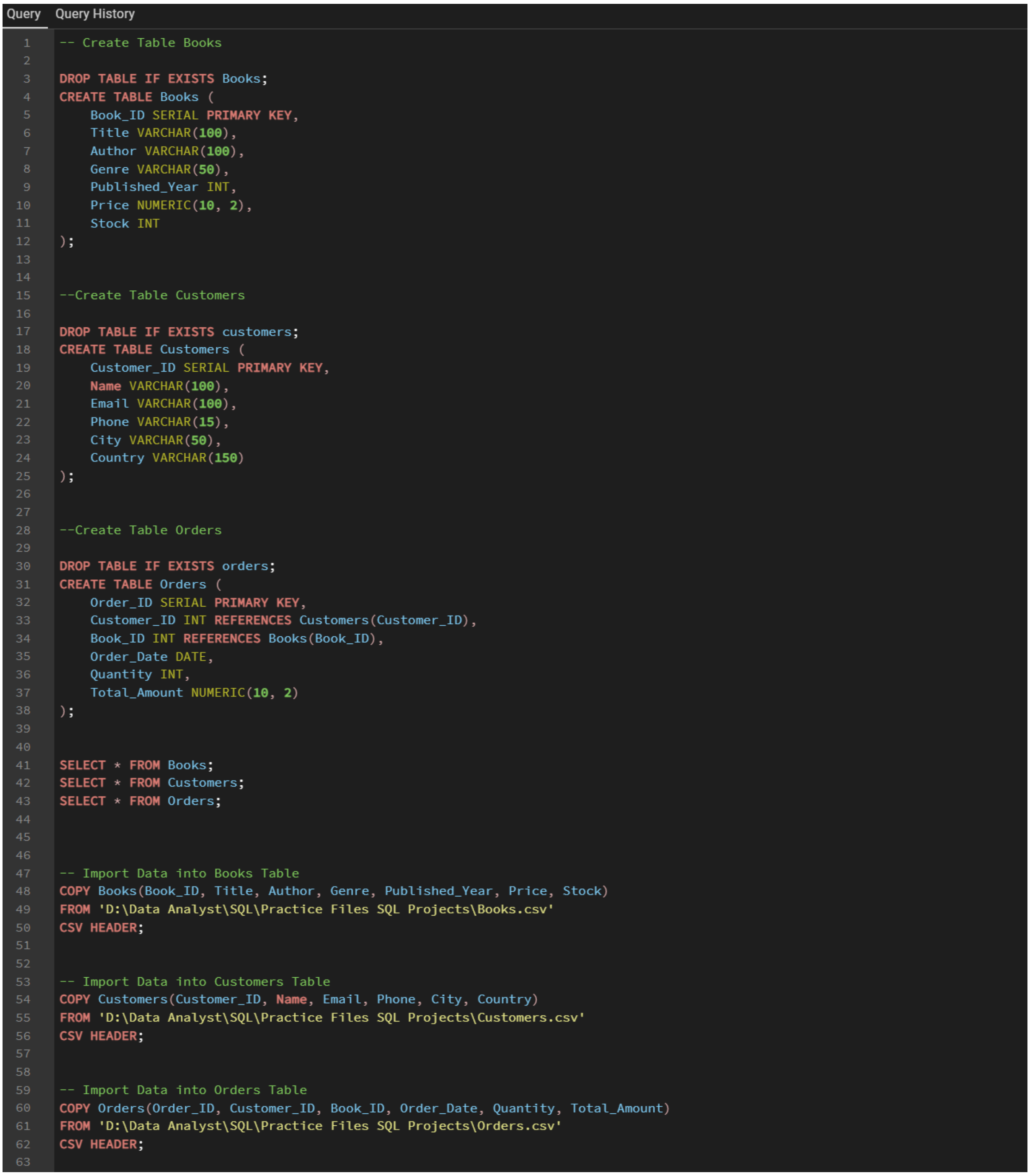
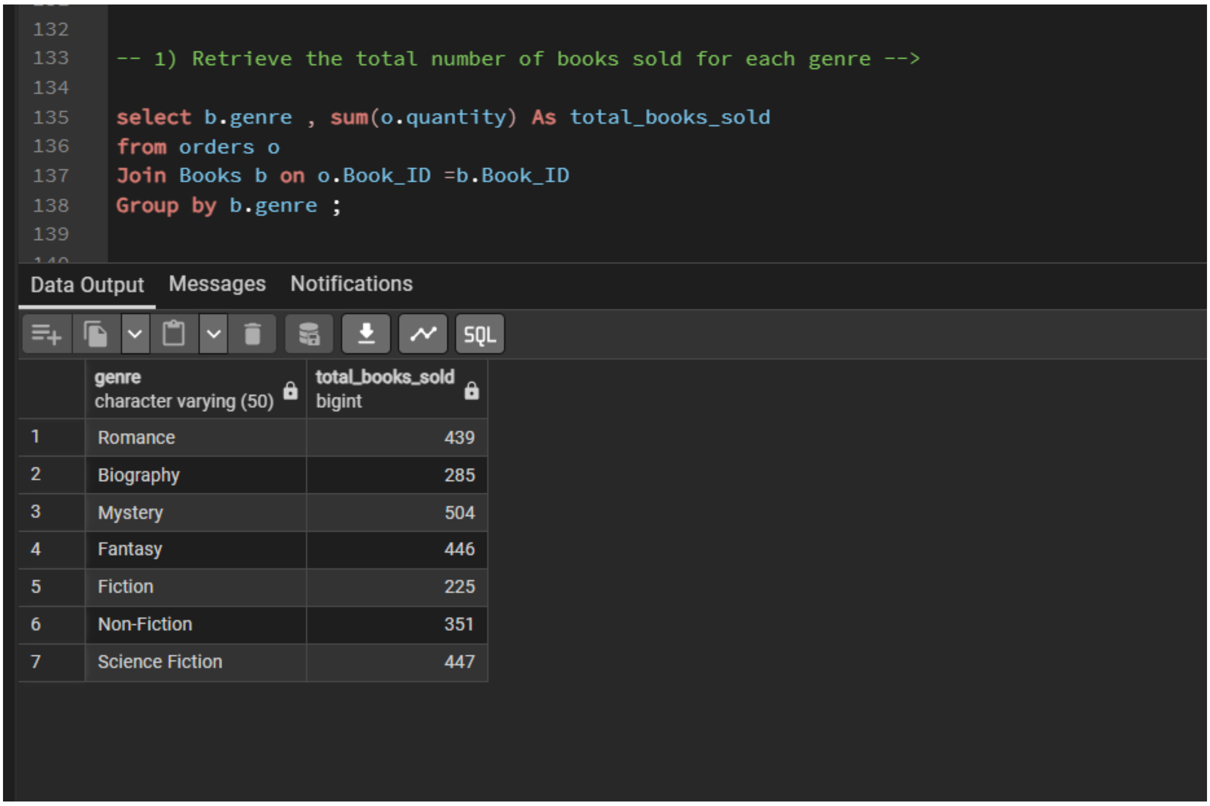
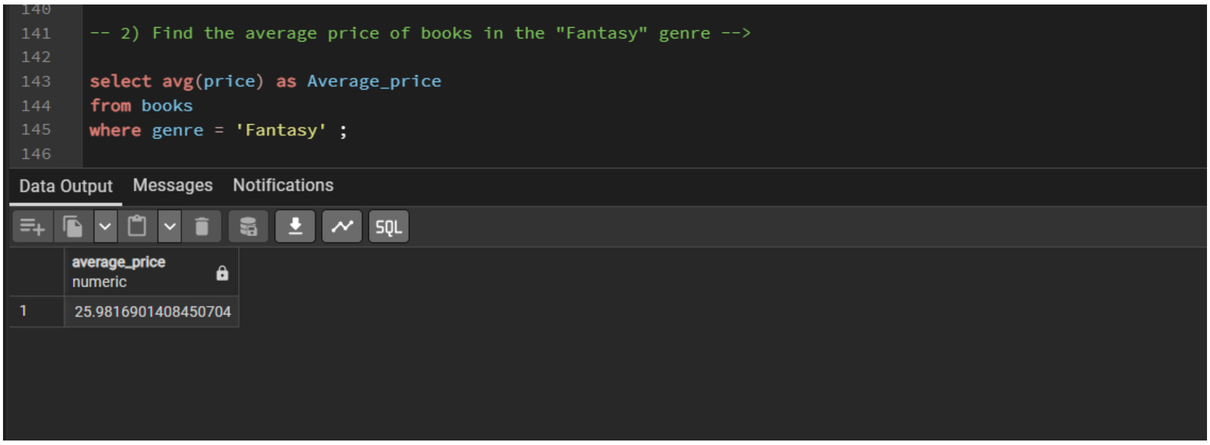


```sql
-- Create Table Books

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
);


--Create Table Customers

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)
);


--Create Table Orders

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)
);


SELECT * FROM Books;
SELECT * FROM Customers;
SELECT * FROM Orders;



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


-- Import Data into Customers Table
COPY Customers(Customer_ID, Name, Email, Phone, City, Country)
FROM 'D:\Data Analyst\SQL\Practice Files SQL Projects\Customers.csv'
CSV HEADER;


-- Import Data into Orders Table
COPY Orders(Order_ID, Customer_ID, Book_ID, Order_Date, Quantity, Total_Amount)
FROM 'D:\Data Analyst\SQL\Practice Files SQL Projects\Orders.csv'
CSV HEADER;

```



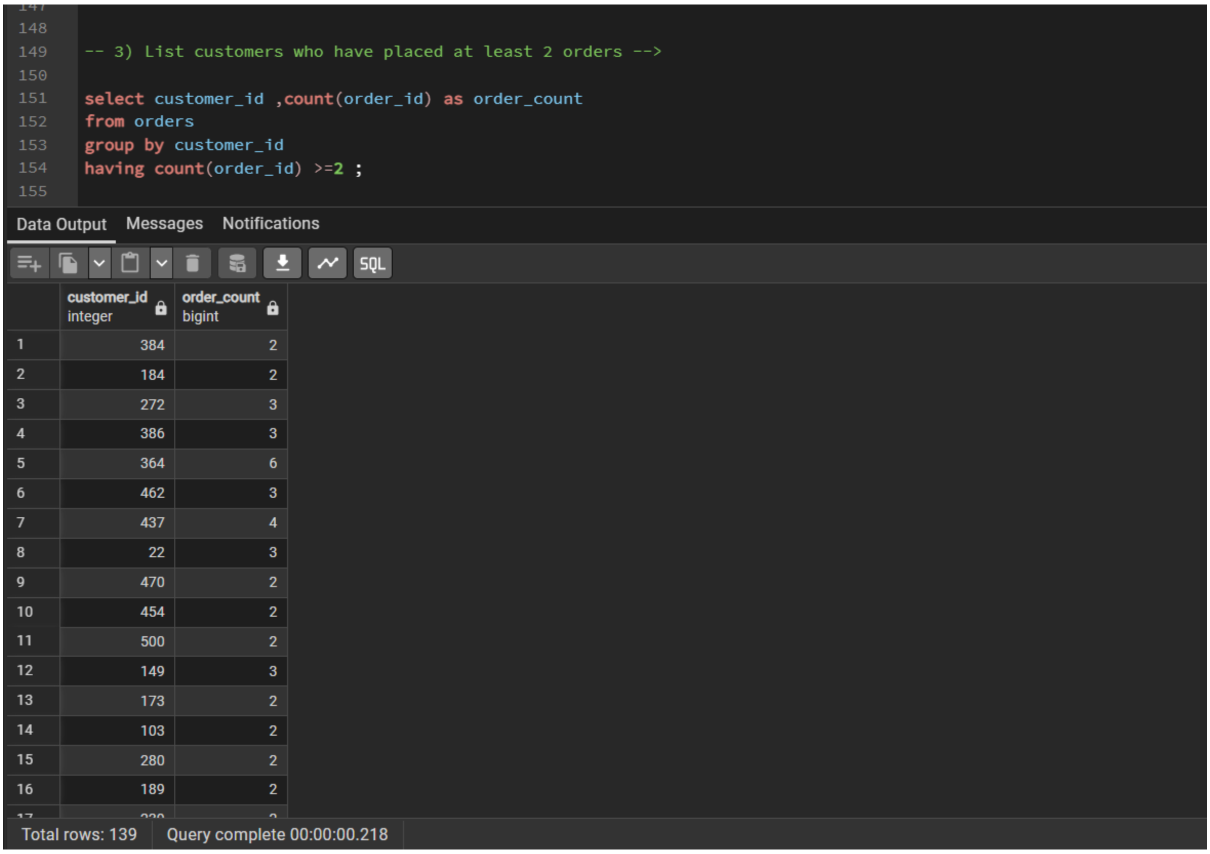
```
-- 1) Retrieve the total number of books sold for each genre -->

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 ;
```

	genre character varying (50)	total_books_sold bigint
1	Romance	439
2	Biography	285
3	Mystery	504
4	Fantasy	446
5	Fiction	225
6	Non-Fiction	351
7	Science Fiction	447



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

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

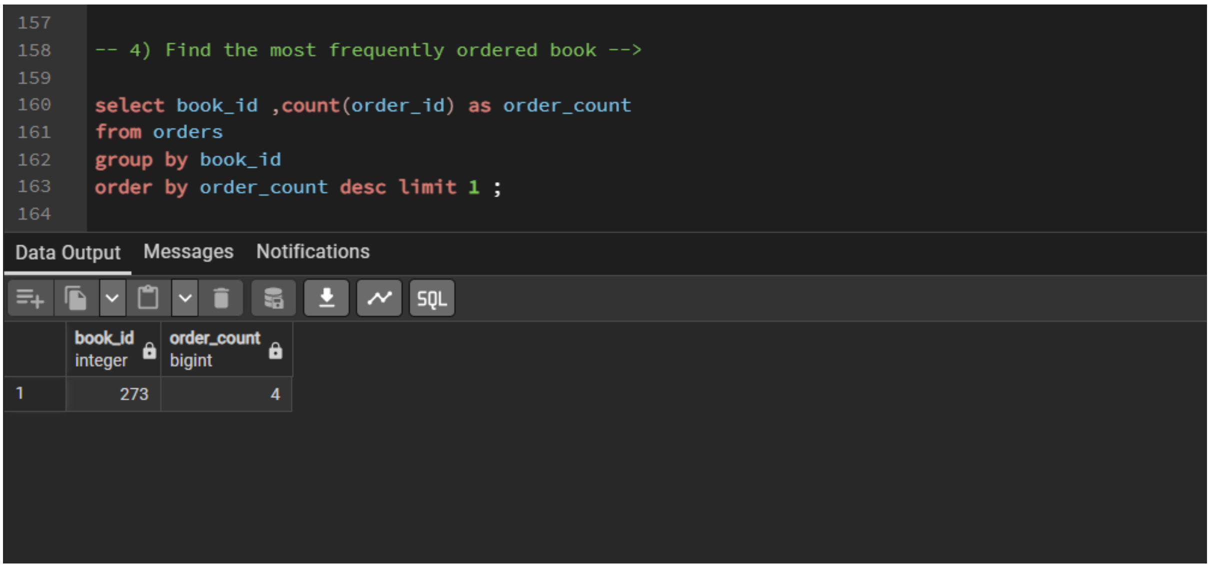
	average_price numeric
1	25.9816901408450704



```
-- 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 ;
```

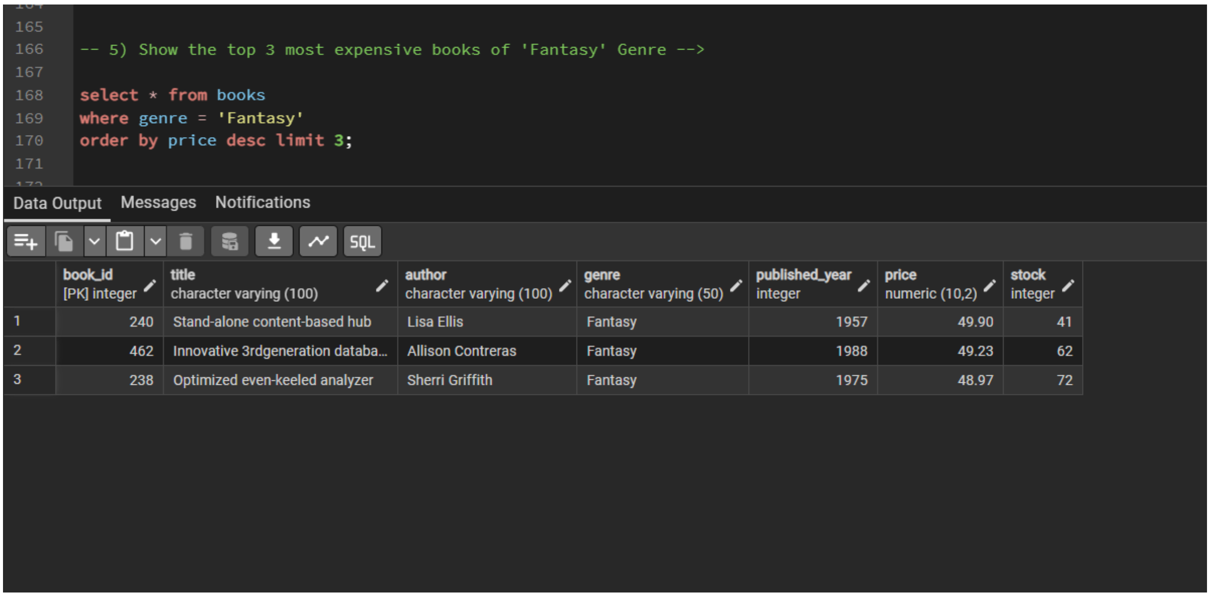
	customer_id integer	order_count bigint
1	384	2
2	184	2
3	272	3
4	386	3
5	364	6
6	462	3
7	437	4
8	22	3
9	470	2
10	454	2
11	500	2
12	149	3
13	173	2
14	103	2
15	280	2
16	189	2

Total rows: 139   Query complete 00:00:00.218



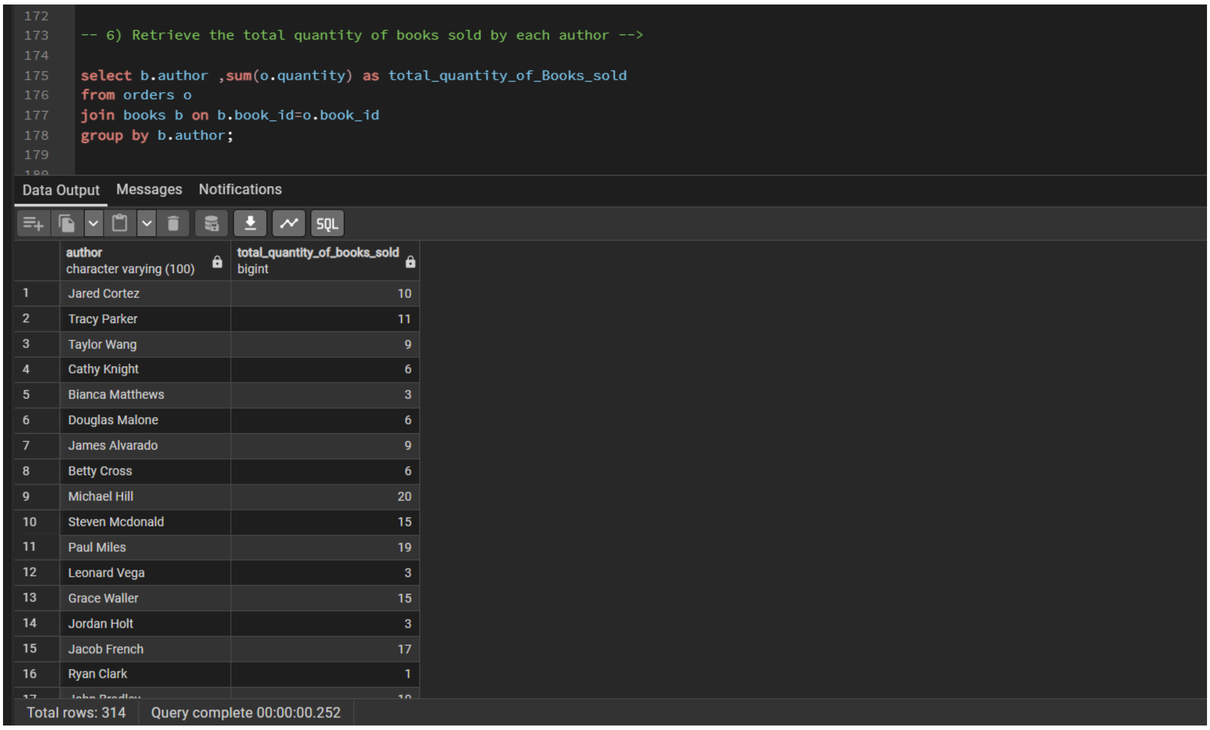
```
-- 4) Find the most frequently ordered book -->

select book_id ,count(order_id) as order_count
from orders
group by book_id
order by order_count desc limit 1 ;
```

	book_id integer	order_count bigint
1	273	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;
```

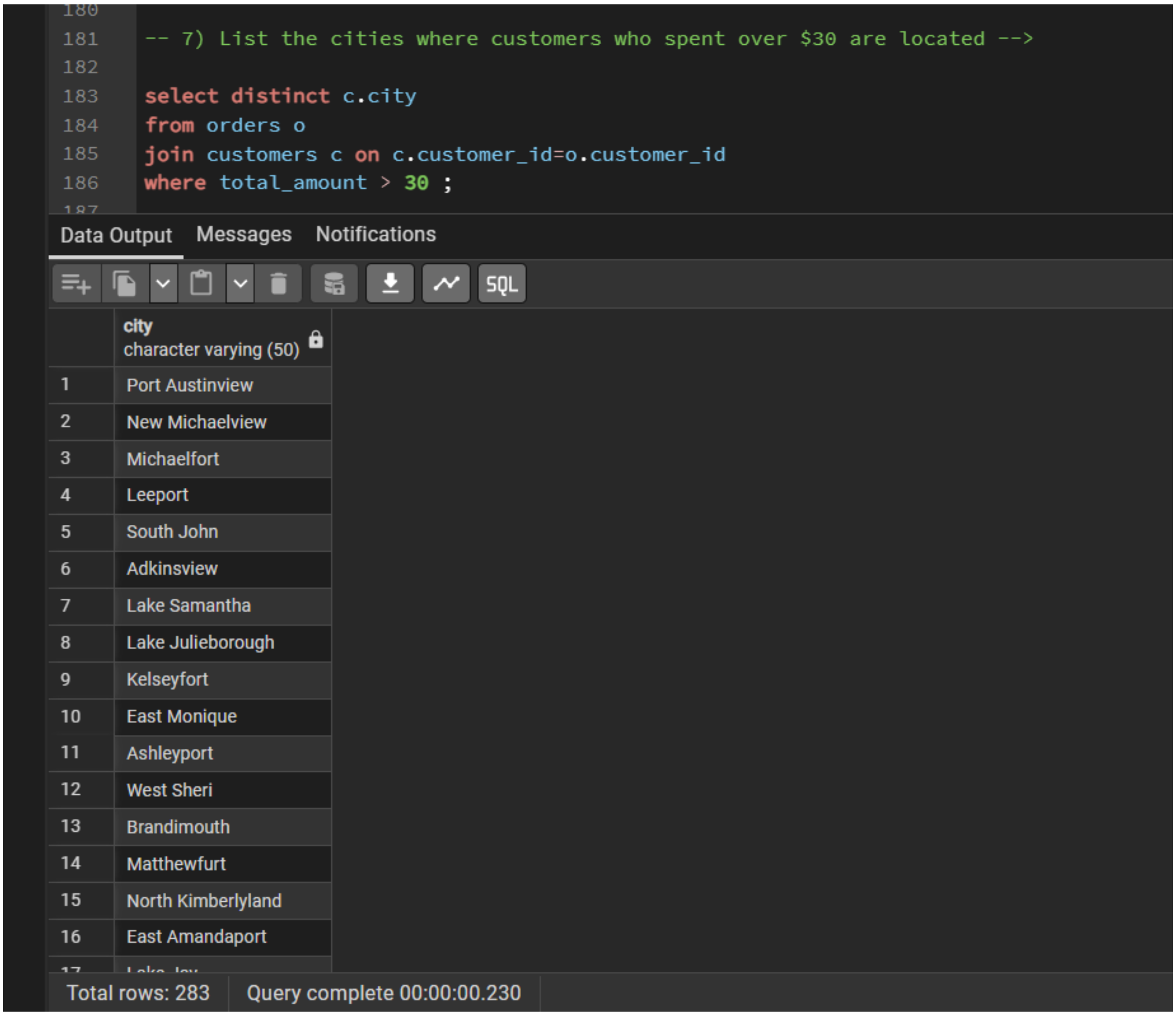
	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	240	Stand-alone content-based hub	Lisa Ellis	Fantasy	1957	49.90	41
2	462	Innovative 3rdgeneration databa...	Allison Contreras	Fantasy	1988	49.23	62
3	238	Optimized even-keeled analyzer	Sherri Griffith	Fantasy	1975	48.97	72



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

select b.author ,sum(o.quantity) as total_quantity_of_Books_sold
from orders o
join books b on b.book_id=o.book_id
group by b.author;
```

	author character varying (100)	total_quantity_of_books_sold bigint
1	Jared Cortez	10
2	Tracy Parker	11
3	Taylor Wang	9
4	Cathy Knight	6
5	Bianca Matthews	3
6	Douglas Malone	6
7	James Alvarado	9
8	Betty Cross	6
9	Michael Hill	20
10	Steven Mcdonald	15
11	Paul Miles	19
12	Leonard Vega	3
13	Grace Waller	15
14	Jordan Holt	3
15	Jacob French	17
16	Ryan Clark	1
17	John Bradley	10

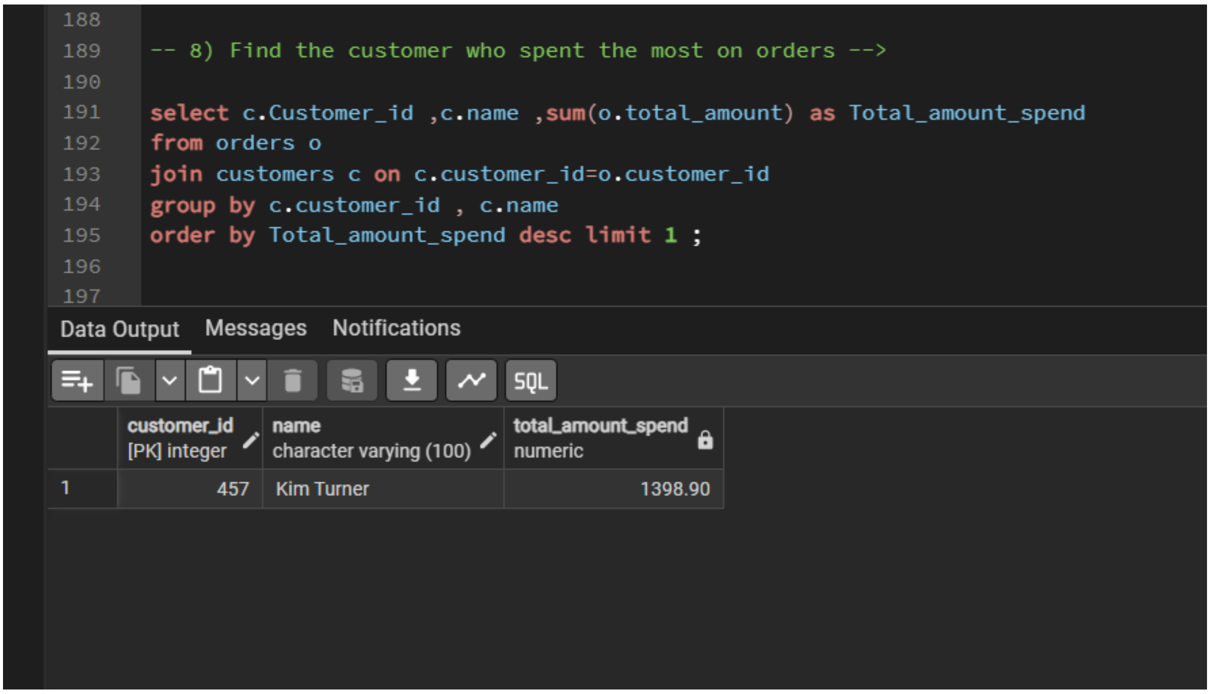
Total rows: 314 Query complete 00:00:00.252



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

select distinct c.city
from orders o
join customers c on c.customer_id=o.customer_id
where total_amount > 30 ;
```

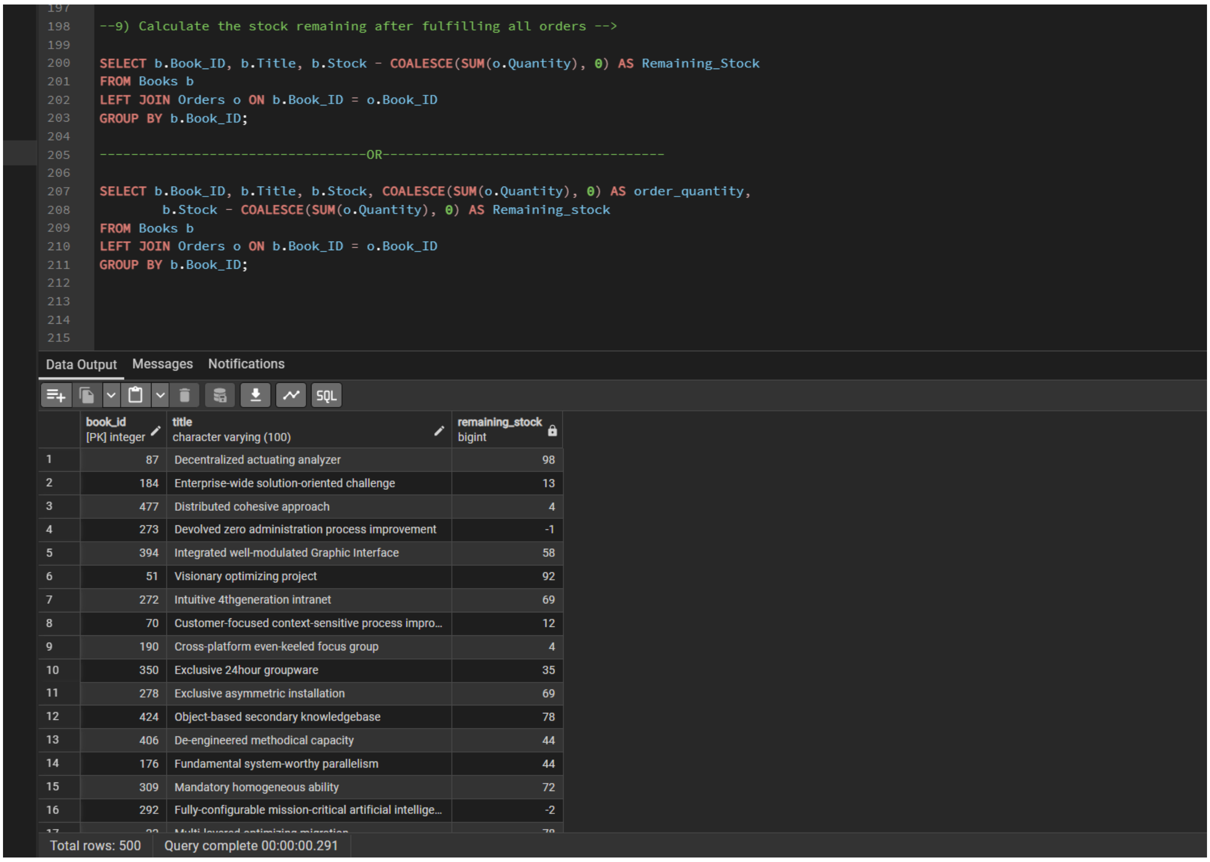
	city character varying (50)
1	Port Austinview
2	New Michaelview
3	Michaelfort
4	Leeport
5	South John
6	Adkinsview
7	Lake Samantha
8	Lake Julieborough
9	Kelseyfort
10	East Monique
11	Ashleyport
12	West Sheri
13	Brandimouth
14	Matthewfurt
15	North Kimberlyland
16	East Amandaport

```
Total rows: 283    Query complete 00:00:00.230
```



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

select c.Customer_id ,c.name ,sum(o.total_amount) as Total_amount_spend
from orders o
join customers c on c.customer_id=o.customer_id
group by c.customer_id , c.name
order by Total_amount_spend desc limit 1 ;
```

	customer_id [PK] integer	name character varying (100)	total_amount_spend numeric
1	457	Kim Turner	1398.90

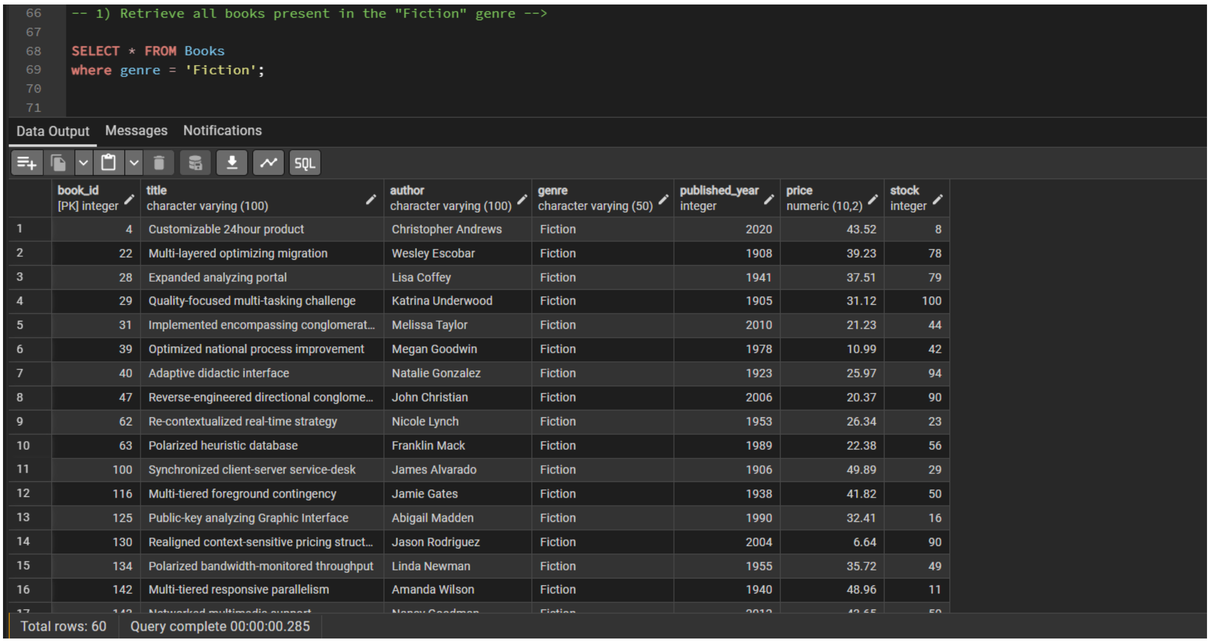


```sql
--9) Calculate the stock remaining after fulfilling all orders -->

SELECT b.Book_ID, b.Title, b.Stock - COALESCE(SUM(o.Quantity), 0) AS Remaining_Stock
FROM Books b
LEFT JOIN Orders o ON b.Book_ID = o.Book_ID
GROUP BY b.Book_ID;

--------------------------------OR----------------------------------

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_stock
FROM Books b
LEFT JOIN Orders o ON b.Book_ID = o.Book_ID
GROUP BY b.Book_ID;
```

	book_id [PK] integer	title character varying (100)	remaining_stock bigint
1	87	Decentralized actuating analyzer	98
2	184	Enterprise-wide solution-oriented challenge	13
3	477	Distributed cohesive approach	4
4	273	Devolved zero administration process improvement	-1
5	394	Integrated well-modulated Graphic Interface	58
6	51	Visionary optimizing project	92
7	272	Intuitive 4thgeneration intranet	69
8	70	Customer-focused context-sensitive process impro...	12
9	190	Cross-platform even-keeled focus group	4
10	350	Exclusive 24hour groupware	35
11	278	Exclusive asymmetric installation	69
12	424	Object-based secondary knowledgebase	78
13	406	De-engineered methodical capacity	44
14	176	Fundamental system-worthy parallelism	44
15	309	Mandatory homogeneous ability	72
16	292	Fully-configurable mission-critical artificial intellige...	-2

Total rows: 500 Query complete 00:00:00.291

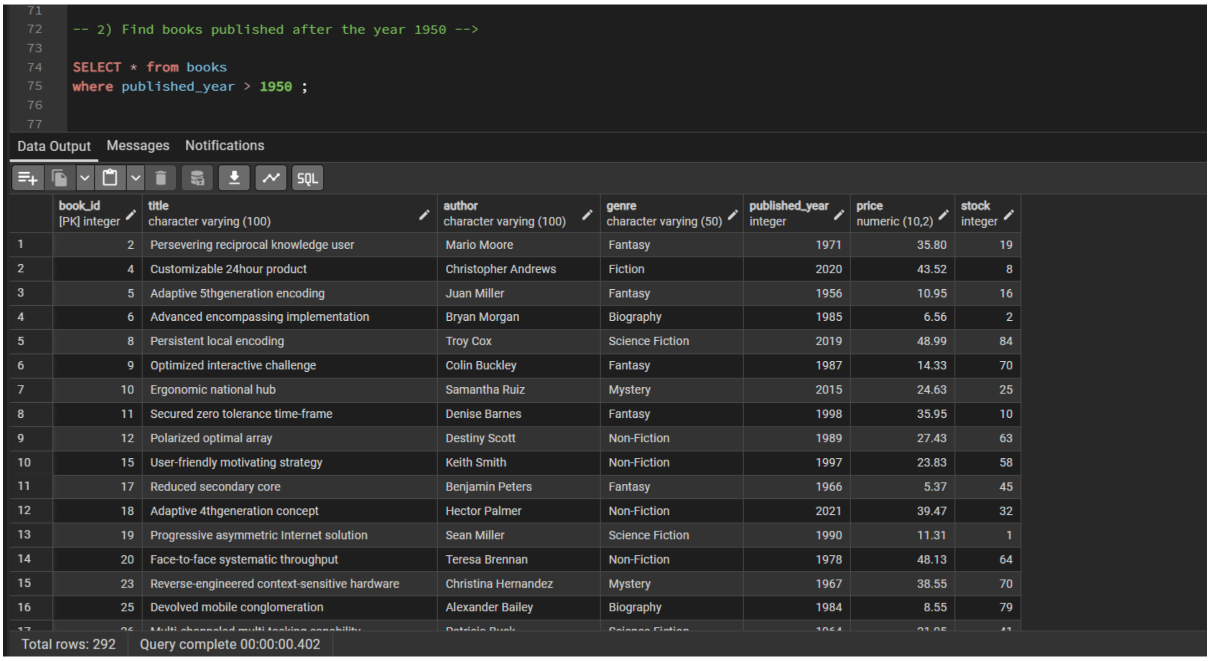
Simple Queries :



```sql
-- 1) Retrieve all books present in the "Fiction" genre -->

SELECT * FROM Books
where genre = 'Fiction';
```

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
2	22	Multi-layered optimizing migration	Wesley Escobar	Fiction	1908	39.23	78
3	28	Expanded analyzing portal	Lisa Coffey	Fiction	1941	37.51	79
4	29	Quality-focused multi-tasking challenge	Katrina Underwood	Fiction	1905	31.12	100
5	31	Implemented encompassing conglomerat...	Melissa Taylor	Fiction	2010	21.23	44
6	39	Optimized national process improvement	Megan Goodwin	Fiction	1978	10.99	42
7	40	Adaptive didactic interface	Natalie Gonzalez	Fiction	1923	25.97	94
8	47	Reverse-engineered directional conglome...	John Christian	Fiction	2006	20.37	90
9	62	Re-contextualized real-time strategy	Nicole Lynch	Fiction	1953	26.34	23
10	63	Polarized heuristic database	Franklin Mack	Fiction	1989	22.38	56
11	100	Synchronized client-server service-desk	James Alvarado	Fiction	1906	49.89	29
12	116	Multi-tiered foreground contingency	Jamie Gates	Fiction	1938	41.82	50
13	125	Public-key analyzing Graphic Interface	Abigail Madden	Fiction	1990	32.41	16
14	130	Realigned context-sensitive pricing struct...	Jason Rodriguez	Fiction	2004	6.64	90
15	134	Polarized bandwidth-monitored throughput	Linda Newman	Fiction	1955	35.72	49
16	142	Multi-tiered responsive parallelism	Amanda Wilson	Fiction	1940	48.96	11

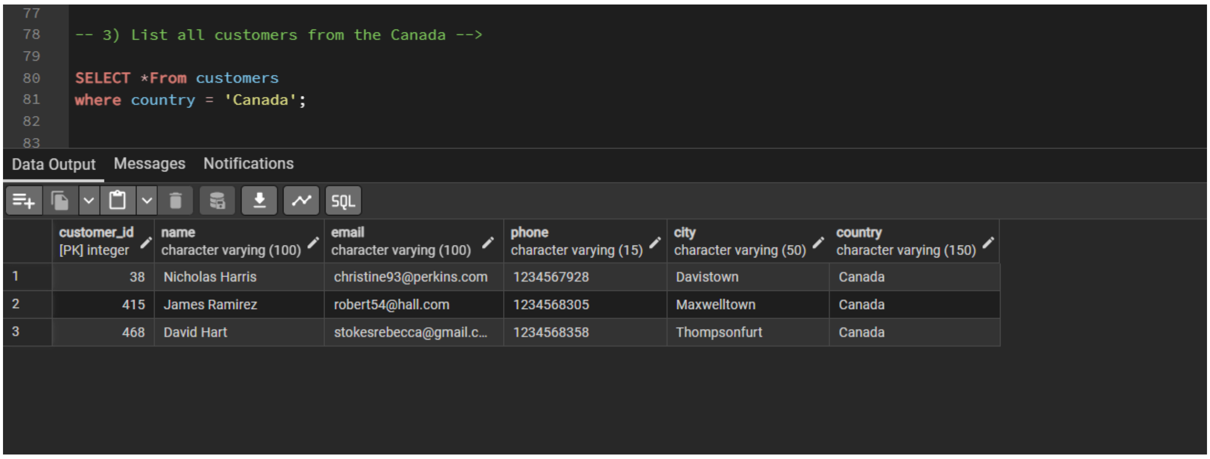
Total rows: 60 Query complete 00:00:00.285



```sql
-- 2) Find books published after the year 1950 -->

SELECT * from books
where published_year > 1950 ;
```

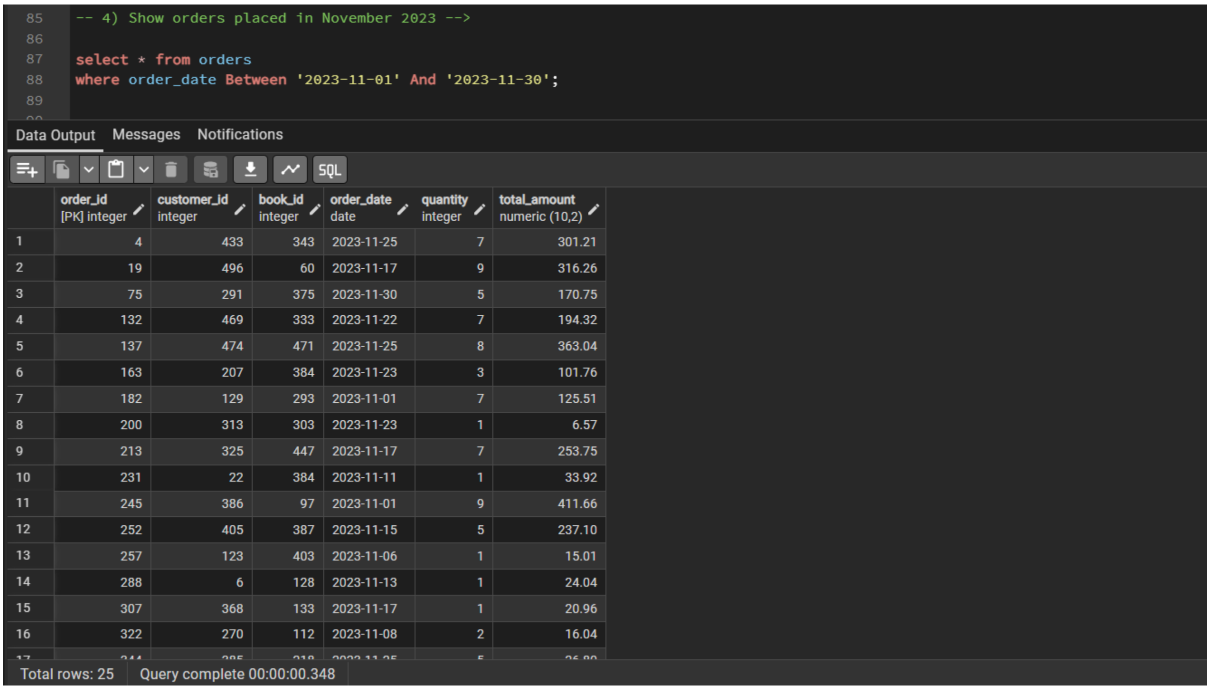
	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19
2	4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
3	5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	10.95	16
4	6	Advanced encompassing implementation	Bryan Morgan	Biography	1985	6.56	2
5	8	Persistent local encoding	Troy Cox	Science Fiction	2019	48.99	84
6	9	Optimized interactive challenge	Colin Buckley	Fantasy	1987	14.33	70
7	10	Ergonomic national hub	Samantha Ruiz	Mystery	2015	24.63	25
8	11	Secured zero tolerance time-frame	Denise Barnes	Fantasy	1998	35.95	10
9	12	Polarized optimal array	Destiny Scott	Non-Fiction	1989	27.43	63
10	15	User-friendly motivating strategy	Keith Smith	Non-Fiction	1997	23.83	58
11	17	Reduced secondary core	Benjamin Peters	Fantasy	1966	5.37	45
12	18	Adaptive 4thgeneration concept	Hector Palmer	Non-Fiction	2021	39.47	32
13	19	Progressive asymmetric Internet solution	Sean Miller	Science Fiction	1990	11.31	1
14	20	Face-to-face systematic throughput	Teresa Brennan	Non-Fiction	1978	48.13	64
15	23	Reverse-engineered context-sensitive hardware	Christina Hernandez	Mystery	1967	38.55	70
16	25	Devolved mobile conglomeration	Alexander Bailey	Biography	1984	8.55	79
17	26	Multi-channeled multi-tasking capability	Patricia Buck	Science Fiction	1964	21.05	41

Total rows: 292 Query complete 00:00:00.402



```sql
-- 3) List all customers from the Canada -->

SELECT *From customers
where country = 'Canada';
```

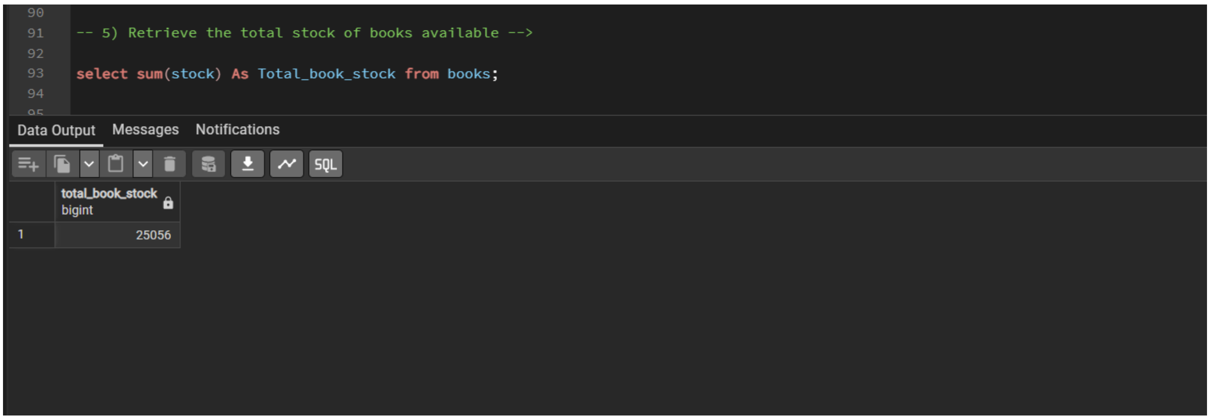
	customer_id [PK] integer	name character varying (100)	email character varying (100)	phone character varying (15)	city character varying (50)	country character varying (150)
1	38	Nicholas Harris	christine93@perkins.com	1234567928	Davistown	Canada
2	415	James Ramirez	robert54@hall.com	1234568305	Maxwelltown	Canada
3	468	David Hart	stokesrebecca@gmail.c...	1234568358	Thompsonfurt	Canada



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

select * from orders
where order_date Between '2023-11-01' And '2023-11-30';
```

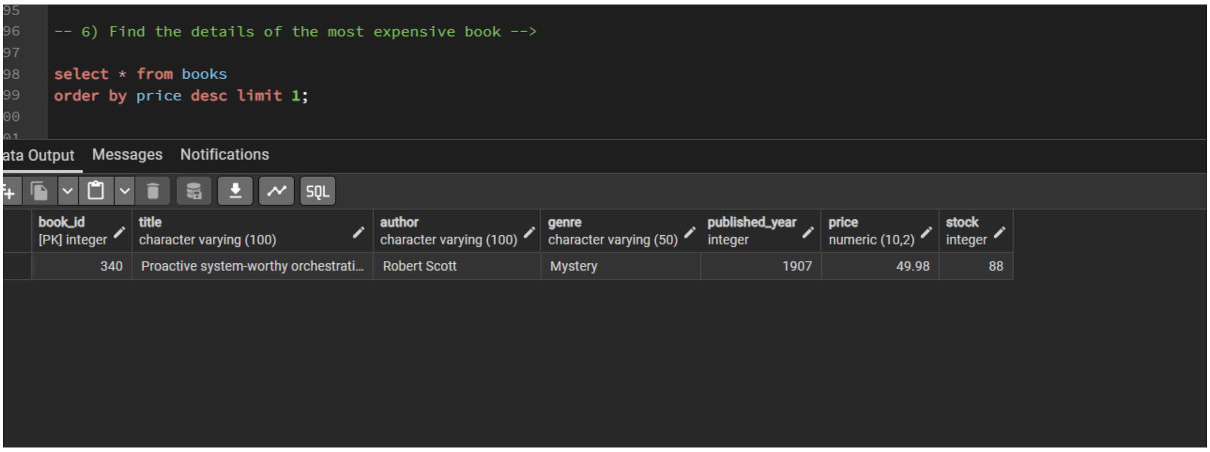
	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_amount numeric (10,2)
1	4	433	343	2023-11-25	7	301.21
2	19	496	60	2023-11-17	9	316.26
3	75	291	375	2023-11-30	5	170.75
4	132	469	333	2023-11-22	7	194.32
5	137	474	471	2023-11-25	8	363.04
6	163	207	384	2023-11-23	3	101.76
7	182	129	293	2023-11-01	7	125.51
8	200	313	303	2023-11-23	1	6.57
9	213	325	447	2023-11-17	7	253.75
10	231	22	384	2023-11-11	1	33.92
11	245	386	97	2023-11-01	9	411.66
12	252	405	387	2023-11-15	5	237.10
13	257	123	403	2023-11-06	1	15.01
14	288	6	128	2023-11-13	1	24.04
15	307	368	133	2023-11-17	1	20.96
16	322	270	112	2023-11-08	2	16.04

```
Total rows: 25    Query complete 00:00:00.348
```



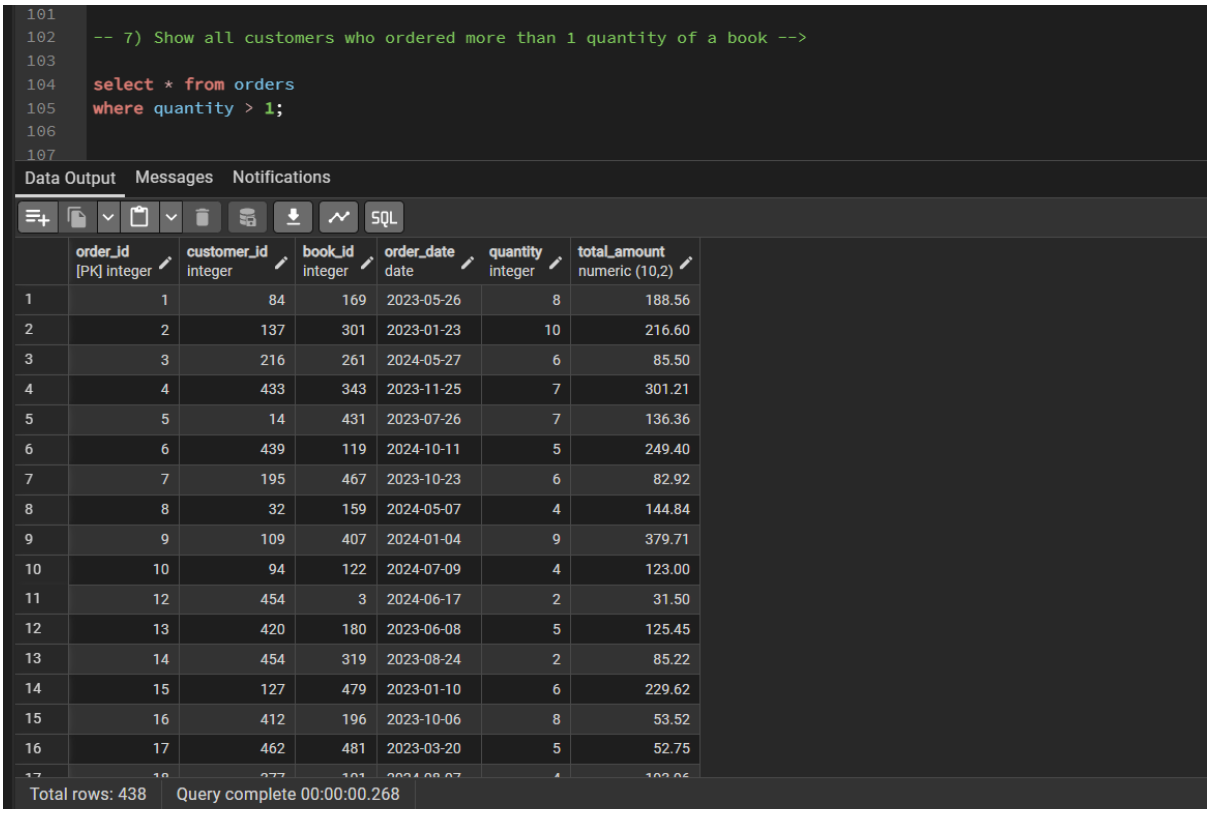
```
-- 5) Retrieve the total stock of books available -->

select sum(stock) As Total_book_stock from books;
```

	total_book_stock bigint
1	25056



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

select * from books
order by price desc limit 1;
```

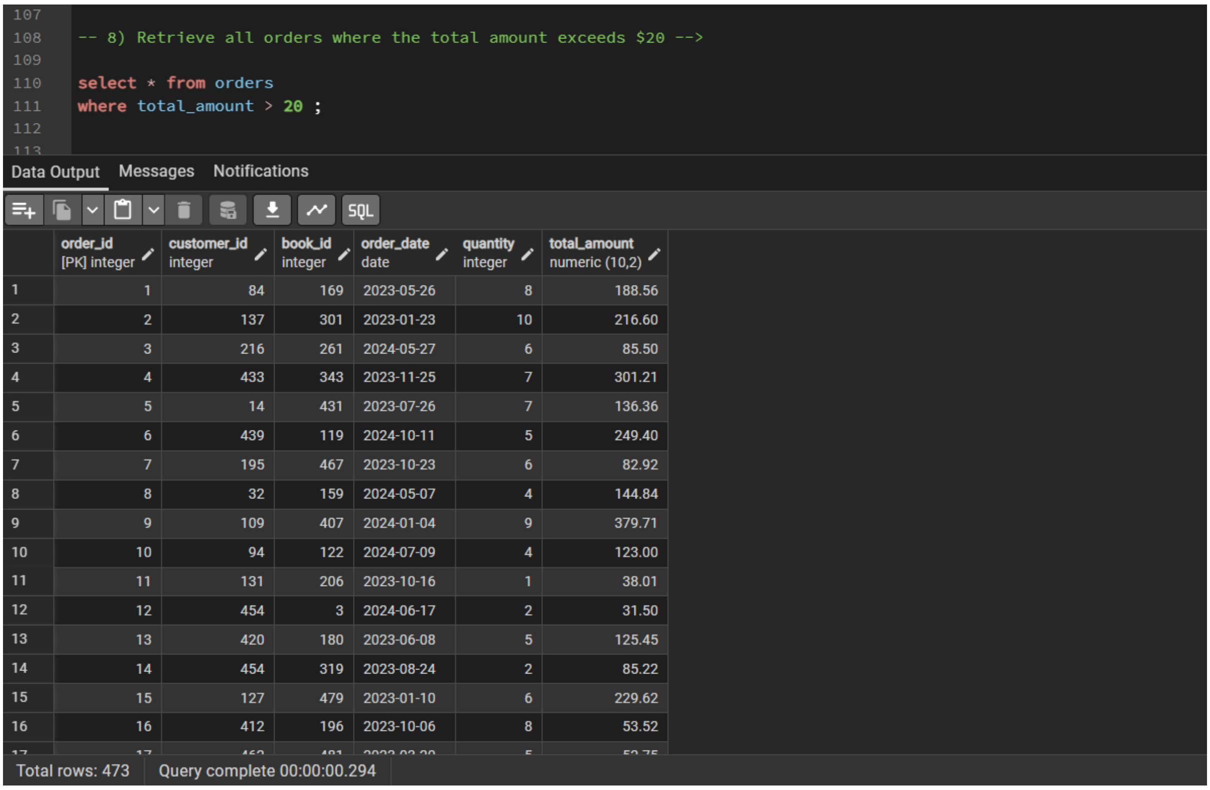
book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
340	Proactive system-worthy orchestrati...	Robert Scott	Mystery	1907	49.98	88



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

select * from orders
where quantity > 1;
```

	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_amount numeric (10,2)
1	1	84	169	2023-05-26	8	188.56
2	2	137	301	2023-01-23	10	216.60
3	3	216	261	2024-05-27	6	85.50
4	4	433	343	2023-11-25	7	301.21
5	5	14	431	2023-07-26	7	136.36
6	6	439	119	2024-10-11	5	249.40
7	7	195	467	2023-10-23	6	82.92
8	8	32	159	2024-05-07	4	144.84
9	9	109	407	2024-01-04	9	379.71
10	10	94	122	2024-07-09	4	123.00
11	12	454	3	2024-06-17	2	31.50
12	13	420	180	2023-06-08	5	125.45
13	14	454	319	2023-08-24	2	85.22
14	15	127	479	2023-01-10	6	229.62
15	16	412	196	2023-10-06	8	53.52
16	17	462	481	2023-03-20	5	52.75

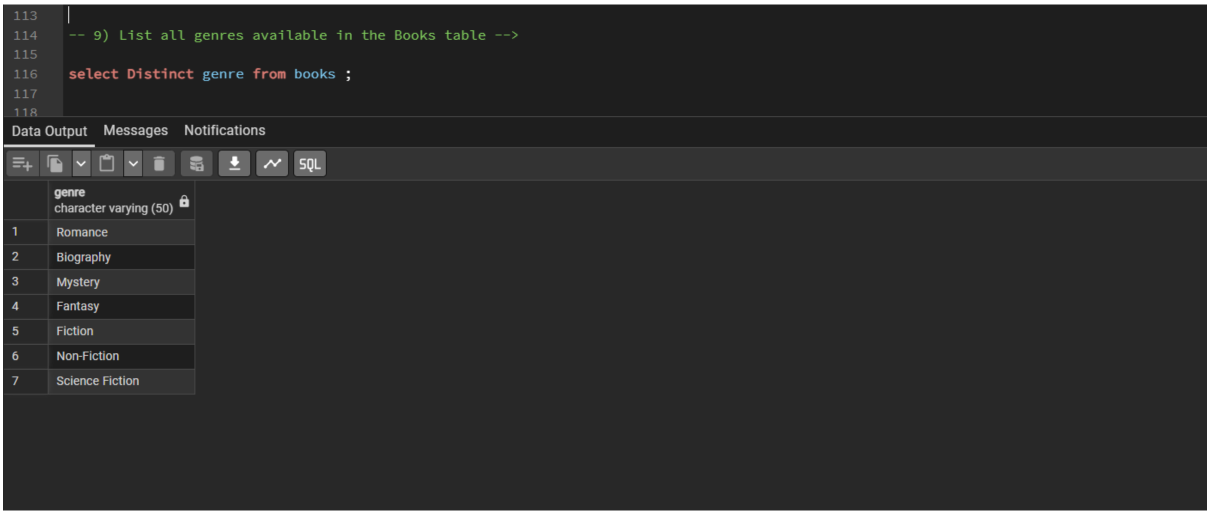
Total rows: 438 Query complete 00:00:00.268



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

select * from orders
where total_amount > 20 ;
```

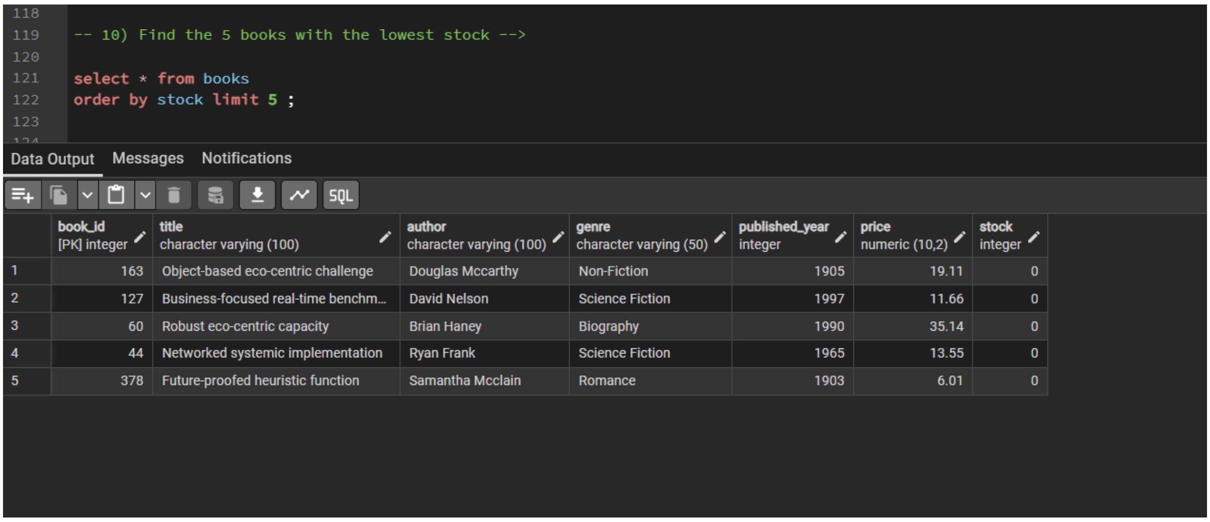
	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_amount numeric (10,2)
1	1	84	169	2023-05-26	8	188.56
2	2	137	301	2023-01-23	10	216.60
3	3	216	261	2024-05-27	6	85.50
4	4	433	343	2023-11-25	7	301.21
5	5	14	431	2023-07-26	7	136.36
6	6	439	119	2024-10-11	5	249.40
7	7	195	467	2023-10-23	6	82.92
8	8	32	159	2024-05-07	4	144.84
9	9	109	407	2024-01-04	9	379.71
10	10	94	122	2024-07-09	4	123.00
11	11	131	206	2023-10-16	1	38.01
12	12	454	3	2024-06-17	2	31.50
13	13	420	180	2023-06-08	5	125.45
14	14	454	319	2023-08-24	2	85.22
15	15	127	479	2023-01-10	6	229.62
16	16	412	196	2023-10-06	8	53.52

Total rows: 473 Query complete 00:00:00.294



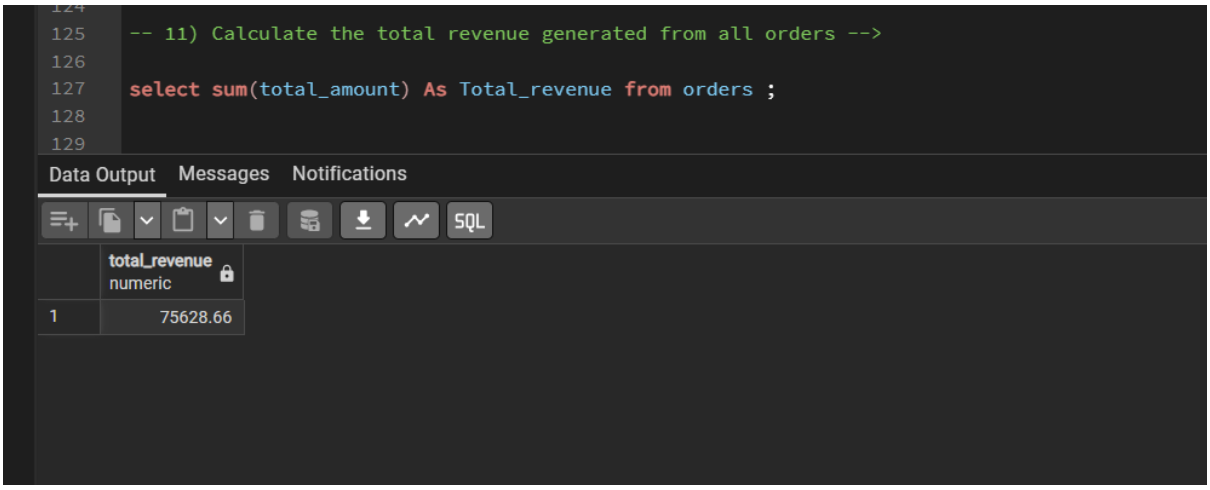
```
-- 9) List all genres available in the Books table -->

select Distinct genre from books ;
```

	genre character varying (50)
1	Romance
2	Biography
3	Mystery
4	Fantasy
5	Fiction
6	Non-Fiction
7	Science Fiction



```
-- 10) Find the 5 books with the lowest stock -->

select * from books
order by stock limit 5 ;
```

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	163	Object-based eco-centric challenge	Douglas Mccarthy	Non-Fiction	1905	19.11	0
2	127	Business-focused real-time benchm...	David Nelson	Science Fiction	1997	11.66	0
3	60	Robust eco-centric capacity	Brian Haney	Biography	1990	35.14	0
4	44	Networked systemic implementation	Ryan Frank	Science Fiction	1965	13.55	0
5	378	Future-proofed heuristic function	Samantha Mcclain	Romance	1903	6.01	0



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

select sum(total_amount) As Total_revenue from orders ;
```

	total_revenue numeric
1	75628.66