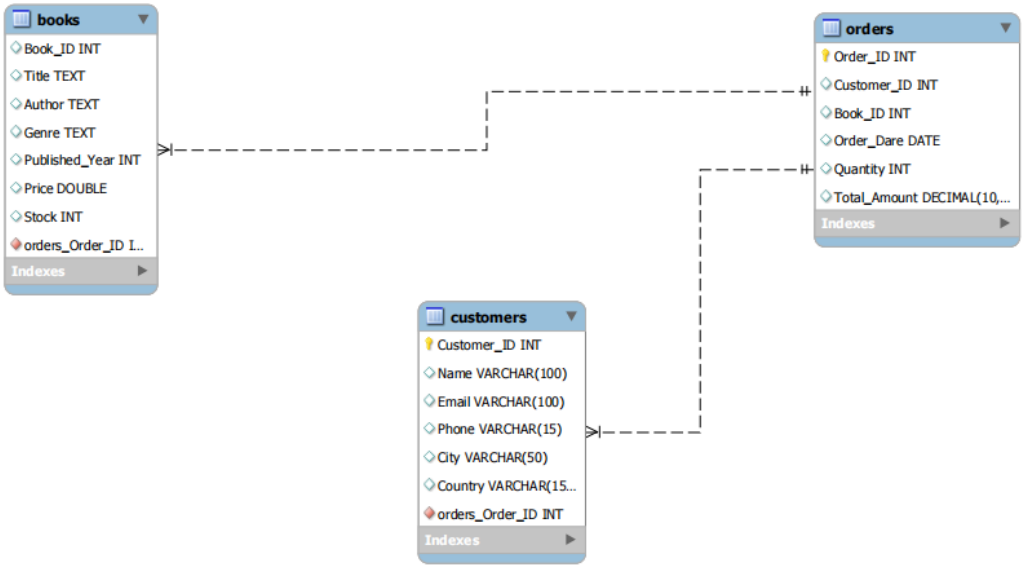
ONLINE BOOK STORE

ER-Diagram



STRUCTURE OF THE TABLE

Books Table

CREATE TABLE Books (

Book_ID INT PRIMARY KEY,

Title VARCHAR(100),

Author VARCHAR(100),

Genre VARCHAR(50),

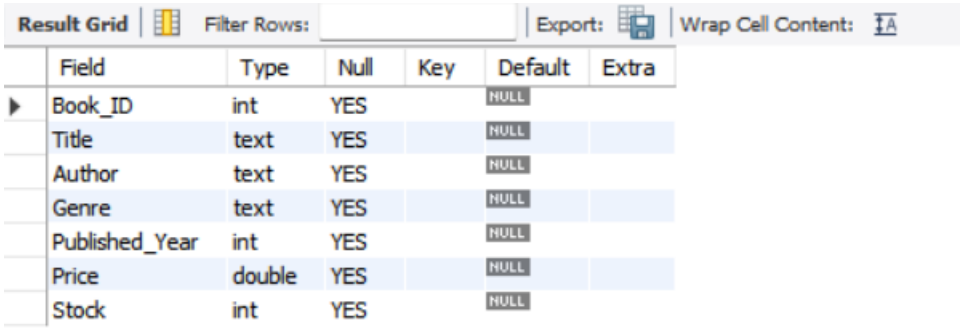
Published_Year INT,

Price NUMERIC(10,2),

Stock int

);

Syntax:- Desc books;



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

Field	Type	Null	Key	Default	Extra
Book_ID	int	YES		NULL	
Title	text	YES		NULL	
Author	text	YES		NULL	
Genre	text	YES		NULL	
Published_Year	int	YES		NULL	
Price	double	YES		NULL	
Stock	int	YES		NULL	

Customers Table

CREATE TABLE Customers (

Customer_ID INT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100),

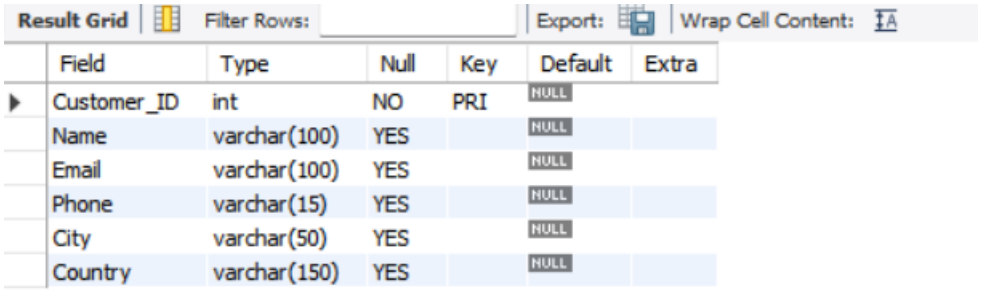
Phone VARCHAR(15),

City VARCHAR(50),

Country VARCHAR(150)

);

Syntax:- Desc Customers;



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

Field	Type	Null	Key	Default	Extra
Customer_ID	int	NO	PRI	NULL	
Name	varchar(100)	YES		NULL	
Email	varchar(100)	YES		NULL	
Phone	varchar(15)	YES		NULL	
City	varchar(50)	YES		NULL	
Country	varchar(150)	YES		NULL	

Orders Table

CREATE TABLE Orders (

Order_ID INT PRIMARY KEY,

Customer_ID INT REFERENCES Customers(Customer_ID),

Book_ID INT REFERENCES Books(Book_ID),

Order_Dare DATE,

Quantity INT,

Total_Amount NUMERIC(10,2)

);

Syntax:- Desc Orders;

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

Field	Type	Null	Key	Default	Extra
Order_ID	int	NO	PRI	NULL	
Customer_ID	int	YES		NULL	
Book_ID	int	YES		NULL	
Order_Dare	date	YES		NULL	
Quantity	int	YES		NULL	
Total_Amount	decimal(10,2)	YES		NULL	

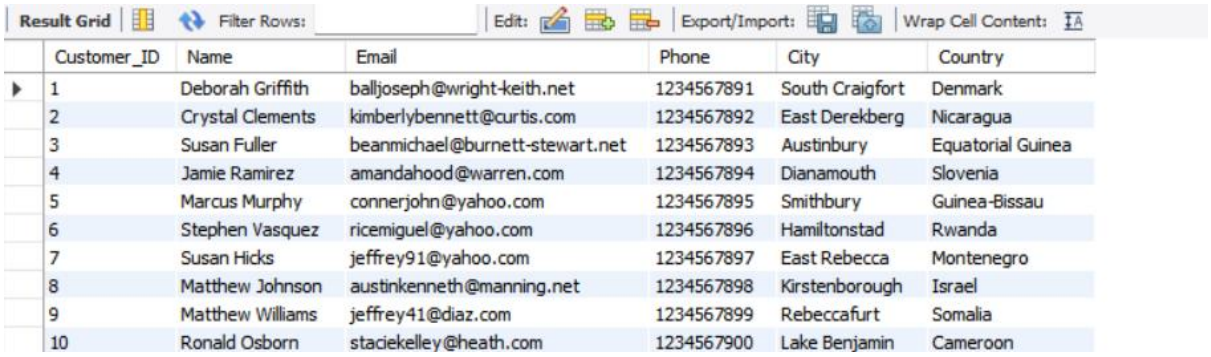
CONTENTS OF THE TABLE

SELECT * FROM Books;

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

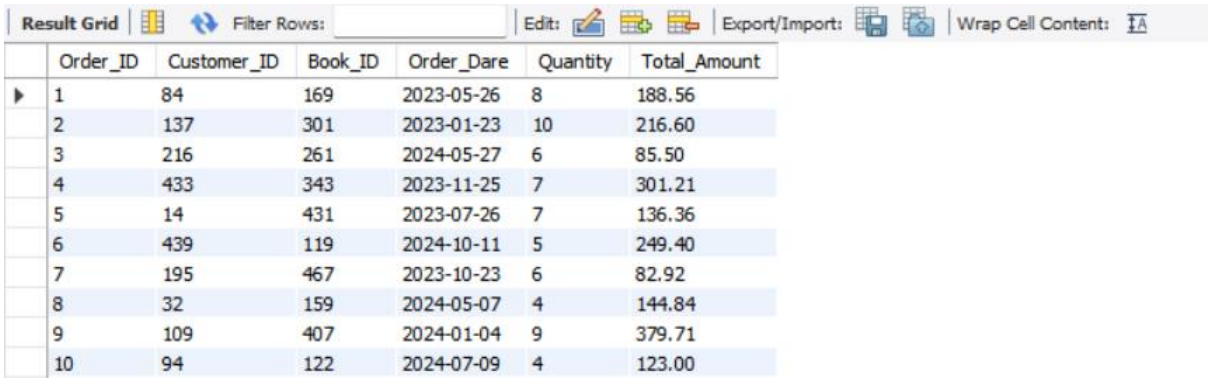
Book_ID	Title	Author	Genre	Published_Year	Price	Stock
1	Configurable modular throughput	Joseph Crane	Biography	1949	21.34	100
2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.8	19
3	Streamlined coherent initiative	Derrick Howard	Non-Fiction	1913	15.75	27
4	Customizable 24hour product	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
7	Open-architected exuding structure	Jacqueline Young	Romance	1927	43.63	95
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

SELECT * FROM Customers;



Customer_ID	Name	Email	Phone	City	Country
1	Deborah Griffith	balljoseph@wright-keith.net	1234567891	South Craigfort	Denmark
2	Crystal Clements	kimberlybennett@curtis.com	1234567892	East Derekberg	Nicaragua
3	Susan Fuller	beanmichael@burnett-stewart.net	1234567893	Austinbury	Equatorial Guinea
4	Jamie Ramirez	amandahood@warren.com	1234567894	Dianamouth	Slovenia
5	Marcus Murphy	connerjohn@yahoo.com	1234567895	Smithbury	Guinea-Bissau
6	Stephen Vasquez	ricemiguel@yahoo.com	1234567896	Hamiltonstad	Rwanda
7	Susan Hicks	jeffrey91@yahoo.com	1234567897	East Rebecca	Montenegro
8	Matthew Johnson	austinkenneth@manning.net	1234567898	Kirstenborough	Israel
9	Matthew Williams	jeffrey41@diaz.com	1234567899	Rebeccafurt	Somalia
10	Ronald Osborn	staciekelley@heath.com	1234567900	Lake Benjamin	Cameroon

SELECT * FROM Orders;



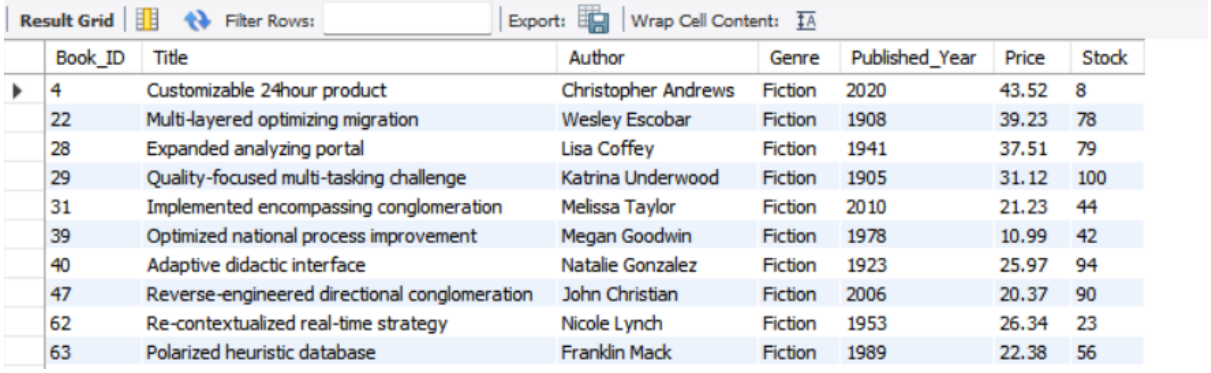
Order_ID	Customer_ID	Book_ID	Order_Dare	Quantity	Total_Amount
1	84	169	2023-05-26	8	188.56
2	137	301	2023-01-23	10	216.60
3	216	261	2024-05-27	6	85.50
4	433	343	2023-11-25	7	301.21
5	14	431	2023-07-26	7	136.36
6	439	119	2024-10-11	5	249.40
7	195	467	2023-10-23	6	82.92
8	32	159	2024-05-07	4	144.84
9	109	407	2024-01-04	9	379.71
10	94	122	2024-07-09	4	123.00

SUB QUERIES

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

Find books published after the year 1950

```
SELECT * FROM Books

WHERE Published_year>1950;
```

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

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.8	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

List all customers from the Canada

```
SELECT * FROM Customers

WHERE country='Canada';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

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
NULL	NULL	NULL	NULL	NULL	NULL

Retrieve the total stock of books available

```
SELECT SUM(stock) AS Total_Stock

From Books;
```

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

Total_Stock
25056

Find the details of the most expensive book

```
SELECT * FROM Books

ORDER BY Price DESC

LIMIT 1;
```

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

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

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

SELECT * FROM Orders

WHERE quantity>1;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Order_ID	Customer_ID	Book_ID	Order_Dare	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

Retrieve all orders where the total amount exceeds $20

SELECT * FROM Orders

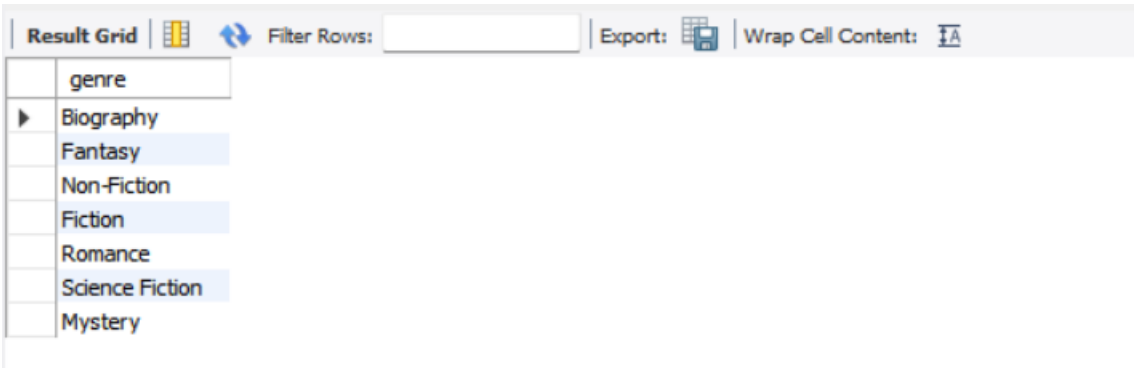
WHERE total_amount>20;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Order_ID	Customer_ID	Book_ID	Order_Dare	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

List all genres available in the Books table

SELECT DISTINCT genre FROM Books;



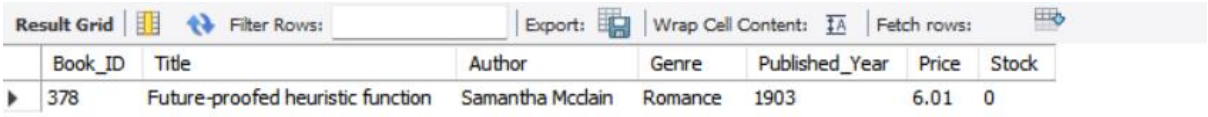
genre
Biography
Fantasy
Non-Fiction
Fiction
Romance
Science Fiction
Mystery

Find the book with the lowest stock

SELECT * FROM Books

ORDER BY stock

LIMIT 1;

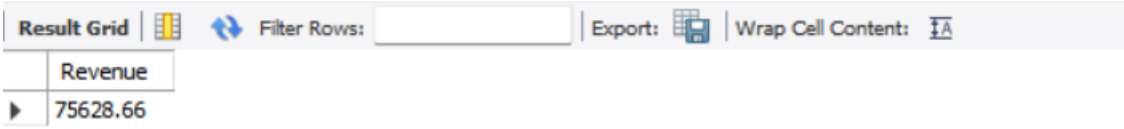


Book_ID	Title	Author	Genre	Published_Year	Price	Stock
378	Future-proofed heuristic function	Samantha Mcclain	Romance	1903	6.01	0

Calculate the total revenue generated from all orders

SELECT SUM(total_amount) AS Revenue

FROM Orders;



Revenue
75628.66

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;

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

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

Find the average price of books in the "Fantasy" genre

```
SELECT AVG(price) AS Average_Prize
FROM Books
WHERE Genre  = 'Fantasy';
```

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

Average_Prize
25.981690140845064

List customers who have placed at least 2 orders

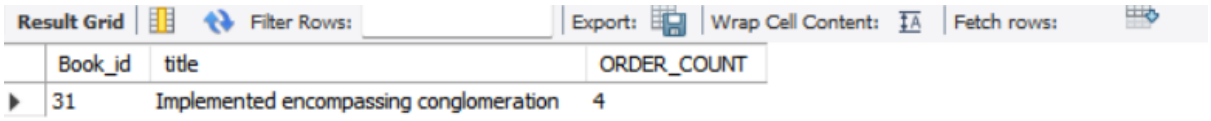
```
SELECT o.customer_id, c.name, COUNT(o.Order_id) AS ORDER_COUNT
FROM orders o
JOIN customers c ON o.customer_id=c.customer_id
GROUP BY o.customer_id, c.name
HAVING COUNT(Order_id) >=2;
```

Result Grid | Filter Rows: | 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

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;
```

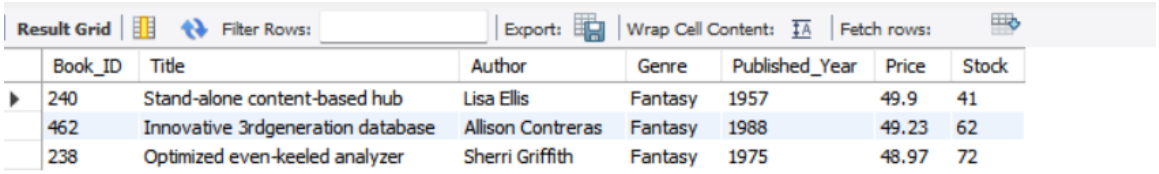


Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Book_id	title	ORDER_COUNT
31	Implemented encompassing conglomeration	4

Show the top 3 most expensive books of 'Fantasy' Genre

```
SELECT * FROM books
WHERE genre ='Fantasy'
ORDER BY price DESC LIMIT 3;
```

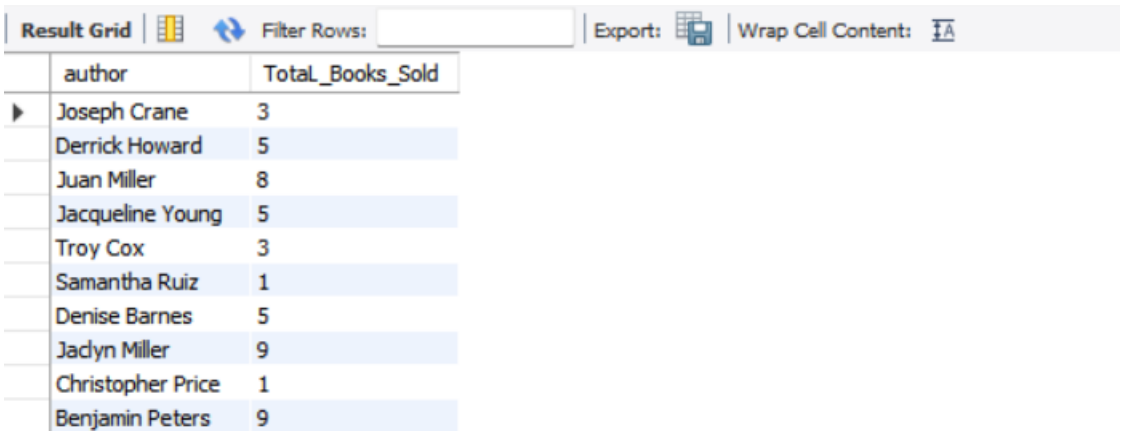


Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
240	Stand-alone content-based hub	Lisa Ellis	Fantasy	1957	49.9	41
462	Innovative 3rdgeneration database	Allison Contreras	Fantasy	1988	49.23	62
238	Optimized even-keeled analyzer	Sherri Griffith	Fantasy	1975	48.97	72

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 o.book_id=b.book_id
GROUP BY b.Author;
```



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

author	TotaL_Books_Sold
Joseph Crane	3
Derrick Howard	5
Juan Miller	8
Jacqueline Young	5
Troy Cox	3
Samantha Ruiz	1
Denise Barnes	5
Jaclyn Miller	9
Christopher Price	1
Benjamin Peters	9

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

```
SELECT DISTINCT c.city, total_amount

FROM orders o

JOIN customers c ON o.customer_id=c.customer_id

WHERE o.total_amount > 30;
```

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

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;
```

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

customer_id	name	Total_Spent
457	Kim Turner	1398.90