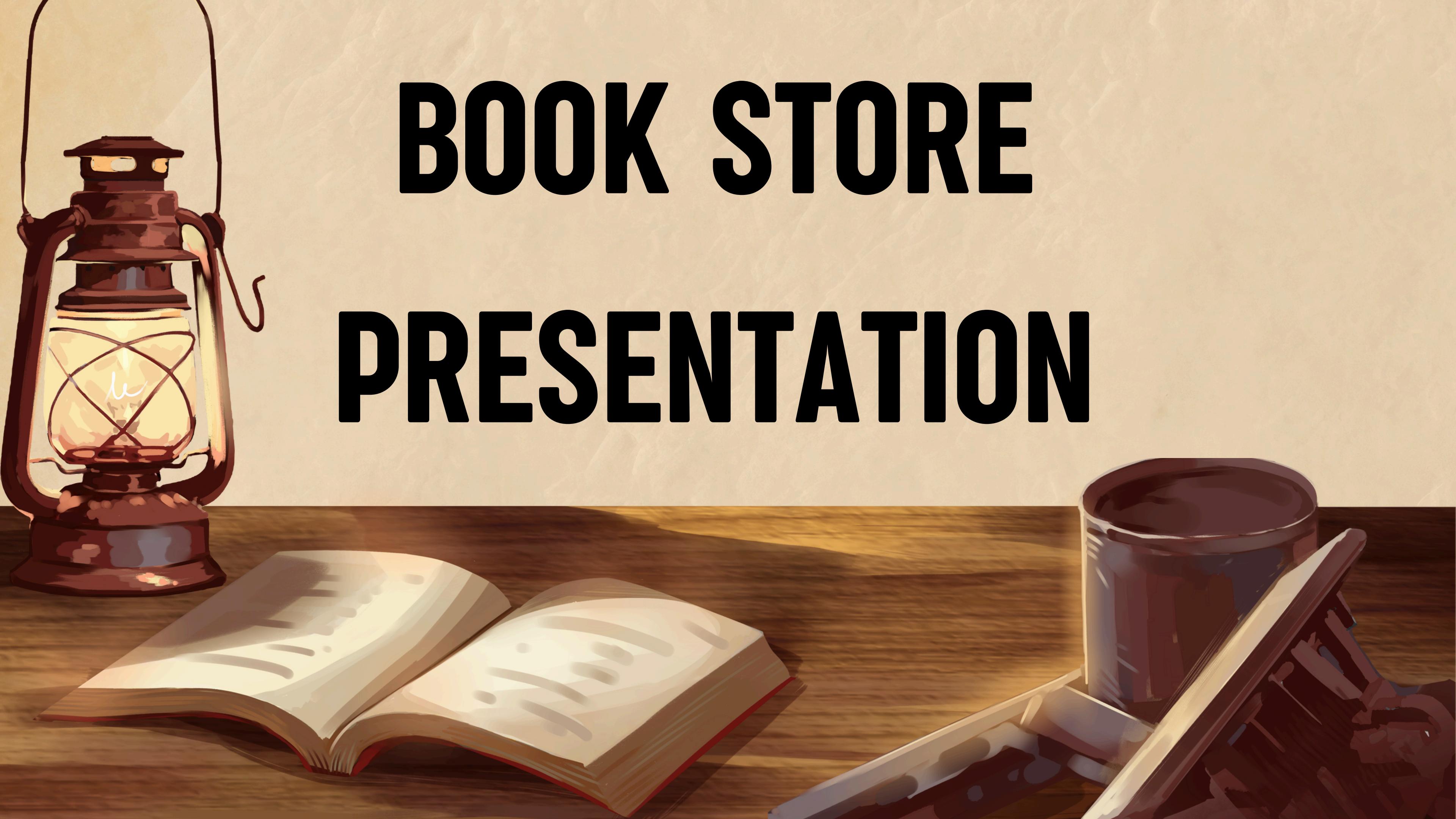


BOOK STORE PRESENTATION

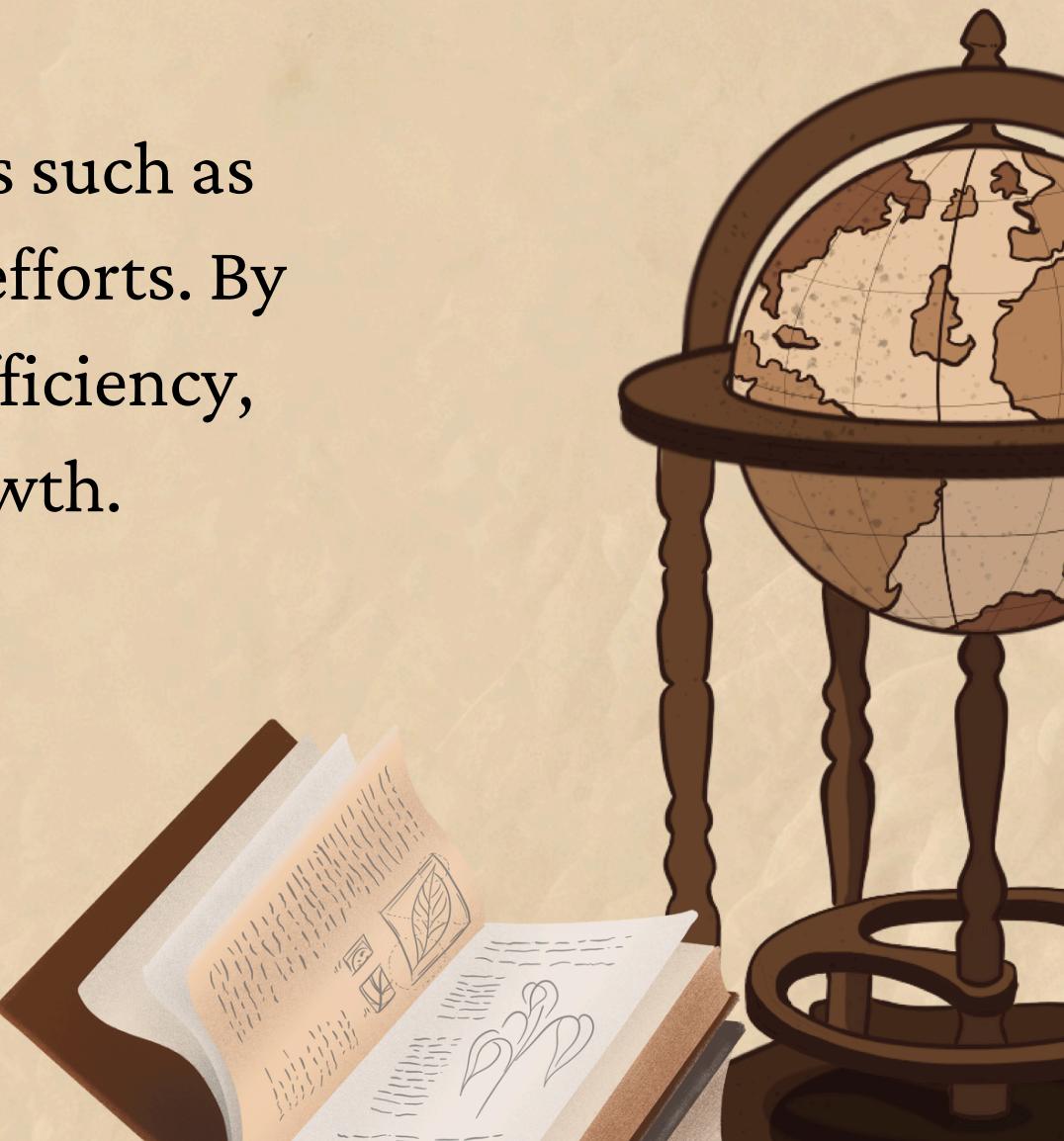




ABOUT THE BOOK STORE

This SQL project focuses on analyzing transactional and inventory data from an online book store to generate actionable insights into sales performance, customer behavior, and inventory management. Using structured queries, key metrics were extracted from the database, which includes tables for books, authors, customers, orders, order items, and categories.

Overall, the analysis provides a foundation for strategic decisions such as targeted marketing, inventory planning, and customer retention efforts. By leveraging data, the online bookstore can enhance operational efficiency, improve the shopping experience, and drive sustainable growth.





QUESTIONS

BASIC

1. Retrieve all the books from fiction genre.
2. Find books published after the year 2000.
3. List all customers from the Canada.
4. Show orders placed in november 2023.
5. Retrieve the total stock of books available.
6. Find the details of the most expensive book.
7. Show all customers who ordered more than one book.
8. Retrieve all orders where total ampoint exceeds \$350.
9. List all genre avilable in the book table.
10. Find the book wuith the lowest stock.
11. Calculate the total revenue generated from all orders.





QUESTIONS

INTERMEDIATE

1. Retrieve the total numbers of books sold for each genre.
2. Find the average price of book in fiction genere.
3. List customers who have at least 2 orders.
4. Find the most frequently ordered book.
5. Select the top 3 most expensive book of fantasy genre.
6. Retrieve the total quantity of books sold by each author.
7. List the cities where customer who spent over \$350 are located.
8. Find the customers who spend most on the orders.



```
1 -- Retrieve all the books from fiction genre
2 • SELECT *
3 *
4 FROM book
5 WHERE
6   Genre = 'Fiction';
7
```

The screenshot shows a database result grid with 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
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

```
9 -- Find books published after the year 2000
10 • SELECT *
11 *
12 FROM book
13 WHERE
14   Published_Year > 2000;
15
```

The screenshot shows a database result grid with the same columns as the first table: Book_ID, Title, Author, Genre, Published_Year, Price, and Stock. The data is as follows:

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
8	Persistent local encoding	Troy Cox	Science Fiction	2019	48.99	84
10	Ergonomic national hub	Samantha Ruiz	Mystery	2015	24.63	25
18	Adaptive 4thgeneration concept	Hector Palmer	Non-Fiction	2021	39.47	32

```
17 -- List all customers from Canada
18 • SELECT
19 *
20 FROM
21     customer
22 WHERE
23     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
*	NULL	NULL	NULL	NULL	NULL	NULL

```
25 -- Show orders placed in november 2023
26 • SELECT
27 *
28 FROM
29     orders
30 WHERE
31     Order_Date BETWEEN '2023-11-01' AND '2023-11-30';
```

	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

```
33 -- Retrieve the total stock of books available  
34 • SELECT  
35     SUM(Stock) AS total_stock  
36 FROM  
37 book;  
38
```

A screenshot of a MySQL Workbench result grid. The grid has one row with the column name 'total_stock' and a value of 25056. The grid includes standard database navigation buttons (back, forward, first, last) and export/import options.

total_stock
25056

```
39 -- Find the details of the most expensive book  
40 • SELECT  
41     *  
42 FROM  
43 book  
44 ORDER BY price DESC  
45 LIMIT 1;
```

A screenshot of a MySQL Workbench result grid. It shows a single row of data from a table with columns: Book_ID, Title, Author, Genre, Published_Year, Price, and Stock. The data is as follows: Book_ID 340, Title 'Proactive system-worthy orchestration', Author 'Robert Scott', Genre 'Mystery', Published_Year 1907, Price 49.98, and Stock 88. The grid includes standard database navigation buttons and export/import options.

	Book_ID	Title	Author	Genre	Published_Year	Price	Stock
1	340	Proactive system-worthy orchestration	Robert Scott	Mystery	1907	49.98	88

```
47 -- Show all customers who ordered more than one book
48 • SELECT
49 *
50 FROM
51     orders
52 WHERE
53     Quantity > 1;
```

A screenshot of a database query results grid. The grid has a header row with columns: Order_ID, Customer_ID, Book_ID, Order_Date, Quantity, and Total_Amount. Below the header are five data rows, each representing an order. Row 1: Order_ID 1, Customer_ID 84, Book_ID 169, Order_Date 2023-05-26, Quantity 8, Total_Amount 188.56. Row 2: Order_ID 2, Customer_ID 137, Book_ID 301, Order_Date 2023-01-23, Quantity 10, Total_Amount 216.60. Row 3: Order_ID 3, Customer_ID 216, Book_ID 261, Order_Date 2024-05-27, Quantity 6, Total_Amount 85.50. Row 4: Order_ID 4, Customer_ID 433, Book_ID 343, Order_Date 2023-11-25, Quantity 7, Total_Amount 301.21. Row 5: Order_ID 5, Customer_ID 14, Book_ID 431, Order_Date 2023-07-26, Quantity 7, Total_Amount 136.36.

	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

```
55 -- Retrieve all orders where total amount exceeds $350
56 • SELECT
57 *
58 FROM
59     orders
60 WHERE
61     Total_Amount > 350;
```

A screenshot of a database query results grid. The grid has a header row with columns: Order_ID, Customer_ID, Book_ID, Order_Date, Quantity, and Total_Amount. Below the header are four data rows. Row 1: Order_ID 9, Customer_ID 109, Book_ID 407, Order_Date 2024-01-04, Quantity 9, Total_Amount 379.71. Row 2: Order_ID 36, Customer_ID 417, Book_ID 260, Order_Date 2024-01-21, Quantity 9, Total_Amount 446.31. Row 3: Order_ID 39, Customer_ID 488, Book_ID 422, Order_Date 2024-08-03, Quantity 8, Total_Amount 367.28. Row 4: Order_ID 41, Customer_ID 418, Book_ID 147, Order_Date 2024-05-05, Quantity 8, Total_Amount 361.60.

	Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
▶	9	109	407	2024-01-04	9	379.71
	36	417	260	2024-01-21	9	446.31
	39	488	422	2024-08-03	8	367.28
	41	418	147	2024-05-05	8	361.60

63 -- List all genre available in the book table

64 • SELECT DISTINCT

65 genre

66 FROM

67 book;

4

genre
► Biography
Fantasy
Non-Fiction
Fiction
Romance
Science Fiction
Mystery

69 -- Find the book with the lowest stock

70 • SELECT

71 *

72 FROM

73 book

74 ORDER BY stock

75 LIMIT 1

```
1 -- Retrieve the total numbers of books sold for each genre
2 • SELECT
3     book.Genre, SUM(quantity) AS total_qty
4 FROM
5     orders
6     JOIN
7         book ON orders.Book_ID = book.Book_ID
8 GROUP BY book.genre;
```

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

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

```
10 -- Find the average price of book in fiction gener
11 • SELECT
12     AVG(price) AS avg_price
13 FROM
14     book
15 WHERE
16     genre = 'Fiction';
```

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

avg_price
28.381333

```
34 -- Select the top 3 most expensive book of fantasy genre
35 • SELECT
36 *
37 FROM
38 book
39 WHERE
40     genre = 'Fantasy'
41 ORDER BY price DESC
42 LIMIT 3;
43
```

A screenshot of a MySQL Workbench result grid. The grid has columns for Book_ID, Title, Author, Genre, Published_Year, Price, and Stock. The data shows three books: Stand-alone content-based hub by Lisa Ellis (Genre: Fantasy), Innovative 3rdgeneration database by Allison Contreras (Genre: Fantasy), and Optimized even-keeled analyzer by Sherri Griffith (Genre: Fantasy). The prices range from 48.97 to 49.90.

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

```
44 -- Retrieve the total quantity of books sold by each author
45 • SELECT
46     author, SUM(quantity) AS total_books_sold
47 FROM
48     orders
49     JOIN
50     book ON orders.Book_ID = book.Book_ID
51 GROUP BY Author;
```

A screenshot of a MySQL Workbench result grid. The grid has columns for author and total_books_sold. It shows five authors with their respective total sales counts: Margaret Moore (8), John Davidson (13), Christopher Fuentes (6), Marissa Smith (16), Christopher Dixon (15), and Tonya Saunders (21).

	author	total_books_sold
▶	Margaret Moore	8
	John Davidson	13
	Christopher Fuentes	6
	Marissa Smith	16
	Christopher Dixon	15
	Tonya Saunders	21

```

53 -- List the cities where customer who spent over $350 are located
54 • SELECT
55 *
56 FROM
57 orders
58 JOIN
59 customer ON customer.Customer_ID = orders.Customer_ID
60 WHERE
61 Total_Amount > 350;
62

```

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

	Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount	Customer_ID	Name	Email	Phone	City	Country
▶	9	109	407	2024-01-04	9	379.71	109	Jacob Kelley	qbarber@shelton.org	1234567999	Ravenberg	Macao
	36	417	260	2024-01-21	9	446.31	417	Melissa Frey	floresricardo@smith.org	1234568307	Erikaberg	Bahamas
	39	488	422	2024-08-03	8	367.28	488	Gerald Miranda	gregsmith@hotmail.com	1234568378	North Brian	Heard Island and McDonald Islands
	41	418	147	2024-05-05	8	361.60	418	Kiara Blankenship MD	grahamvictor@sutton.com	1234568308	Brandimouth	Belarus
	60	404	49	2023-04-26	9	445.50	404	Lisa Duran	christina28@espinoza.com	1234568294	Lake Tyler	Korea

```
63  -- Find the customers who spend most on the orders
64 • SELECT
65      customer.customer_id, name, SUM(total_amount) AS total_spent
66 FROM
67     orders
68     JOIN
69     customer ON customer.Customer_ID = orders.Customer_ID
70 GROUP BY customer.customer_id , customer.name
71 ORDER BY total_spent DESC
72 LIMIT 10;
```

< [REDACTED]

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

	customer_id	name	total_spent
▶	457	Kim Turner	1398.90
	174	Jonathon Strickland	1080.95
	364	Carrie Perez	1052.27
	405	Julie Smith	991.00
	386	Pamela Gordon	986.30
	425	Ashley Perez	942.62
	474	Anthony Young	929.19
	163	Robert Clark	746.65
	167	Justin Spencer	719.93
	214	Alexander Scott	682.15

```
18 -- List customers who have at least 2 orders
19 • SELECT
20     customer_id, COUNT(order_id) AS total_orders
21 FROM
22     orders
23 GROUP BY Customer_ID
24 HAVING COUNT(Order_ID) >= 2;
```

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

customer_id	total_orders
84	2
137	2
216	2
14	2
195	3
109	2

```
26 -- Find the most frequently ordered book
27 • SELECT
28     book_id, COUNT(order_id) AS total_order
29 FROM
30     orders
31 GROUP BY Book_ID
32 ORDER BY total_order DESC;
```

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

	book_id	total_order
▶	88	4
	491	4
	333	4
	31	4

Thank You

