

ONLINE BOOK STORE - SQL ANALYSIS PROJECT

chandan kumar mahaseth

Online Book Store - SQL Analysis Project



This project focuses on building a structured SQL database for an online bookstore and performing a series of business-driven queries to uncover customer behavior, sales performance, inventory status, and genre trends.



★ KEY HIGHLIGHTS: ★

Created normalized tables: Customers, Orders, and Books

Wrote and optimized 15+ real-world SQL queries

Answered key business questions using GROUP BY, HAVING, JOINS, subqueries, CTEs, and aggregations

Categorized queries into Basic, Intermediate, and Advanced levels for clarity

Simulated real-world use cases like top-spending customers, genre revenue, and stock tracking

TOOLS USED:

MYSQL / SQL SERVER, STRUCTURED QUERY LANGUAGE (SQL)



SKILLS APPLIED:

DATABASE DESIGN, DATA AGGREGATION, DATA CLEANING,
BUSINESS INTELLIGENCE, ANALYTICAL THINKING



Project Objective & Business Goals



THE OBJECTIVE OF THIS PROJECT IS TO ANALYZE THE OPERATIONAL AND SALES PERFORMANCE OF A FICTIONAL ONLINE BOOKSTORE BY LEVERAGING SQL QUERIES ON A RELATIONAL DATABASE.

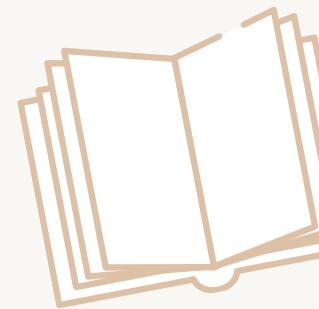
★ BUSINESS GOALS: ★

- Understand customer purchasing behavior (e.g., order frequency, quantity, diversity)
- Identify best-selling books and authors based on total revenue and order volume
- Track genre-wise performance to inform future stocking decisions
- Detect high-value customers and develop potential marketing strategies

OUTCOME:

The project provides a 360° view of bookstore operations using SQL as a data analysis tool, making it ideal for showcasing data querying skills in real-world business scenarios.

RETRIEVE ALL BOOKS IN THE 'FICTION' GENRE.



```
1  SELECT
2    *
3  FROM
4    books
5 WHERE
6    genre = 'fiction';
```

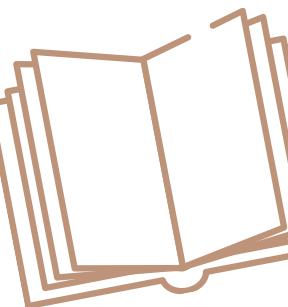
Result Grid | Filter Rows: Export: Wrap Cell Content:

	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	Advanced didactic interface	Natalia Gonzalez	Fiction	1922	25.87	84	

OBJECTIVE: Identify books under the Fiction category.

DISPLAY CUSTOMERS WHO ORDERED MORE THAN ONE COPY OF ANY BOOK.

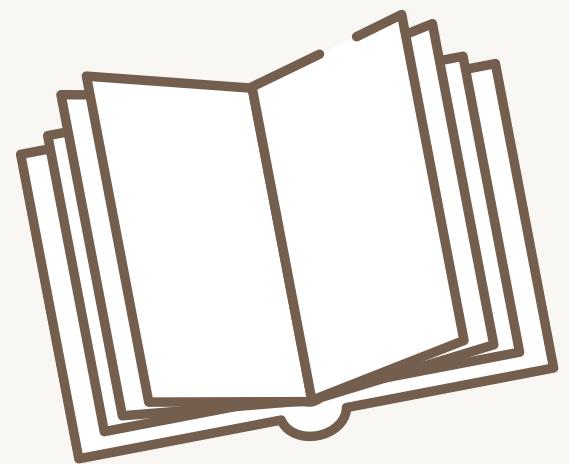
```
SELECT DISTINCT c.name  
FROM customers c  
JOIN orders o ON c.customer_id = o.customer_id  
WHERE o.quantity > 1;
```



Result Grid | Filter Rows:

	name
▶	Crystal Clements
▶	Stephen Vasquez
▶	Matthew Johnson
▶	Ronald Osborn
▶	Thomas Garcia
▶	Jennifer Murray

OBJECTIVE: Identify bulk buyers.



CALCULATE THE TOTAL NUMBER OF BOOKS ORDERED BY EACH CUSTOMER



| ⚡ | 📄 | 🔎 | 🖐️ | ⏷ | ✅ | ✗ | 📱 | Limit to 1000 rows | ⚡ | 🌟+ | 📈 | 🔎 | 📊 | +

SELECT

```
c.customer_id,  
c.name,  
SUM(o.quantity) AS total_books_ordered  
FROM customers c  
JOIN orders o ON c.customer_id = o.customer_id  
GROUP BY c.customer_id, c.name;
```

OBJECTIVE: To measure individual customer engagement by total books ordered

Result Grid | Filter Rows:

	customer_id	name	total_books_ordered
▶	2	Crystal Clements	10
	6	Stephen Vasquez	4
	7	Susan Hicks	1
	8	Matthew Johnson	8
	10	Ronald Osborn	6
	11	Thomas Garcia	5
	12	Jennifer Murray	3
	13	Kristine Kim	14

LIST CUSTOMERS WHO ORDERED MORE THAN 3 DISTINCT BOOKS



```
1 • SELECT
2     c.customer_id,
3     c.name,
4     COUNT(DISTINCT o.book_id) AS distinct_books_ordered
5 FROM customers c
6 JOIN orders o ON c.customer_id = o.customer_id
7 GROUP BY c.customer_id, c.name
8 HAVING COUNT(DISTINCT o.book_id) > 3;
```

Result Grid | Filter Rows: | Export:

	customer_id	name	distinct_books_ordered
▶	107	Amy Hunt	4
	174	Jonathon Strickland	4
	325	Emily Vargas	4
	364	Carrie Perez	6
	405	Julie Smith	4
	425	Ashley Perez	4
	437	Cynthia Cooper	4
	457	Kim Turner	4

OBJECTIVE: To find customers with diverse reading interests or high product exposure.

FIND THE AVERAGE QUANTITY ORDERED PER GENRE

```
1 • SELECT
2     b.genre,
3     ROUND(AVG(o.quantity), 2) AS avg_quantity
4 FROM books b
5 JOIN orders o ON b.book_id = o.book_id
6 GROUP BY b.genre;
```

genre	avg_quantity
Biography	5.18
Non-Fiction	4.94
Fantasy	5.51
Romance	5.49
Science Fiction	5.32
Mystery	6.07
Fiction	4.89

OBJECTIVE: To understand customer demand intensity by genre.

SHOW THE TOTAL SALES AMOUNT PER GENRE



```
1 • SELECT
2     b.genre,
3     SUM(o.total_amount) AS genre_sales
4 FROM books b
5 JOIN orders o ON b.book_id = o.book_id
6 GROUP BY b.genre;
```

genre	genre_sales
Biography	7870.06
Non-Fiction	11603.06
Fantasy	11238.380000000003
Romance	13086.980000000005
Science Fiction	11770.510000000002
Mystery	12788.45
Fiction	7271.220000000001

OBJECTIVE: To evaluate which genres are driving the most revenue.

Display the top 5 books with the highest total sales



```
SELECT  
    b.title,  
    SUM(o.total_amount) AS total_sales  
FROM books b  
JOIN orders o ON b.book_id = o.book_id  
GROUP BY b.title  
ORDER BY total_sales DESC  
LIMIT 5;
```



	title	total_sales
▶	Integrated secondary access	1104.69
	Multi-tiered responsive parallelism	1077.12
	Switchable modular moratorium	1047.48
	Cross-platform next generation website	952.98
	Grass-roots systematic moderator	872.29

OBJECTIVE: To highlight the bookstore's top-performing products.



IDENTIFY THE AUTHOR WHOSE BOOKS GENERATED THE HIGHEST TOTAL SALES



```
1 • SELECT
2   b.author,
3   SUM(o.total_amount) AS total_revenue
4   FROM books b
5   JOIN orders o ON b.book_id = o.book_id
6   GROUP BY b.author
7   ORDER BY total_revenue DESC
8   LIMIT 1;
```

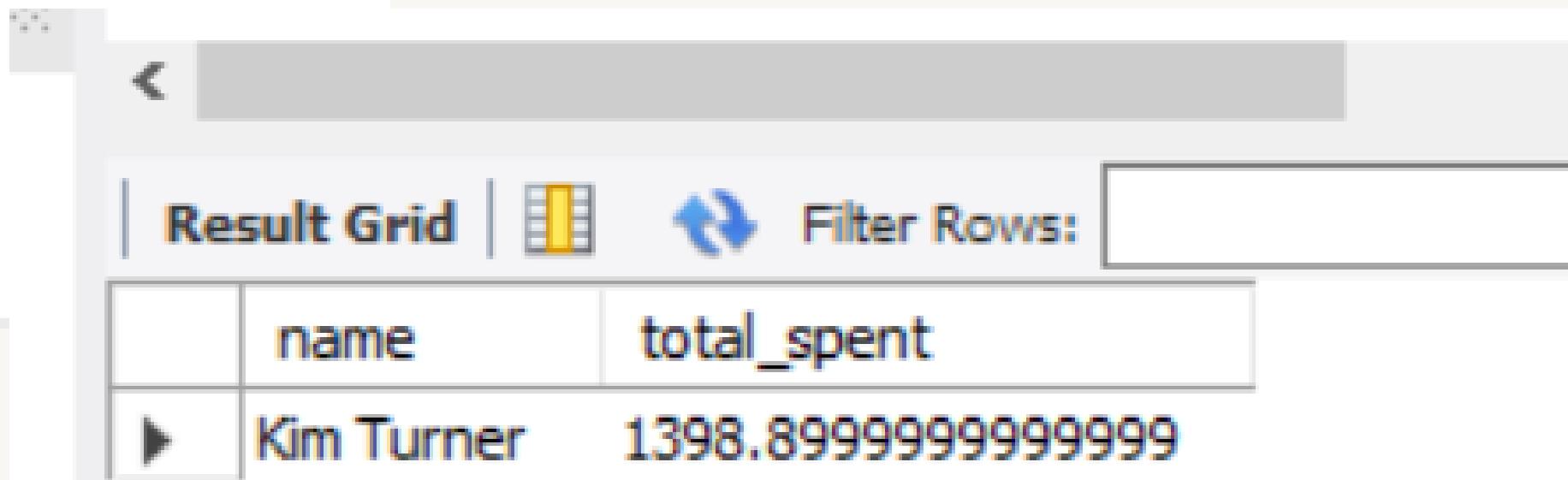
Result Grid | Filter Rows:

	author	total_revenue
▶	Sheena Harris	1104.69

OBJECTIVE: To recognize the most profitable author for future stocking decisions.

FIND THE CUSTOMER WHO SPENT THE MOST OVERALL

```
SELECT  
    c.name,  
    SUM(o.total_amount) AS total_spent  
FROM customers c  
JOIN orders o ON c.customer_id = o.customer_id  
GROUP BY c.customer_id, c.name  
ORDER BY total_spent DESC  
LIMIT 1;
```



The screenshot shows the MySQL Workbench interface with the following details:

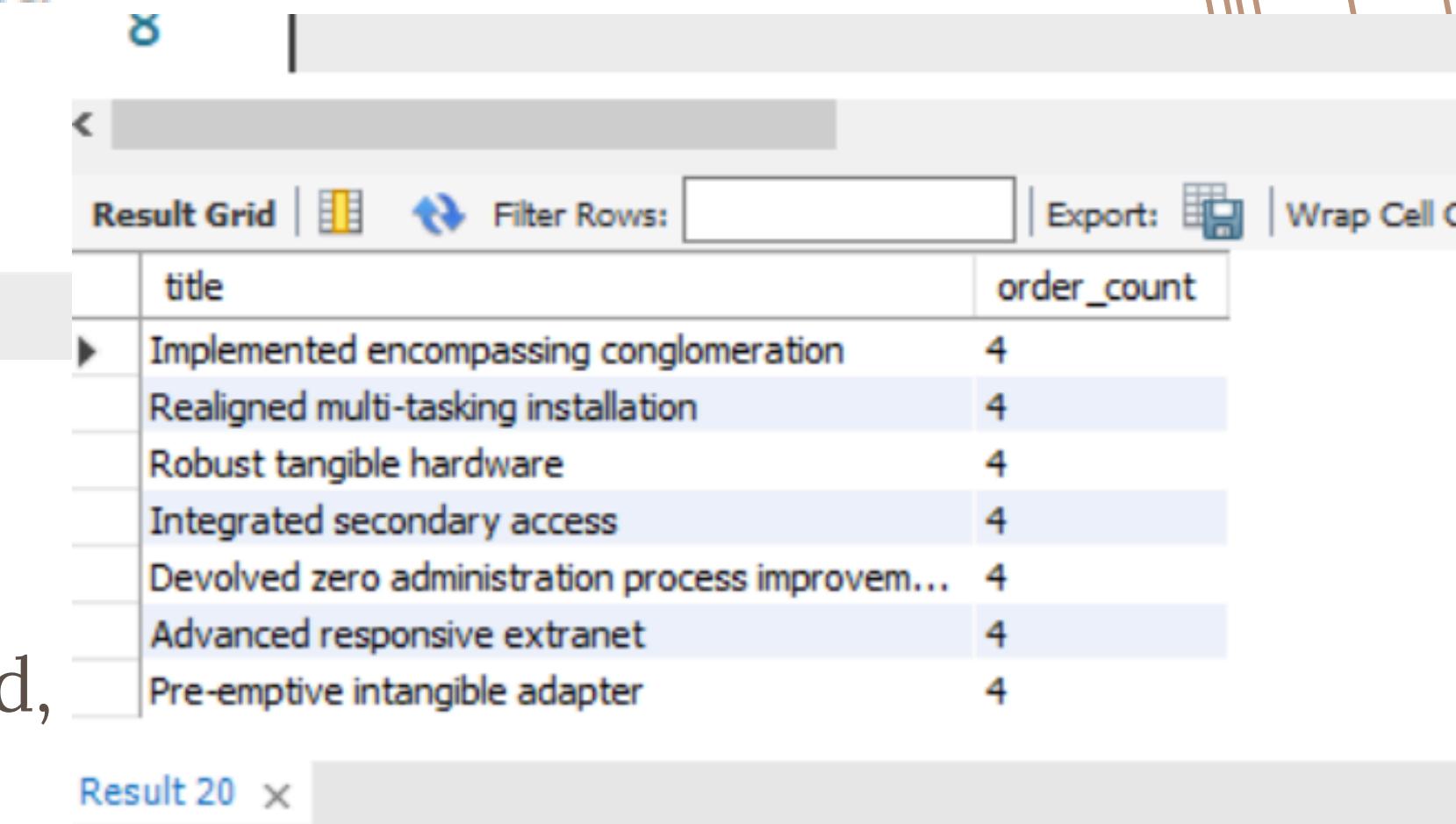
- The title bar says "MySQL Workbench - Local".
- The toolbar includes "File", "Edit", "View", "Tools", "Help", and icons for "New", "Open", "Save", "Run", "Stop", and "Close".
- The main area has tabs for "Result Grid" and "SQL Editor".
- The "Result Grid" tab is selected, showing a single row of data.
- The data grid has columns: "name" and "total_spent".
- The row contains "Kim Turner" and "1398.8999999999999".
- Below the grid are navigation buttons: "First", "Previous", "Next", "Last", and "Rows: 1".
- On the right, there is a "Filter Rows:" input field.

OBJECTIVE: To identify high-value customers for targeted retention strategies.

IDENTIFY BOOKS THAT WERE ORDERED MORE THAN 3 TIMES



```
1 • SELECT
2     b.title,
3     COUNT(o.order_id) AS order_count
4 FROM books b
5 JOIN orders o ON b.book_id = o.book_id
6 GROUP BY b.book_id, b.title
7 HAVING COUNT(o.order_id) > 3;
```

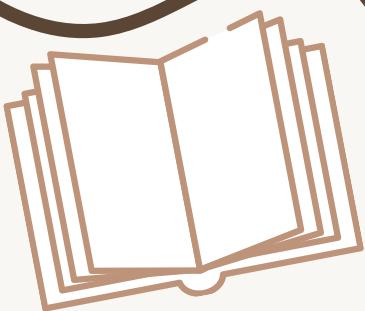


title	order_count
Implemented encompassing conglomeration	4
Realigned multi-tasking installation	4
Robust tangible hardware	4
Integrated secondary access	4
Devolved zero administration process improvem...	4
Advanced responsive extranet	4
Pre-emptive intangible adapter	4

Result 20 ×

Objective: To uncover books with high repeat demand, signaling popularity.

LIST CUSTOMERS WHO HAVE NEVER ORDERED A BOOK FROM THE 'ROMANCE' GENRE



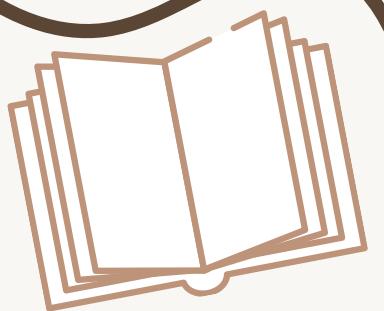
```
1 • ⏷ WITH Romance_Customers AS (
2     SELECT DISTINCT o.Customer_ID
3     FROM orders o
4     JOIN books b ON o.Book_ID = b.Book_ID
5     WHERE b.Genre = 'Romance'
6 )
7     SELECT *
8     FROM customers
9     WHERE Customer_ID NOT IN (SELECT Customer_ID FROM Romance_Customers);
```

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

	Customer_ID	Name	Email	Phone	City	Country
1	Deborah Griffith	balljoseph@wright-keith.net	1234567891	South Craigfort	Denmark	
3	Susan Fuller	beanmichael@burnett-stewart.net	1234567893	Austinbury	Equatorial Guinea	
4	Jamie Ramirez	amandahood@warren.com	1234567894	Dianemouth	Slovenia	
5	Marcus Murphy	connerjohn@yahoo.com	1234567895	Smithbury	Guinea-Bissau	
7	Susan Hicks	jeffrey91@yahoo.com	1234567897	East Rebecca	Montenegro	
8	Matthew Johnson	austinkenneth@manning.net	1234567898	Kirstenborough	Israel	

OBJECTIVE: To identify untapped audience segments for Romance genre promotions.

FIND THE BOOK WITH THE HIGHEST AVERAGE QUANTITY PER ORDER



SELECT

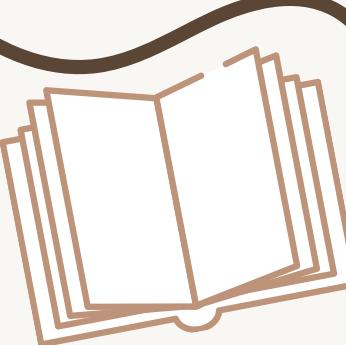
```
b.title,  
ROUND(AVG(o.quantity), 2) AS avg_quantity  
FROM books b  
JOIN orders o ON b.book_id = o.book_id  
GROUP BY b.book_id, b.title  
ORDER BY avg_quantity DESC  
LIMIT 1;
```

Result Grid | Filter Rows: Export:

	title	avg_quantity
▶	Synergistic dedicated concept	10.00

OBJECTIVE: - To find books frequently bought in bulk—ideal for promotions or bundling.

CALCULATE REMAINING STOCK AFTER FULFILLING ALL ORDERS



```
book_store  books  customers  orders  SQL File 6*  SQL File 7* ×  
| Limit to 1000 rows |  
1 • SELECT  
2     b.book_id,  
3     b.title,  
4     b.stock,  
5     COALESCE(SUM(o.quantity), 0) AS total_ordered,  
6     b.stock - COALESCE(SUM(o.quantity), 0) AS remaining_stock  
7   FROM books b  
8   LEFT JOIN orders o ON b.book_id = o.book_id  
9   GROUP BY b.book_id, b.title, b.stock;  
10 |
```

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

	book_id	title	stock	total_ordered	remaining_sto
▶	1	Configurable modular throughput	100	3	97
	2	Persevering reciprocal knowledge user	19	0	19
	3	Streamlined coherent initiative	27	5	22
	4	Customizable 24hour product	8	0	8
	5	Adaptive 5thgeneration encoding	16	8	8
	6	Advanced encompassing implementation	2	0	2
	7	Open-architected exuding structure	95	5	90
	8	Persistent local encoding	84	3	81

Result 31 ×

OBJECTIVE: To assess real-time inventory status after customer fulfillment.

THANK YOU FOR STAY

Linkdin :- <https://www.linkedin.com/in/mahaseth>

GitHub :- <https://github.com/imcmahaseth>

GMAIL:- iammahasethchandan@gmail.com

