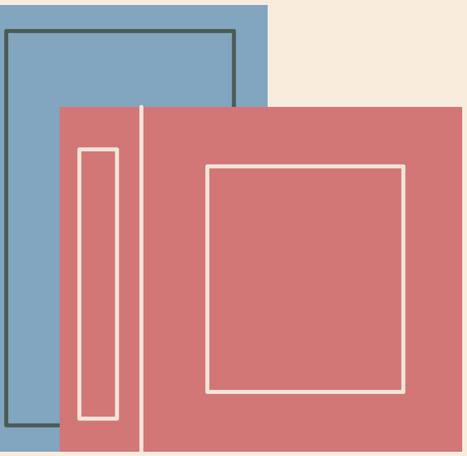
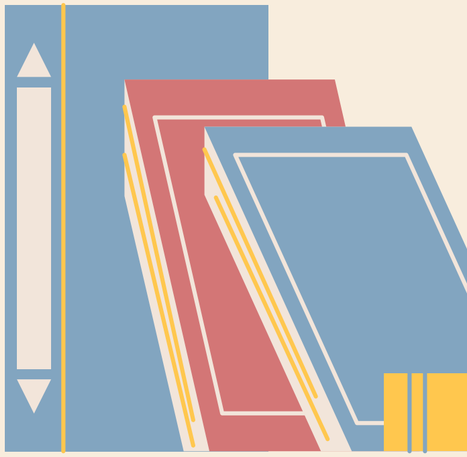
WELCOME TO OUR

ONLINE BOOK STORE

Analysis By SQL



OBJECTIVE

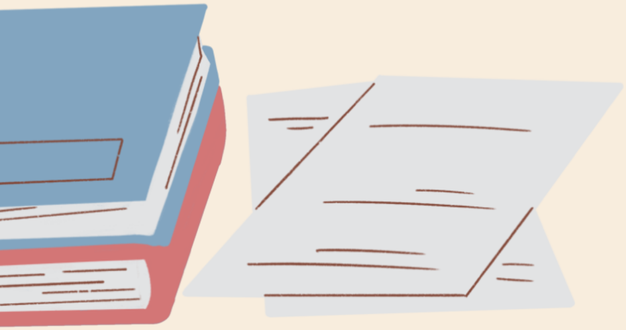


Purpose: To analyze an Online Book Store dataset using SQL to extract key insights on sales, customers, and products.

Why It Matters: Helps in understanding business performance, improving sales strategy, and making data-driven decisions.

Data Source: Simulated relational database with tables: Customers, Books, Orders

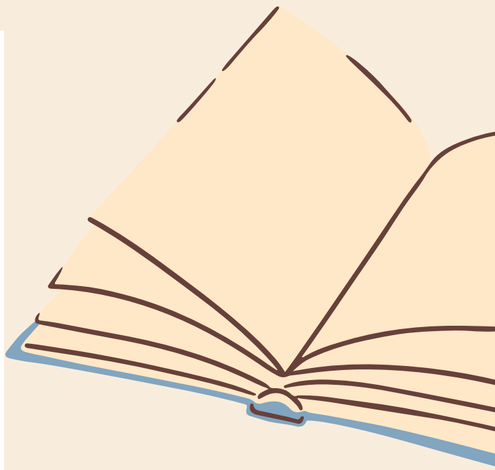
Tools Used: SQL , MySQL (for database querying)



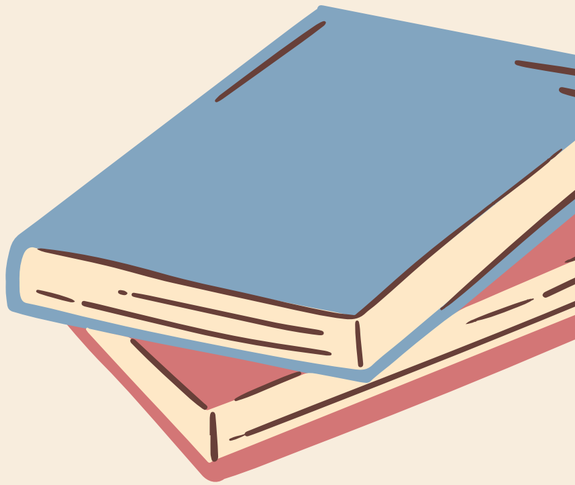
DATASETS

- books
  - Columns
    - Book_ID
    - Title
    - Author
    - Genre
    - Published_Year
    - Price
    - Stock

- customers
  - Columns
    - Customer_ID
    - Name
    - Email
    - Phone
    - City
    - Country



- orders
  - Columns
    - Order_ID
    - Customer_ID
    - Book_ID
    - Order_Date
    - Quantity
    - Total_Amount



1.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

2.Find books published after the year 1950

```
SELECT * FROM books
WHERE Published_Year > 1950;
```

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

3.List all customers from the Canada

```
select * from customers
where 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

4.Show orders placed in November 2023

```
SELECT * FROM orders
WHERE 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

5.Find the details of the most expensive book

```
SELECT * FROM books
WHERE Price = (SELECT MAX(Price)
FROM books);
```

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

6.Retrieve the total stock of books available

```
SELECT SUM(Stock) AS Total_Stock
FROM books;
```

Total_Stock
25056

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

```
SELECT * FROM orders
WHERE Quantity>1;
```

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	216.6
3	216	261	2024-05-27	6	85.5
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	249.4

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

```
SELECT * FROM orders
WHERE Total_Amount>20;
```

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	216.6
3	216	261	2024-05-27	6	85.5
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	249.4

9.List all genres available in the books table

```
SELECT DISTINCT Genre
FROM books;
```

Genre
Biography
Fantasy

Non-Fiction
Fiction
Romance

Science Fiction
Mystery

10.Find the book with the lowest stock

```
SELECT * FROM books
ORDER BY Stock ASC
LIMIT 1;
```

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

11.Calculate the total revenue generated from all orders

```
SELECT
    Round(SUM(Total_Amount),2) AS Total_Revenue
FROM orders;
```

Total_Revenue
75628.66

12.Retrieve the total number of books sold for each genre

```
SELECT
    books.Genre, SUM(orders.Quantity) AS Total_book_sold
FROM
    books
        JOIN
    orders ON books.Book_ID = orders.Book_ID
GROUP BY books.Genre;
```

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

13.Find the average price of books in the fantasy genre

```sql
SELECT
    AVG(price) AS Average_Price
FROM
    books
WHERE
    Genre = 'Fantasy';
```

Average_Price
25.981690140845064

14.List customers who have placed at least 2 orders

```sql
SELECT
    orders.Customer_ID,
    customers.Name,
    COUNT(orders.Order_ID) AS Order_Count
FROM
    orders
        JOIN
    customers ON orders.Customer_ID = customers.Customer_ID
GROUP BY orders.Customer_ID , customers.Name
HAVING COUNT(orders.Order_ID) >= 2;
```

Customer_ID	Name	Order_Count
2	Crystal Clements	2
6	Stephen Vasquez	2
8	Matthew Johnson	2
13	Kristine Kim	2
14	John Wood	2
15	Vanessa Gaines	2
16	Stacey Flores	3
21	Edgar Frost	2

15.find the most Frequently ordered book

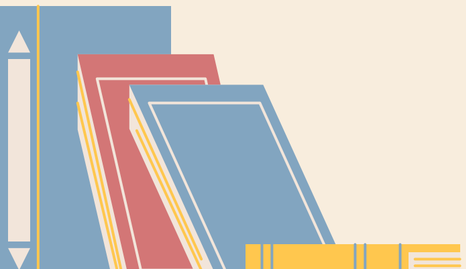
```sql
SELECT
    o.Book_ID, b.Title, COUNT(o.Order_ID) AS Order_count
FROM
    orders AS o
        JOIN
    books AS b ON o.Book_ID = b.Book_ID
GROUP BY o.Book_ID , b.Title
ORDER BY Order_count DESC
LIMIT 1;
```

Book_ID	Title	Order_count
31	Implemented encompassing conglomeration	4

16.Show the top 3 most expensive books of "Fantasy" Genre

```sql
SELECT * FROM books
WHERE Genre = 'Fantasy'
ORDER BY Price DESC
LIMIT 3;
```

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



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

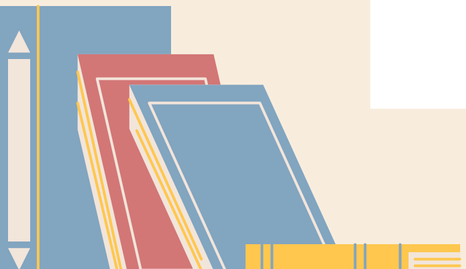
```sql
SELECT
    books.Author, SUM(orders.Quantity) AS Total_book_sold
FROM
    books
        JOIN
    orders ON books.Book_ID = orders.Book_ID
GROUP BY books.Author
ORDER BY Total_book_sold DESC;
```

Author	Total_book_sold
Patrick Contreras	28
Melissa Taylor	27
Thomas Trujillo	24
Emily James	24
Sheena Harris	23
Ellen Doyle	23
Erica Parker	23
Valerie Moore	23

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

```sql
SELECT DISTINCT
    (customers.City), orders.Total_Amount
FROM
    customers
        JOIN
    orders ON customers.Customer_ID = orders.Customer_ID
WHERE
    orders.Total_Amount > 30;
```

City	Total_Amount
East Derekberg	298.06
Hamiltonstad	148.02
Kirstenborough	95.85
Kirstenborough	44.61
Lake Benjamin	192.12
West Monicabury	221.8
South Ashleychester	39.51
Lake Robert	48.8



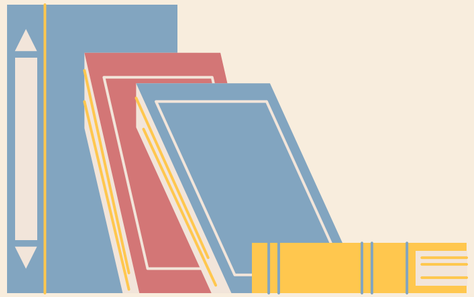
20 Calculate the stock remaining after fulfilling all orders

```sql
SELECT
    books.Book_ID,
    books.Title,
    books.Stock,
    COALESCE(SUM(orders.Quantity), 0) AS Order_Quantity,
    books.Stock - COALESCE(SUM(orders.Quantity), 0) AS Remaining_Quantity
FROM
    books
        LEFT JOIN
    orders ON books.Book_ID = orders.Book_ID
GROUP BY books.Book_ID , books.Title,
    books.Stock;
```

Book_ID	Title	Stock	Order_Quantity	Remaining_Quantity
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

Book_ID	Title	Stock	Order_Quantity	Remaining_Quantity
9	Optimized interactive challenge	70	0	70
10	Ergonomic national hub	25	1	24
11	Secured zero tolerance time-frame	10	5	5
12	Polarized optimal array	63	0	63
13	Adaptive 5thgeneration orchestration	99	9	90
14	Re-engineered demand-driven parallelism	95	0	95
15	User-friendly motivating strategy	58	0	58
16	Vision-oriented tangible project	8	1	7

19.Find the customer who spent the most on orders



```sql
SELECT
    customers.Customer_ID,
    customers.Name,
    Round(SUM(orders.Total_Amount),2) AS Total_Spent
FROM
    customers
        JOIN
    orders ON customers.Customer_ID = orders.Customer_ID
GROUP BY customers.Customer_ID , customers.Name
ORDER BY Total_Spent DESC limit 1;
```

	Customer_ID	Name	Total_Spent
▶	457	Kim Turner	1398.9

FOR YOUR ATTENTION

THANK YOU

Presentation by
Anjali Keshri