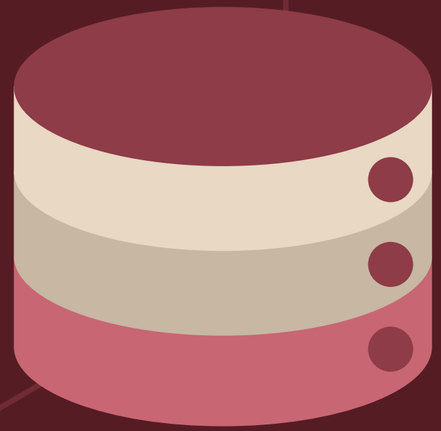
📚 SQL PROJECT ON BOOK SALES ANALYSIS

Exploring Advanced SQL Queries for Data Insights



INTRODUCTION

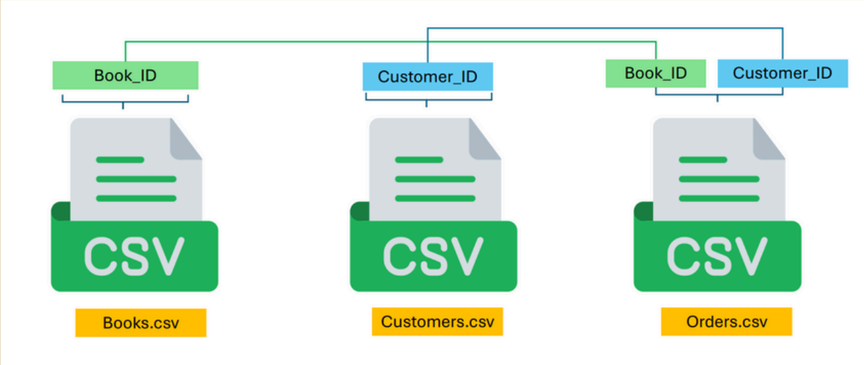
UNDERSTANDING THE DATASET 📊

1 BOOKS TABLE 📚

- BOOK_ID – UNIQUE IDENTIFIER FOR EACH BOOK
- TITLE – NAME OF THE BOOK
- AUTHOR – AUTHOR OF THE BOOK
- GENRE – CATEGORY/GENRE OF THE BOOK
- PUBLISH_YEAR – YEAR THE BOOK WAS PUBLISHED
- PRICE – COST OF THE BOOK
- STOCK – AVAILABLE QUANTITY IN INVENTORY

2 CUSTOMERS TABLE 👤

- CUSTOMER_ID – UNIQUE IDENTIFIER FOR EACH CUSTOMER
- NAME – CUSTOMER'S FULL NAME
- EMAIL – CONTACT EMAIL
- PHONE – CUSTOMER'S PHONE NUMBER
- CITY – CITY WHERE THE CUSTOMER RESIDES
- COUNTRY – COUNTRY OF THE CUSTOMER



3 ORDERS TABLE 🛒

- ORDER_ID – UNIQUE ID FOR EACH ORDER
- CUSTOMER_ID – ID LINKING TO THE CUSTOMER WHO PLACED THE ORDER
- BOOK_ID – ID OF THE PURCHASED BOOK
- ORDER_DATE – DATE OF PURCHASE
- QUANTITY – NUMBER OF COPIES BOUGHT
- TOTAL_AMOUNT – TOTAL COST OF THE ORDER

KEY PROBLEM STATEMENTS

01 Total Books Sold Per Genre 📖

02 Average Price of Fantasy Books 🏰

03 Frequent Customers 👥

04 Most Ordered Book 📊

05 Top 3 Expensive Fantasy Books 💰

06 Books Sold Per Author ✍️

07 Customer Locations Based on Spending 🌍

08 Highest Spending Customer 🏆

09 Remaining Stock Calculation 📦

QUERY 1 – IDENTIFY THE TOTAL NUMBER OF BOOKS SOLD FOR EACH GENRE.

```sql
SELECT
    book.Genre, SUM(Quantity) AS total_book_sold
FROM
    online_book_store.book
        JOIN
    orderss ON book.Book_ID = orderss.Book_ID
GROUP BY book.genre

```

Result Grid | Filter Rows:

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

Result 8

QUERY 2 - CALCULATE THE AVERAGE PRICE OF BOOKS IN THE "FANTASY" GENRE.

```sql
SELECT
    ROUND(AVG(price), 2) AS avg_price
FROM
    online_book_store.book
WHERE
    genre = 'fantasy';
```

Result Grid

	avg_price
▶	25.98

QUERY 3 - LIST CUSTOMERS WHO HAVE PLACED AT LEAST 2 ORDERS.

```
1  SELECT
2      customers.name,
3      customers.Customer_ID,
4      COUNT(orderss.Order_ID) AS order_count
5  FROM
6      online_book_store.orderss
7          JOIN
8      customers ON orderss.Customer_ID = customers.Customer_ID
9  GROUP BY customers.Customer_ID , customers.name
10 HAVING order_count >= 2;
```

Result Grid | Filter Rows: | Export: | Wrap

name	Customer_ID	order_count
Gary Blair	84	2
Steven Miller	137	2
Phillip Allen	216	2
John Wood	14	2
Dominique Turner	195	3
Jacob Kelley	109	2
Mr. David Cox	94	3

QUERY 4 - DETERMINE THE MOST FREQUENTLY ORDERED BOOK.

```sql
SELECT
    book.title,
    orderss.book_id,
    COUNT(orderss.order_id) AS order_count
FROM
    online_book_store.orderss
        JOIN
    book ON orderss.Book_ID = book.Book_ID
GROUP BY book.title , orderss.Book_ID
ORDER BY order_count DESC
LIMIT 1;
```

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

title	book_id	order_count
Robust tangible hardware	88	4

QUERY 5 - SHOW THE THREE MOST EXPENSIVE BOOKS IN THE "FANTASY" GENRE.

Limit to 1000 rows

```sql
SELECT
    *
FROM
    online_book_store.book
WHERE
    genre = 'fantasy'
ORDER BY price DESC
LIMIT 3;
```

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

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
240	Stand-alone content-based hub	Lisa Ellis	Fantasy	1957	49.90	41
462	Innovative 3rdgeneration datab...	Allison Contreras	Fantasy	1988	49.23	62
238	Optimized even-keeled analyzer	Sherri Griffith	Fantasy	1975	48.97	72

QUERY 6 - RETRIEVE THE TOTAL QUANTITY OF BOOKS SOLD BY EACH AUTHOR.

```
Limit to 1000 rows

1  SELECT
2      book.Author, SUM(orderss.Quantity) AS total_quantity_sold
3  FROM
4      online_book_store.book
5          JOIN
6      orderss ON orderss.Book_ID = book.Book_ID
7  GROUP BY book.author;
```

Result Grid | Filter Rows: | Export:

Author	total_quantity_sold
Margaret Moore	8
John Davidson	13
Christopher Fuentes	6
Marissa Smith	16
Christopher Dixon	15
Tonya Saunders	21
Larry Hunt	6

QUERY 7 - LIST THE CITIES WHERE CUSTOMERS WHO SPENT OVER $30 ARE LOCATED.

Limit to 1000 rows

```sql
SELECT DISTINCT
    customers.city, SUM(Total_amount) AS total
FROM
    online_book_store.customers
        JOIN
    orderss ON customers.Customer_ID = orderss.Customer_ID
GROUP BY customers.city
HAVING total > 30;
```

Result Grid | Filter Rows:

city	total
Lake Paul	231.68
North Keith	286.92
Kelseyfort	157.80
East David	301.21
Richardsonville	383.06
Ramosstad	249.40
Rogersborough	480.42
New Carlosbury	144.84

QUERY 8 - FIND THE CUSTOMER WHO SPENT THE MOST ON ORDERS.

```
Limit to 1000 rows

1 •  SELECT
2        customers.name, SUM(Total_amount) AS total_spent
3    FROM
4        online_book_store.customers
5            JOIN
6        orderss ON customers.Customer_ID = orderss.Customer_ID
7    GROUP BY customers.name
8    ORDER BY total_spent DESC
9    LIMIT 1;
```

Result Grid | Filter Rows:

name	total_spent
Kim Turner	1398.90

QUERY 9 - CALCULATE THE STOCK LEFT AFTER FULFILLING ALL ORDERS.

```sql
SELECT
    book.book_id,
    book.title,
    book.stock,
    COALESCE(SUM(orderss.quantity), 0) AS order_quantity,
    book.stock - COALESCE(SUM(orderss.quantity), 0) AS remaining_quantity
FROM
    online_book_store.book
        LEFT JOIN
    orderss ON book.Book_ID = orderss.Book_ID
GROUP BY book.book_id;
```

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

book_id	title	stock	order_quantity	remaining_quantity
1	Configurable modular throughput	100	3	97
2	Persevering reciprocal knowled...	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 imple...	2	0	2

THANK YOU FOR YOUR TIME! 🎉

Presentation on SQL Book Sales Analysis Completed ✅