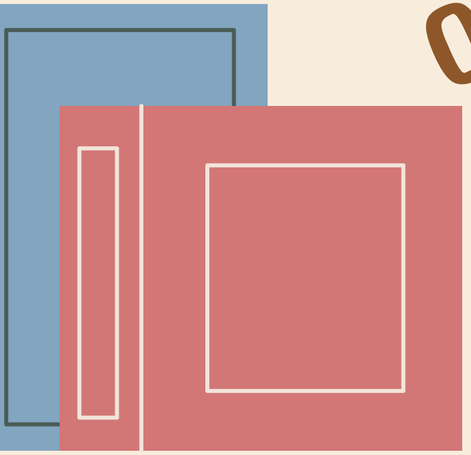
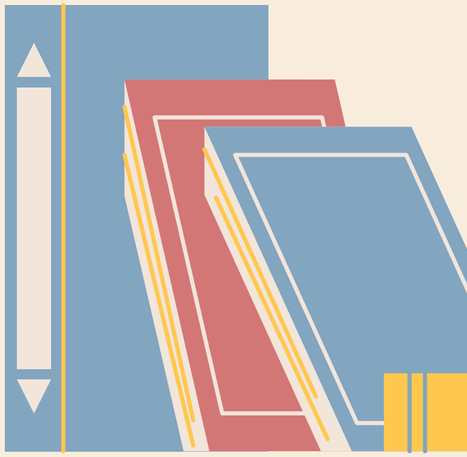
WELCOME TO OUR

BOOK STORE

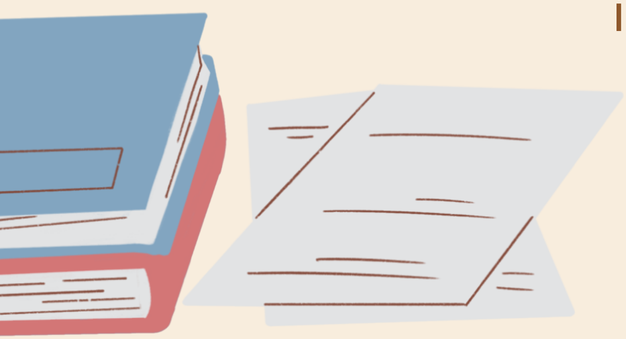
Presentation by
Riya Sharma

ONLINE BOOK STORE DATA ANALYSIS





This project involves analyzing data from an online book store using SQL. The dataset includes information on books, customers, orders. The goal is to extract meaningful insights such as best-selling books, customer purchasing patterns, most active users, and revenue trends. Key SQL operations include joins, aggregations, subqueries . The project helps in understanding how data-driven decisions can enhance marketing strategies, inventory management, and customer engagement.



BASIC SQL QUERIES

1. Retrieve all books in the "Fiction" genre.
2. Find books published after the year 1950.
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 1 quantity of a book.
8. Retrieve all orders where the total amount exceeds $20.
9. List all genres available in the Books table.
10. Find the book with the lowest stock.
11. Calculate the total revenue generated from all orders.

Retrieve all books in the "Fiction" genre

```
SELECT * FROM Books
WHERE Genre='Fiction';
```

Find books published after the year 1950

```
SELECT * FROM Books
WHERE Published_year>1950;
```

List all customers from the Canada

```
SELECT * FROM Customers
WHERE country='Canada';
```

Show orders placed in November 2023

```
SELECT * FROM Orders
WHERE order_date BETWEEN '2023-11-01' AND '2023-11-30';
```

Retrieve the total stock of books available

```
SELECT SUM(stock) AS Total_Stock
From Books;
```

Find the details of the most expensive book

```
SELECT * FROM Books
ORDER BY Price DESC
LIMIT 1;
```

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

```
SELECT * FROM Orders
WHERE quantity>1;
```

Retrieve all orders where the total amount exceeds $20

```
SELECT * FROM Orders
WHERE total_amount>20;
```

List all genres available in the Books table:

```
SELECT DISTINCT genre FROM Books;
```

Find the book with the lowest stock

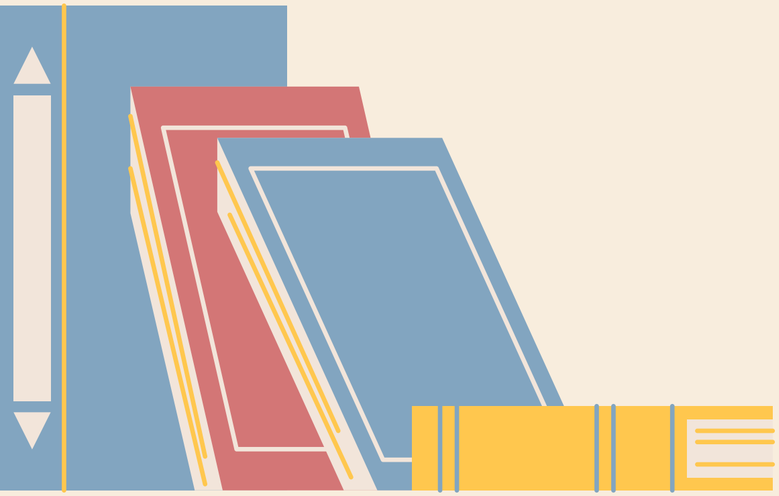
```
SELECT * FROM Books
ORDER BY stock
LIMIT 1;
```

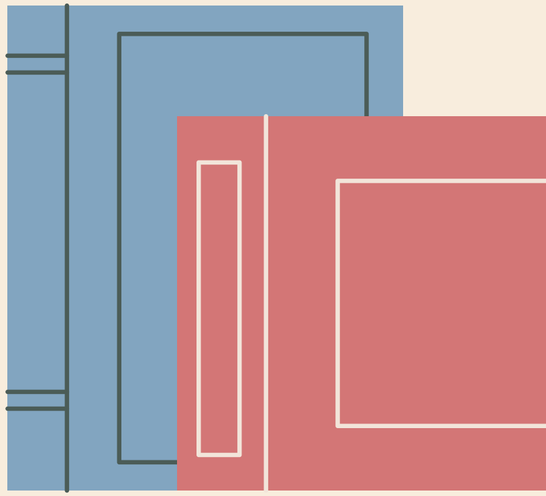
Calculate the total revenue generated from all orders

```
SELECT SUM(total_amount) As Revenue
FROM Orders;
```

ADVANCE SQL QUERIES

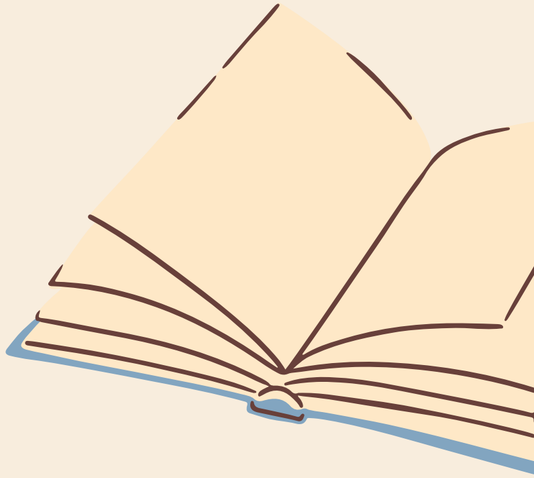
1. Retrieve the total number of books sold for each genre.
2. Find the average price of books in the "Fantasy" genre .
3. List customers who have placed at least 2 orders.
4. Find the most frequently ordered book.
5. Show the top 3 most expensive books of 'Fantasy' Genre .
6. Retrieve the total quantity of books sold by each author.
7. List the cities where customers who spent over $30 are located .
8. Find the customer who spent the most on orders.





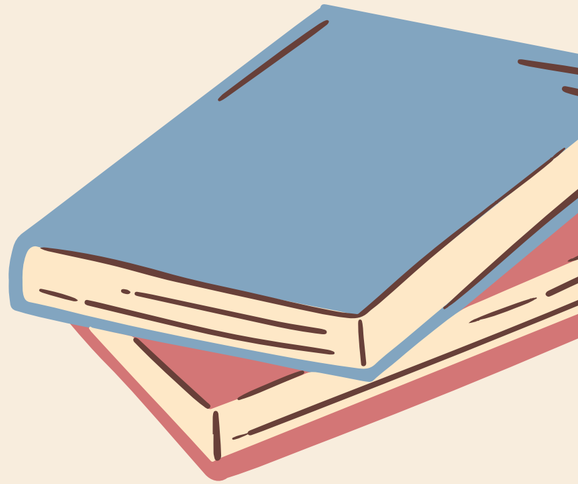
1) Retrieve the total number of books sold for each genre

```sql
SELECT
    books.genre, SUM(orders.Quantity) AS total_books
FROM
    books
        JOIN
    orders ON books.Book_ID = orders.Book_ID
GROUP BY books.genre;
```

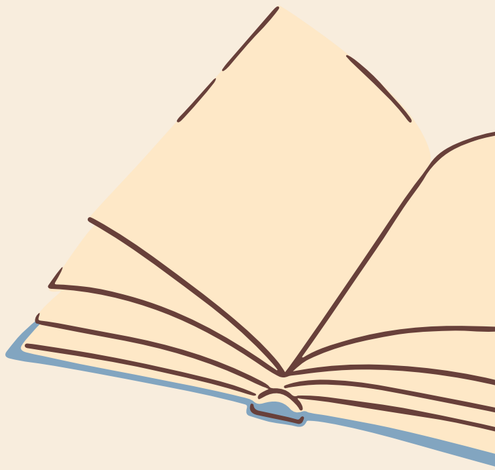


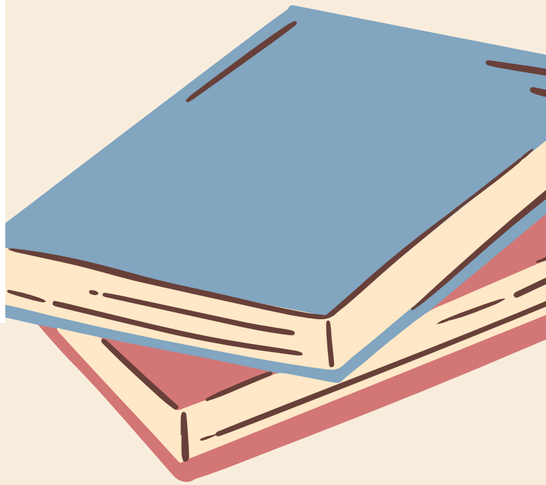
2) Find the average price of books in the "Fantasy" genre

```sql
SELECT
    genre, AVG(price)
FROM
    books
WHERE
    Genre = 'fantasy'
GROUP BY genre;
```



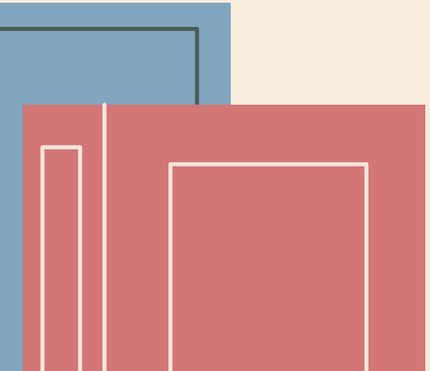
```sql
-- 3) List customers who have placed at least 2 orders:
SELECT
    customers.Customer_ID,
    customers.name,
    COUNT(orders.order_id) AS total_count
FROM
    orders
        JOIN
    customers ON orders.Customer_ID = customers.Customer_ID
GROUP BY customers.name , customers.Customer_ID
HAVING COUNT(order_id) >= 2;
```



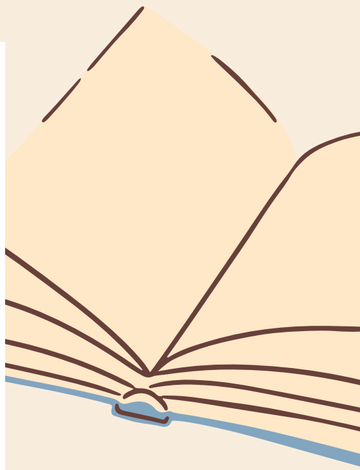


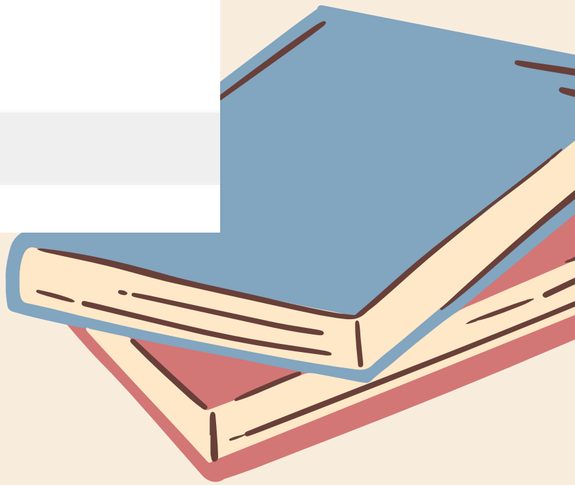
```sql
-- 4) Find the most frequently ordered book:

SELECT
    books.Book_ID,
    books.title,
    COUNT(orders.order_id) AS frequently_order
FROM
    orders
        JOIN
    books ON orders.Book_ID = books.Book_ID
GROUP BY books.Book_ID , books.title
ORDER BY frequently_order DESC
LIMIT 1;
```

```sql
-- 5) Show the top 3 most expensive books of 'Fantasy' Genre :

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



```sql
-- 6) Retrieve the total quantity of books sold by each author:

SELECT DISTINCT
    books.author, SUM(orders.quantity) AS total_quantity
FROM
    books
        JOIN
    orders ON books.Book_ID = orders.Book_ID
GROUP BY books.author;
```





```sql
-- 7) List the cities where customers who spent over $30 are located:

SELECT DISTINCT
    customers.City, orders.Total_Amount AS prices
FROM
    customers
        JOIN
    orders ON customers.Customer_ID = orders.Customer_ID
WHERE
    orders.Total_Amount > 30;
```

```sql
-- 8) Find the customer who spent the most on orders:

SELECT
    customers.Name, SUM(orders.total_amount) AS amount
FROM
    orders
        JOIN
    customers ON customers.Customer_ID = orders.Customer_ID
GROUP BY customers.Name
ORDER BY amount DESC
LIMIT 1;
```

FOR YOUR ATTENTION

THANK YOU