

# SQl PROJECT ON ONLINE BOOK STORE

Presentation by  
Rahul Prajapati

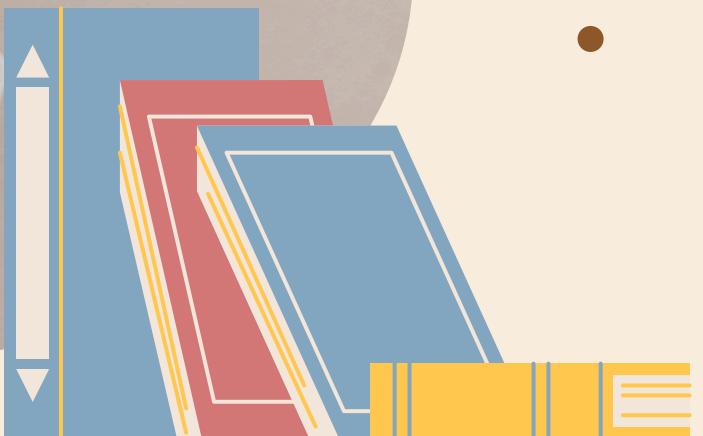


# PROJECT OVERVIEW

This project aims to analyze transactional and customer data from an online bookstore to derive meaningful insights that can help improve business decisions related to sales, inventory, customer engagement, and revenue optimization.

# PROJECT SCOPE

- Work with a relational database containing multiple interconnected tables such as Books, Orders and Customers.
- Solve real-world business queries using SQL, focusing on customer behavior, best-selling books, revenue trends, and product performance.
- Clean, transform, and query data using SQL (PostgreSQL) to produce actionable results.



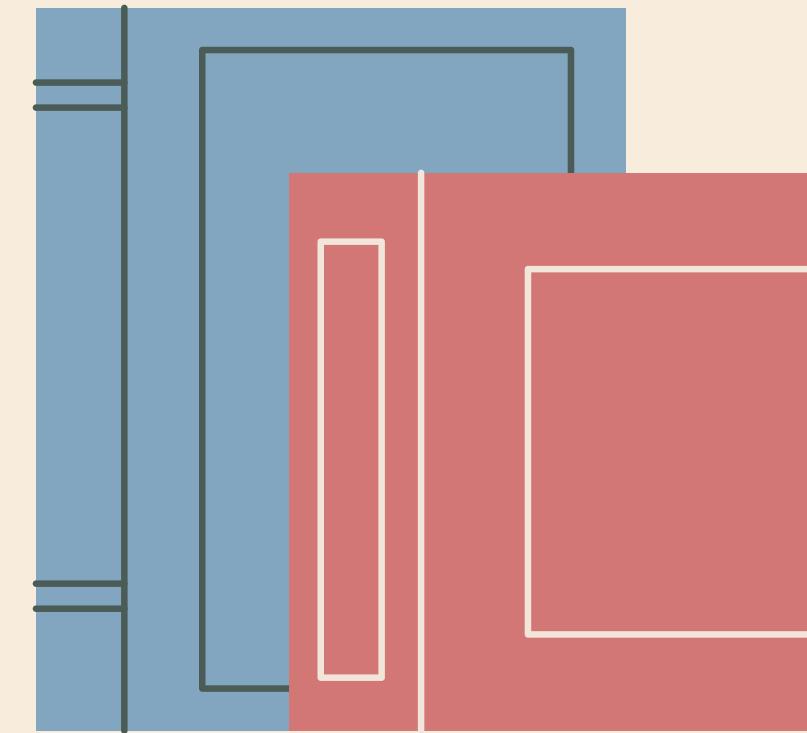
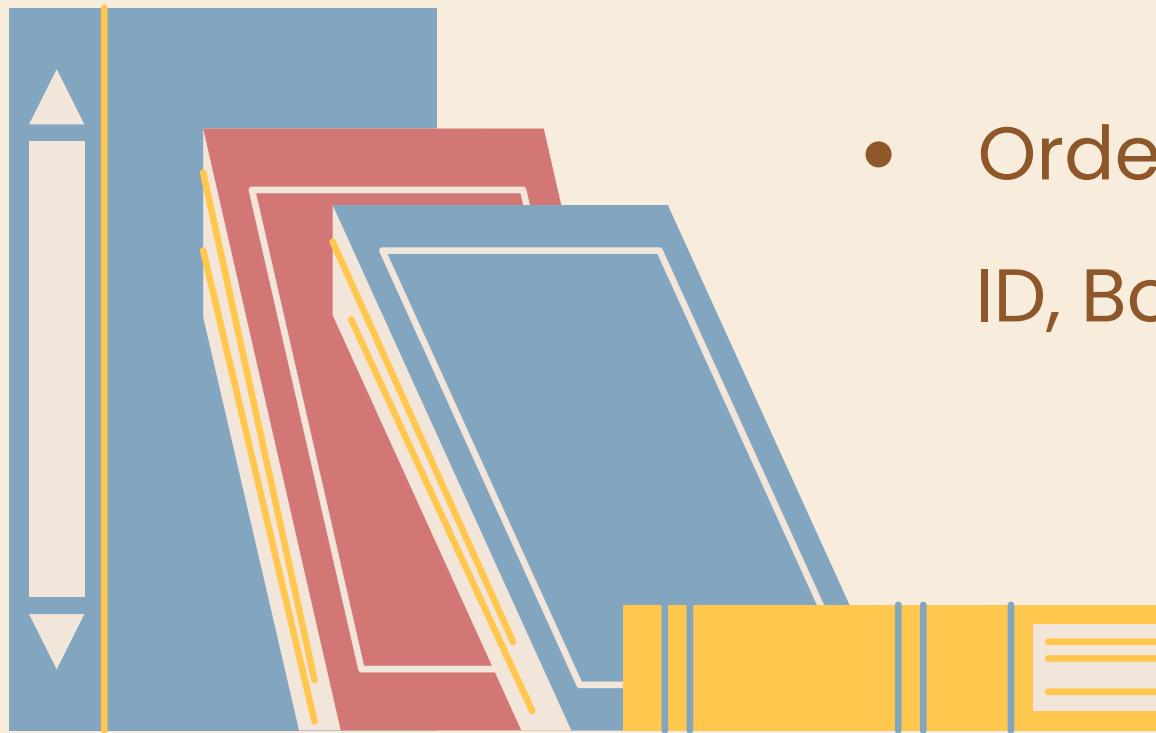
# KEY GOALS

- Identify top-selling books and authors
- Analyze customer purchase patterns
- Evaluate revenue by category, month and year
- Understand inventory and order trends
- Make data-driven recommendations for growth



# TABLES USED

- Books – Book details (Book ID, Title, Author, Price, Genre, Published Year, Stock )
- Customers – Customer details(Customer ID, Name, Email , Phone, City, Country )
- Orders – Order details(Order ID, date, customer ID, Book ID, Order Date. Quantity , Total Amount)



# KEY QUESTIONS SOLVED



1. retrieve all the books in the fiction 'genre'
2. find the books published after the year 1950
3. list all the customers from 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 the customers who ordered more than 1  
quantity of a book
8. retrieve all the orders where the total amount  
exceeds \$20
9. list all the genres available in the books table
10. find the books with the lowest stocks



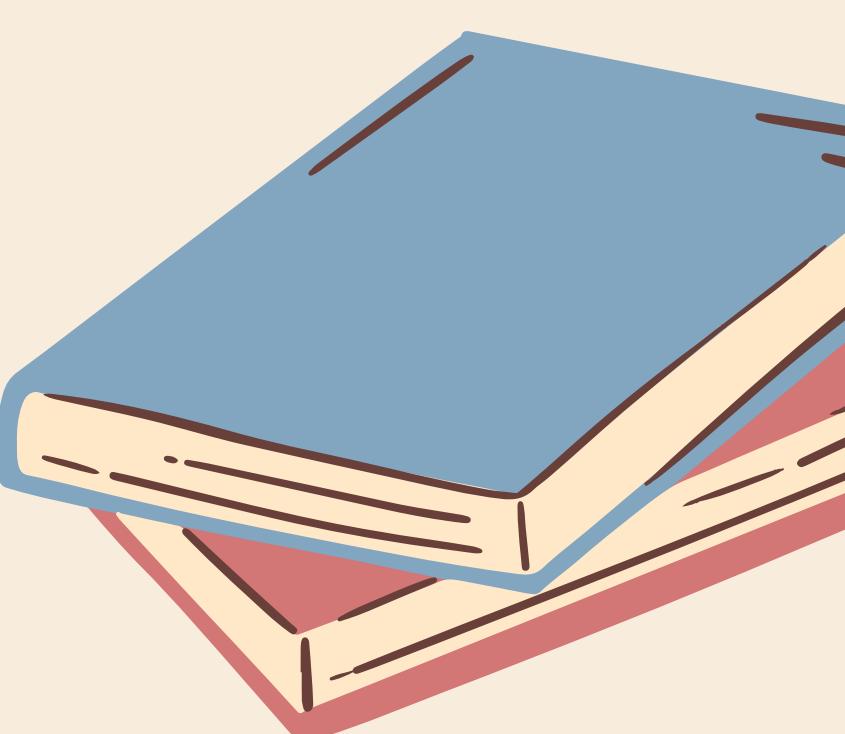
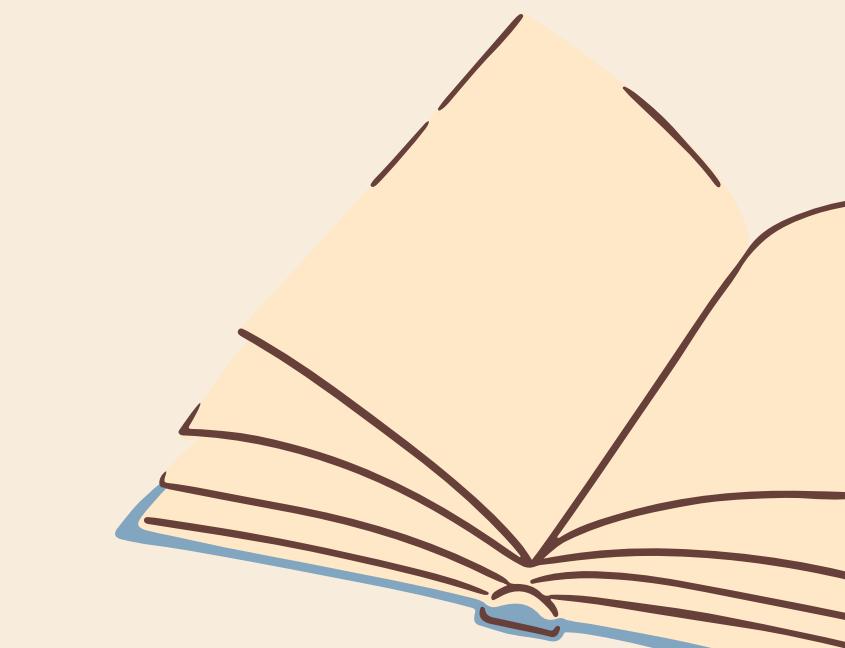
# KEY QUESTIONS SOLVED

- 
- 
- 
11. calculate the total revenue generated from all the orders
  12. retrieve the total number of books sold for each genre
  13. find the average price of books in the 'fantacy' genre
  14. list customers who have placed atleast 2 orders
  15. find the most frequently ordered book
  16. show the top three most expensive books of 'fantasy' genre
  17. retrieve the total quantity of books sold by each author
  18. list the cities where customers who spent over \$30 are located
  19. find the customer who spent most on orders

# Retrieve all the books in the fiction 'genre'

```
SELECT  
    BOOK_ID,  
    TITLE,  
    GENRE  
FROM  
    BOOKS  
WHERE  
    GENRE = 'Fiction';
```





# Find the books published after the year 1950

SELECT

\*

FROM

BOOKS

WHERE

PUBLISHED\_YEAR > 1950;

# List all the customers from canada

SELECT

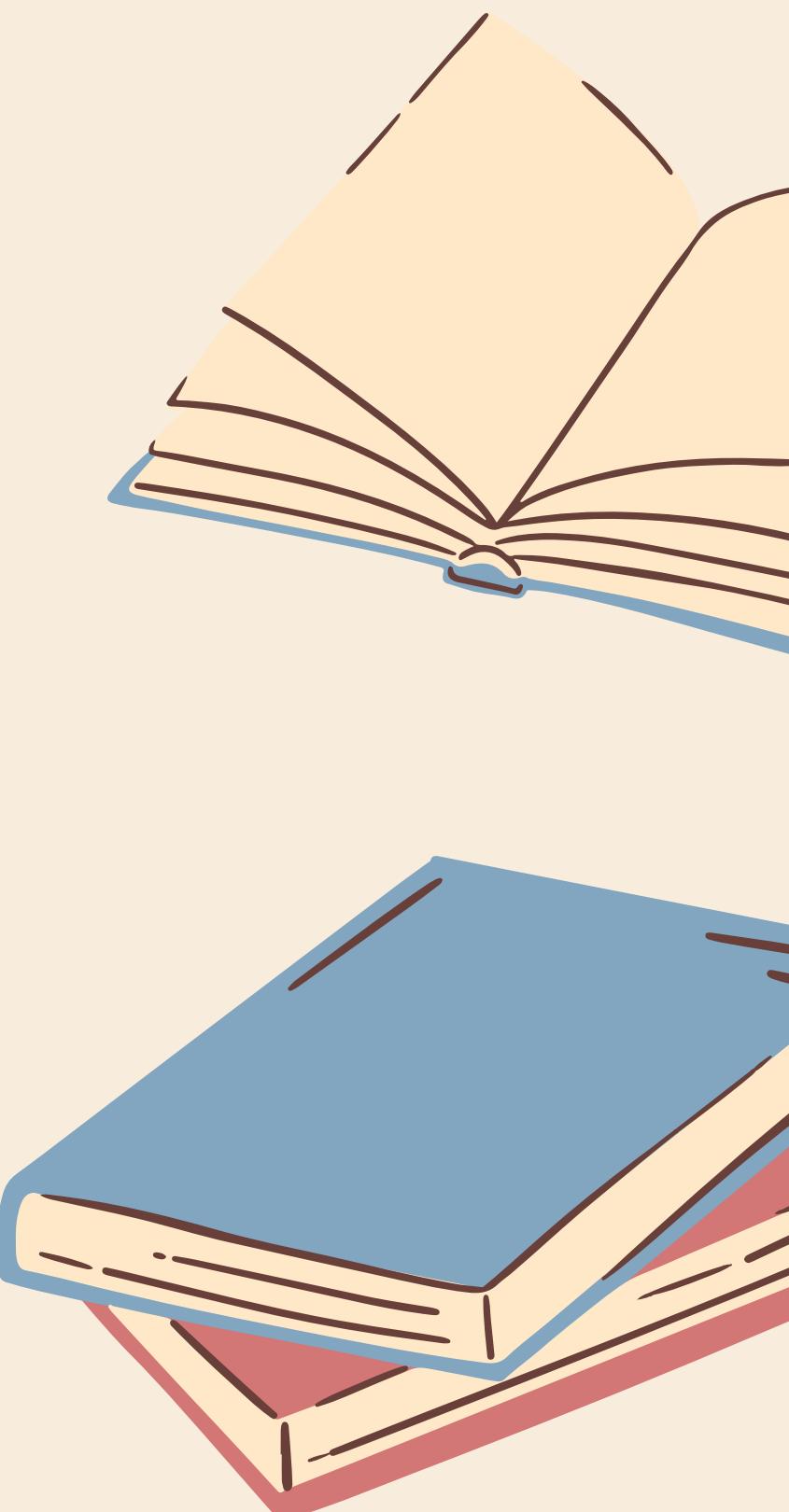
\*

FROM

CUSTOMERS

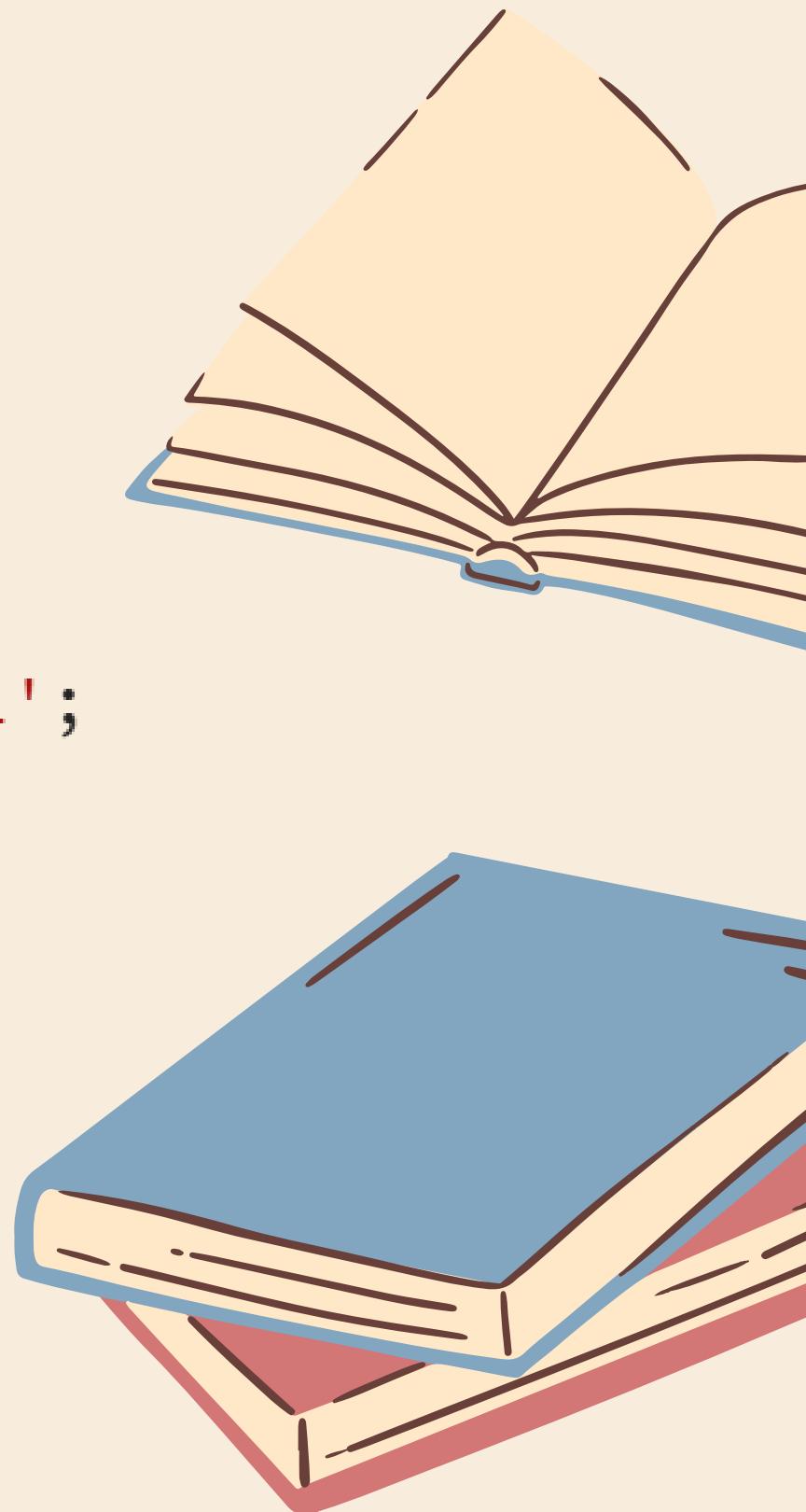
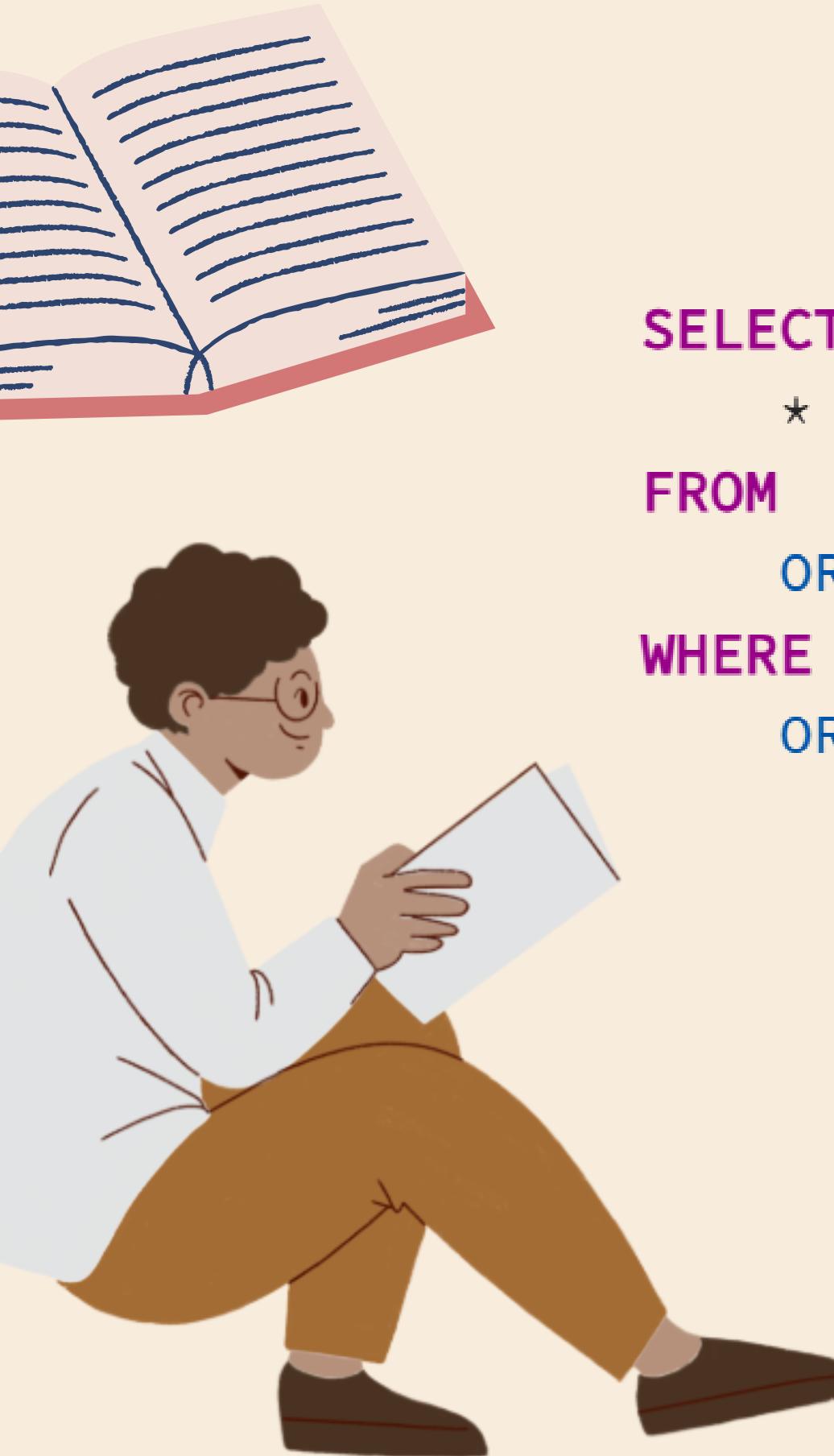
WHERE

COUNTRY = 'Canada' ;



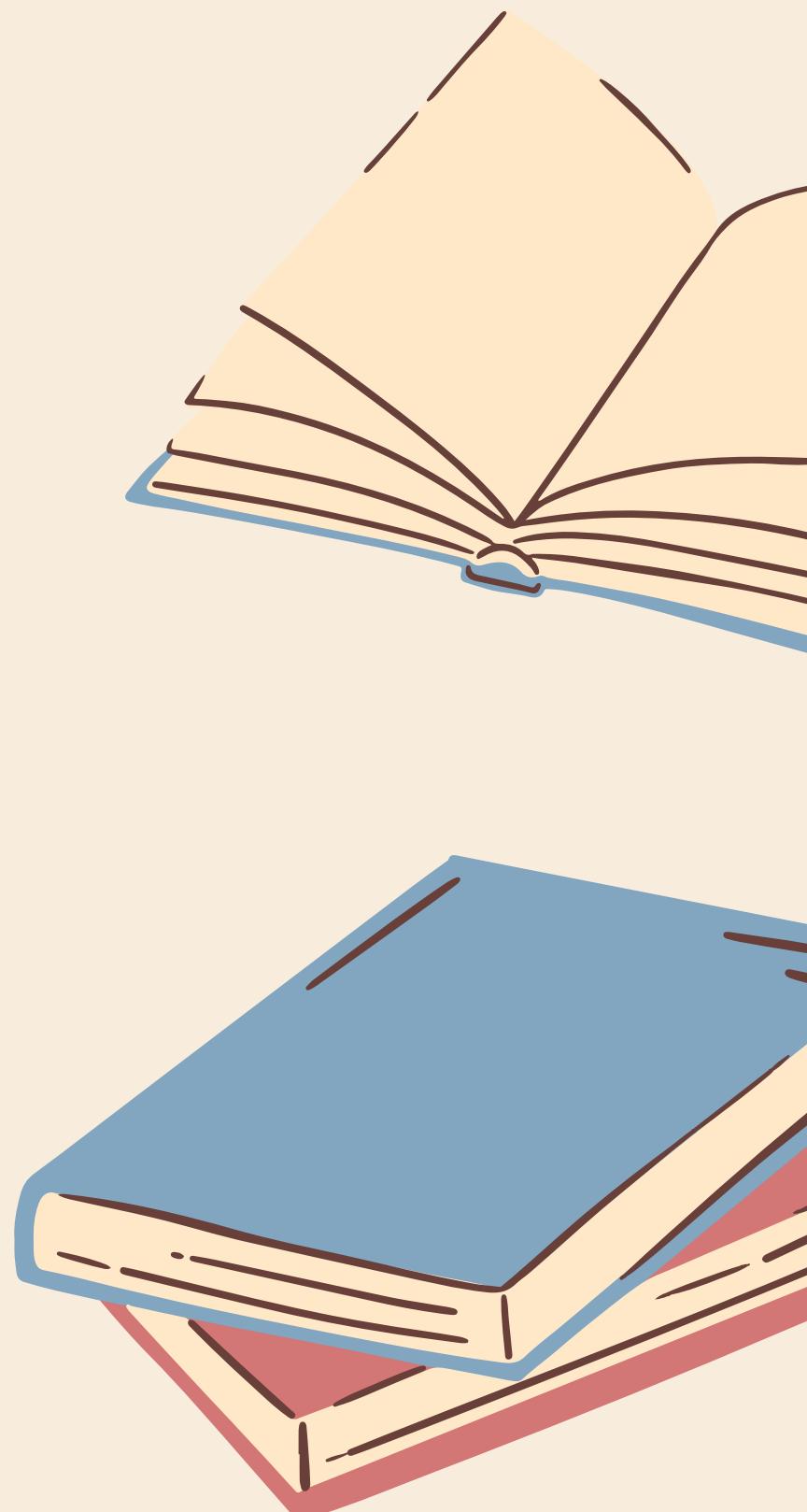
# Show orders placed in november 2023

```
SELECT
  *
FROM
  ORDERS
WHERE
  ORDER_DATE BETWEEN '2023-10-31' AND '2023-12-01';
```



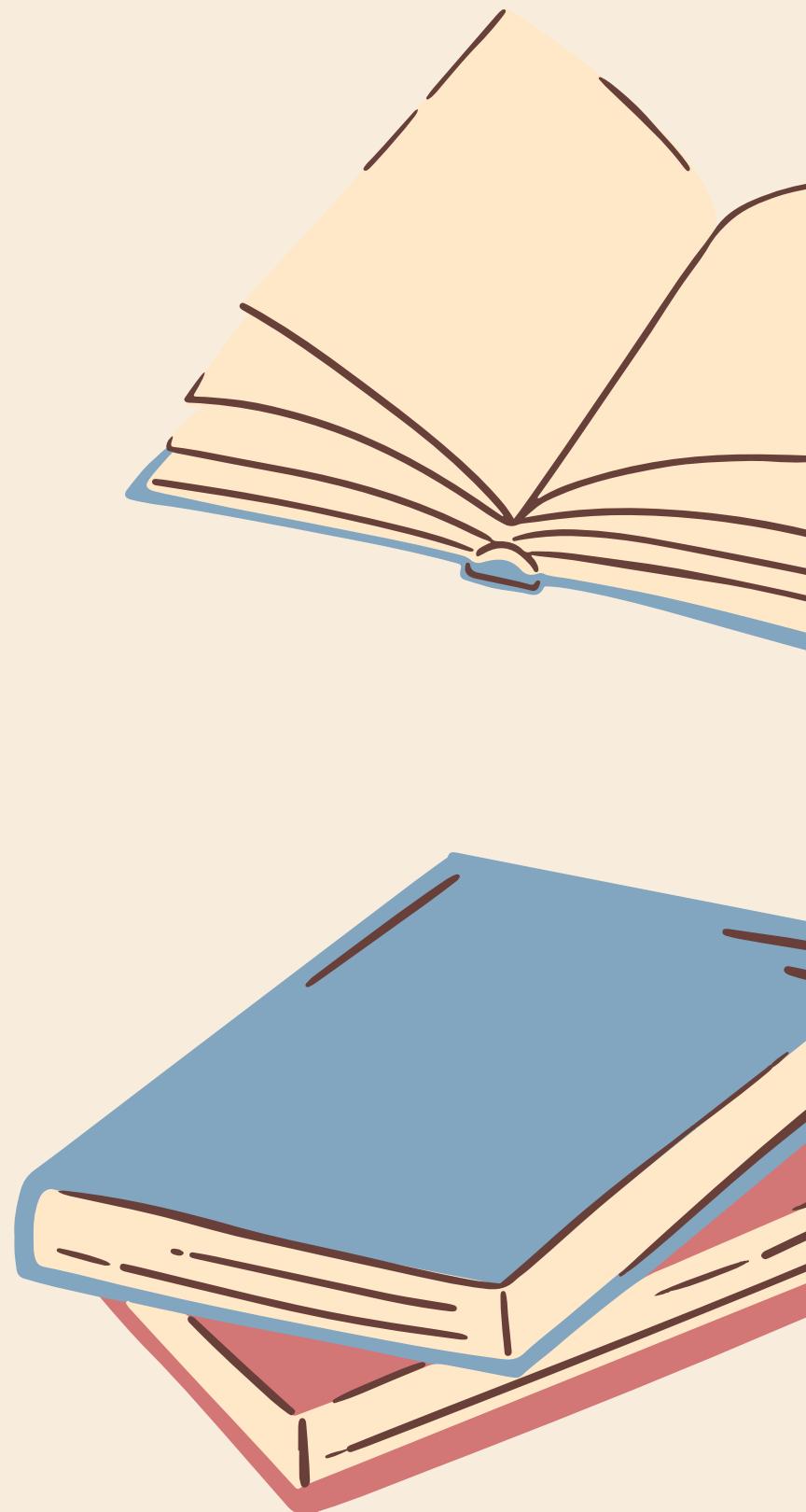
# Retrieve the total stock of books available

```
SELECT  
    SUM(STOCK) AS SUM_OF_STOCKS  
FROM  
    BOOKS;
```



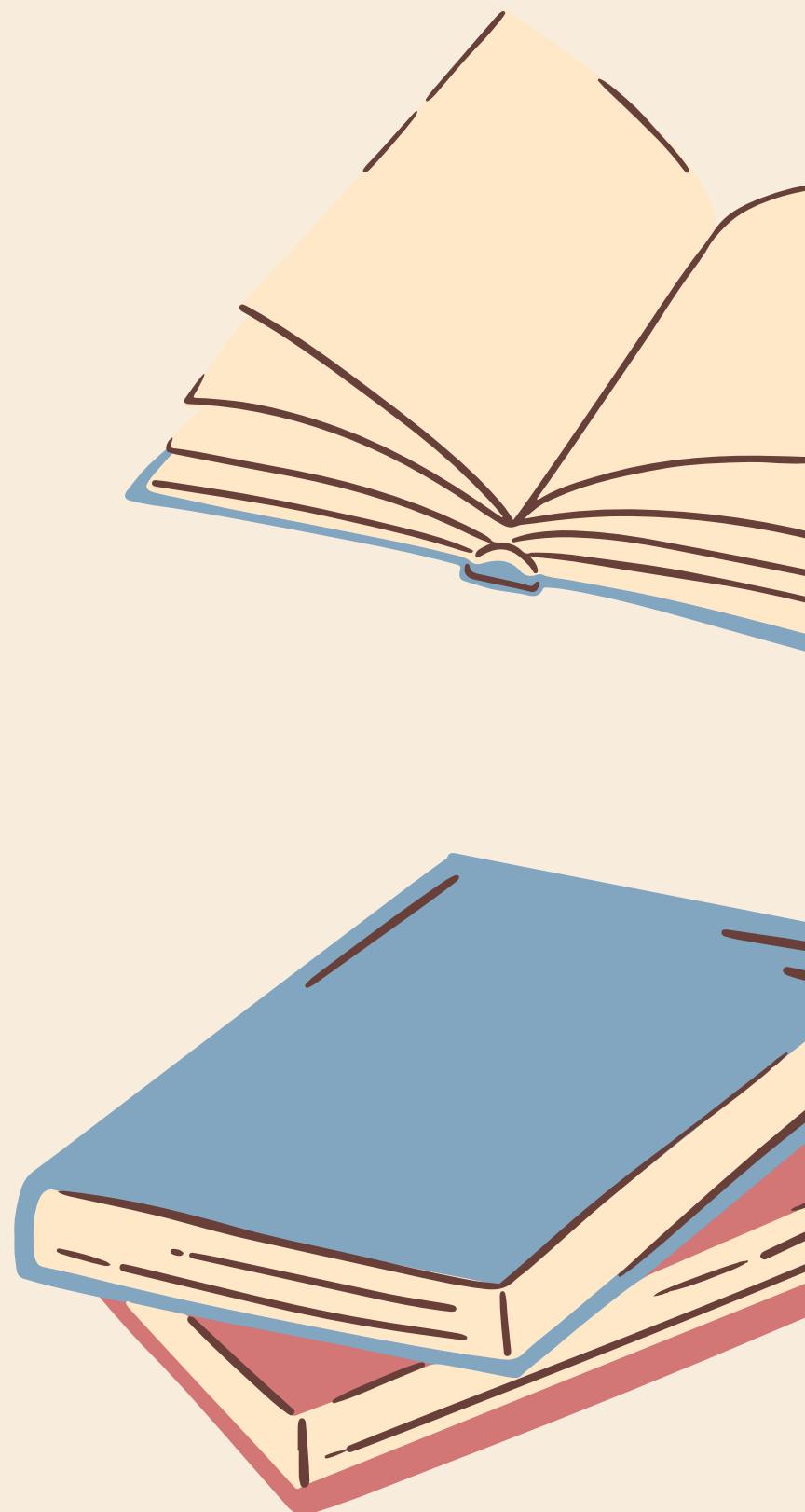
# Find the details of the most expensive book

```
SELECT  
    *  
FROM  
    BOOKS  
ORDER BY  
    PRICE DESC  
LIMIT  
    1;
```



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

```
SELECT
    *
FROM
    ORDERS
WHERE
    QUANTITY >1;|
```



**Retrieve all the orders where the  
total amount exceeds \$20**

**SELECT**

**\***

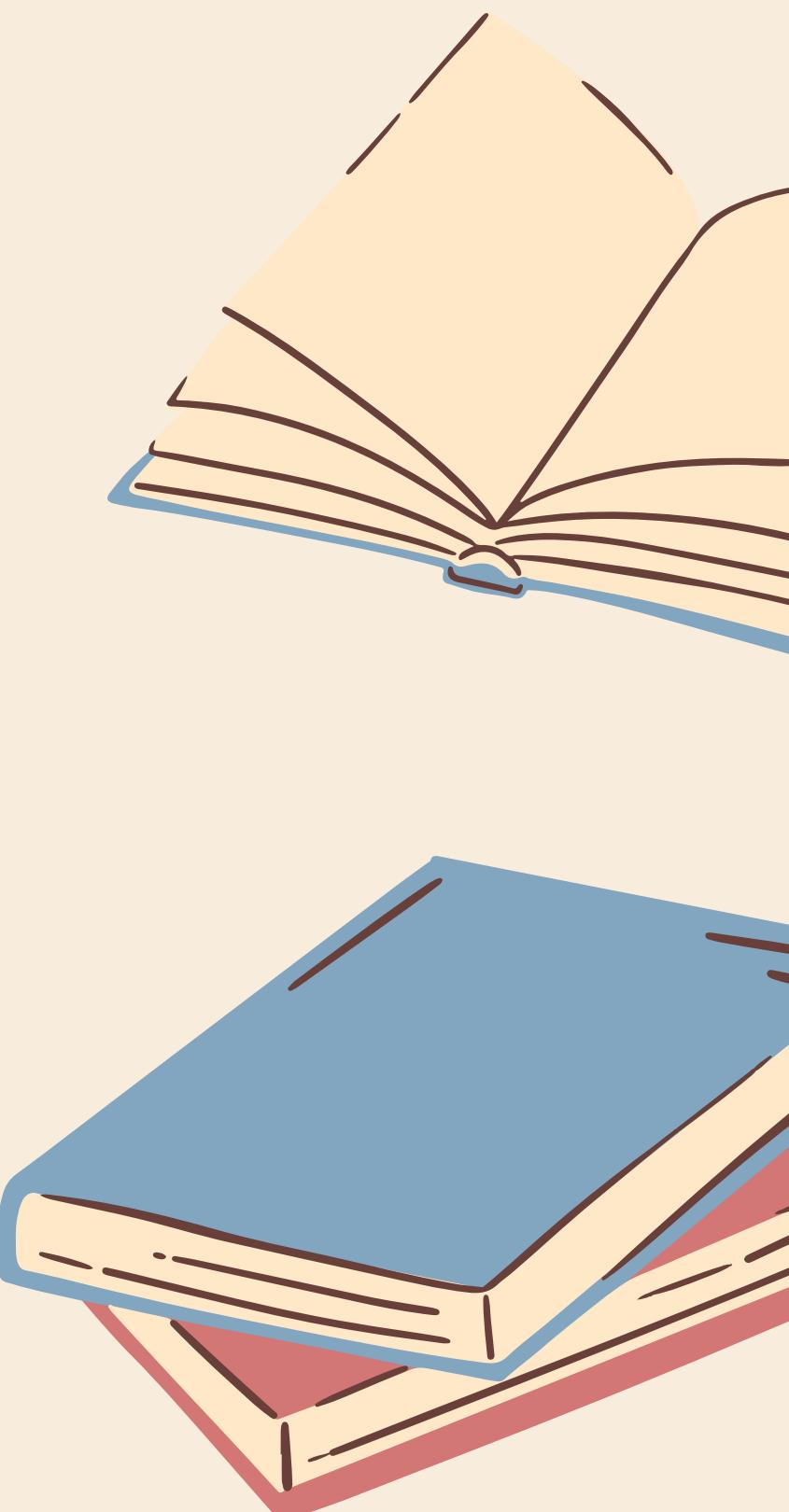
**FROM**

**ORDERS**

**WHERE**

**TOTAL\_AMOUNT >20;**





# List all the genres available in the books table

**SELECT DISTINCT  
GENRE  
FROM  
BOOKS ;**

# Find the books with the lowest stocks

```
SELECT *  
FROM BOOKS  
ORDER BY STOCK  
LIMIT 5;
```



# calculate the total revenue generated from all the orders

```
SELECT  
    SUM(TOTAL_AMOUNT) AS TOTAL_REVENUE  
FROM  
    ORDERS;
```



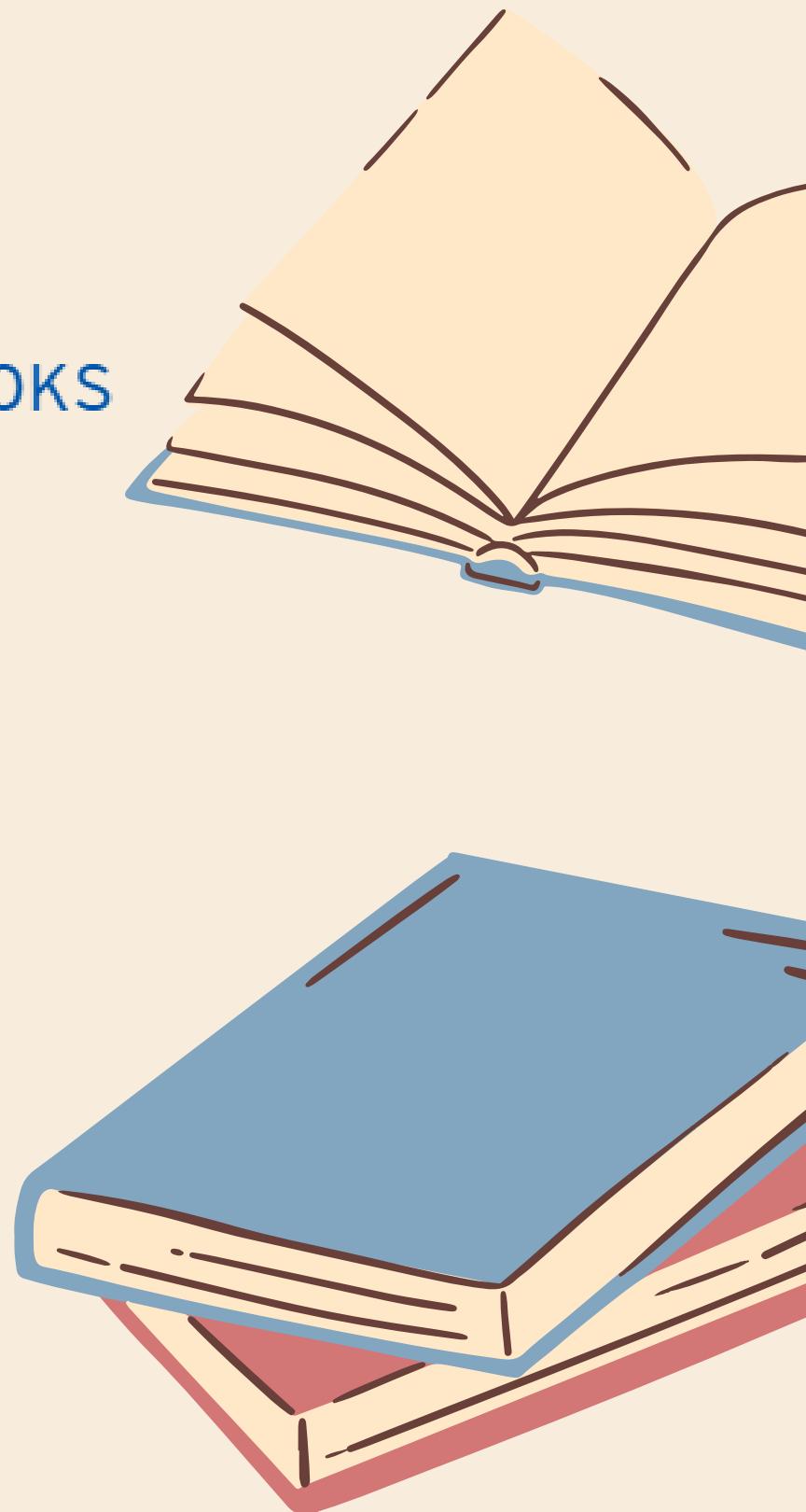
# Retrieve the total number of books sold for each genre

```
SELECT  
    B.GENRE,  
    SUM(O.QUANTITY) AS TOTAL_BOOKS  
FROM  
    BOOKS B  
    JOIN ORDERS O ON B.BOOK_ID = O.BOOK_ID  
GROUP BY  
    B.GENRE;
```



# Find the average price of books in the 'fantacy' genre

```
SELECT  
    AVG(PRICE) AS AVERAGE_PRICE_OF_FANTACY_BOOKS  
FROM  
    BOOKS  
WHERE  
    GENRE = 'Fantasy';
```



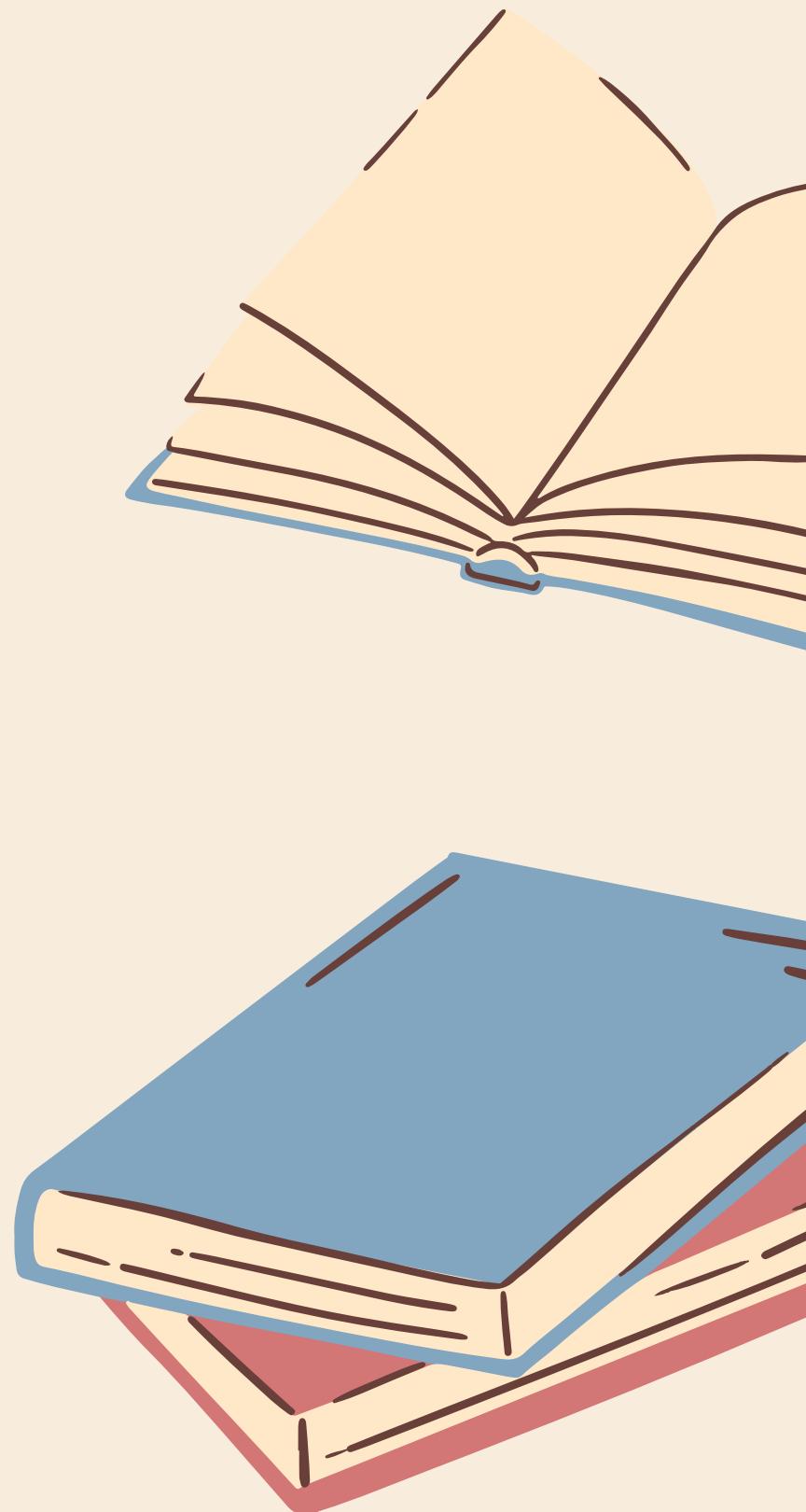
# List customers who have placed atleast 2 orders

```
SELECT  
    CUSTOMER_ID,  
    COUNT(ORDER_ID) AS ORDER_COUNT  
FROM  
    ORDERS  
GROUP BY  
    CUSTOMER_ID  
HAVING  
    COUNT(ORDER_ID) >= 2;
```



# Find the most frequently ordered book

```
SELECT  
    B.TITLE,  
    O.BOOK_ID,  
    COUNT(O.ORDER_ID) AS ORDER_COUNT  
FROM  
    ORDERS O  
    JOIN BOOKS B ON O.BOOK_ID = B.BOOK_ID  
GROUP BY  
    O.BOOK_ID,  
    B.TITLE  
ORDER BY  
    ORDER_COUNT DESC  
LIMIT  
    1;
```



# Show the top three most expensive books of 'fantasy' genre

SELECT

\*

FROM

BOOKS

WHERE

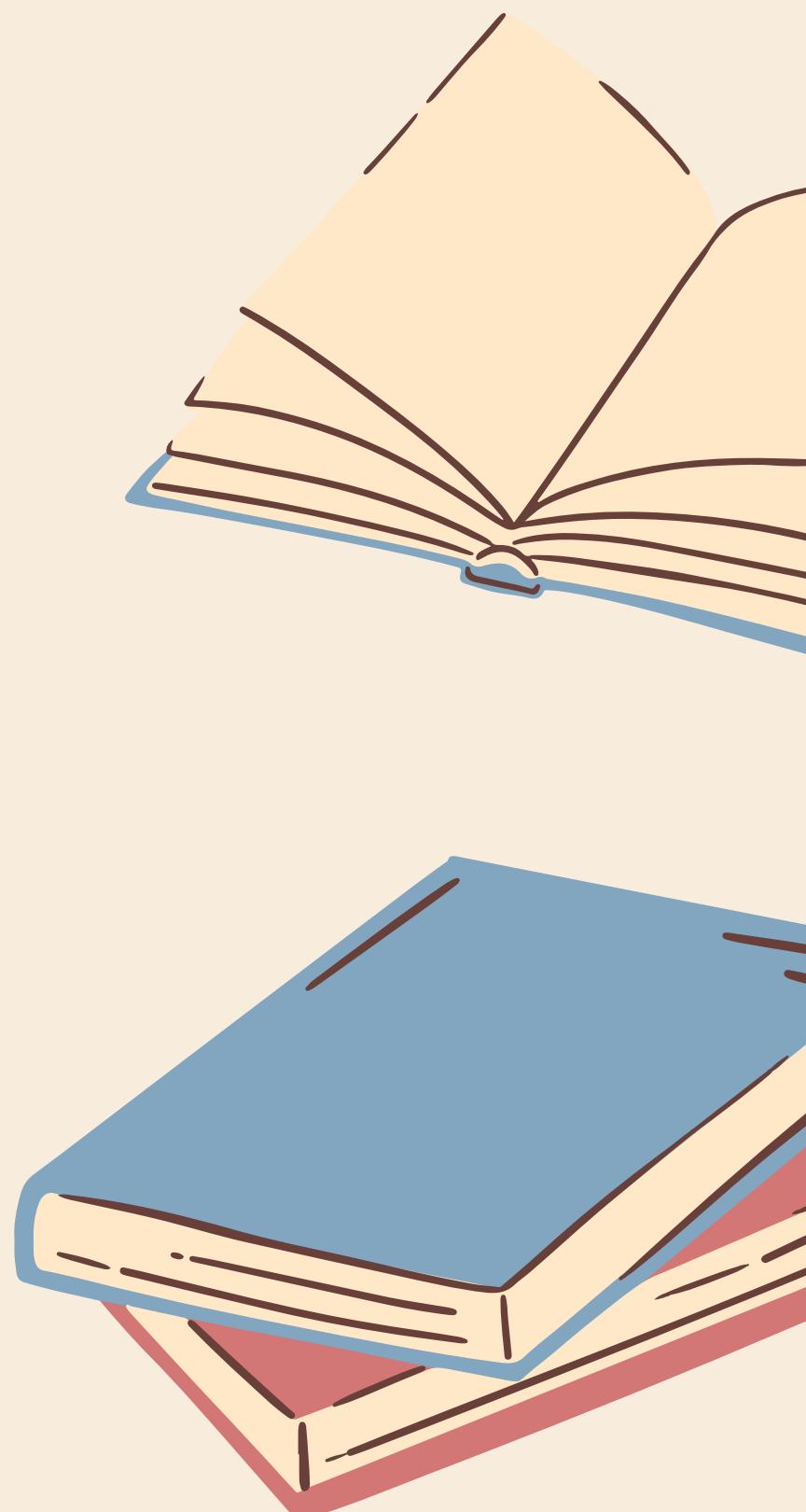
GENRE = 'Fantasy'

ORDER BY

PRICE DESC

LIMIT

3;



# Retrieve the total quantity of books sold by each author

```
SELECT  
    B.AUTHOR,  
    SUM(O.QUANTITY) AS TOTAL_BOOKS_SOLD  
FROM  
    ORDERS O  
    JOIN BOOKS B ON O.BOOK_ID = B.BOOK_ID  
GROUP BY  
    B.AUTHOR;
```



# List the cities where customers who spent over \$30 are located

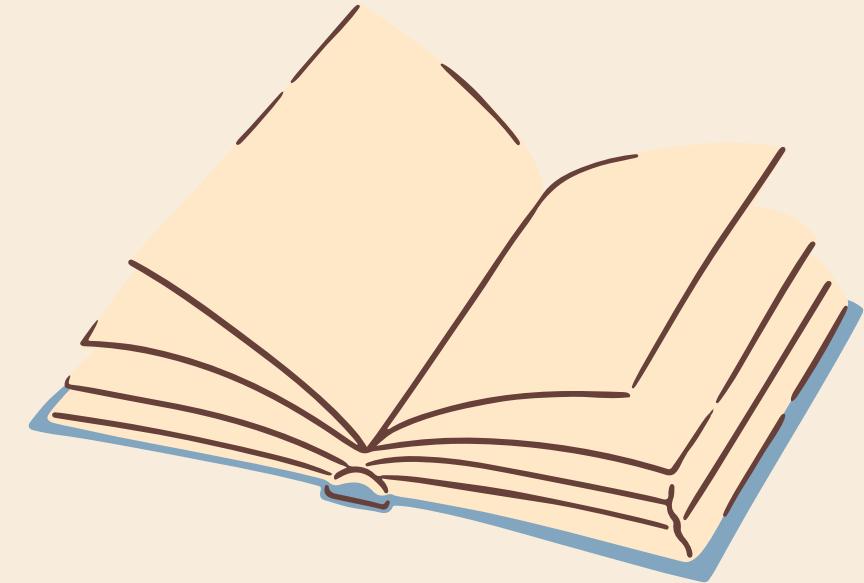
```
SELECT DISTINCT  
    C.CITY,  
    O.TOTAL_AMOUNT  
FROM  
    CUSTOMERS C  
    JOIN ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID  
GROUP BY  
    C.CITY,  
    O.TOTAL_AMOUNT  
HAVING  
    O.TOTAL_AMOUNT >= 30;
```



# Find the customer who spent most on orders

```
SELECT  
    C.CUSTOMER_ID,  
    C.NAME,  
    SUM(O.TOTAL_AMOUNT) AS TOTAL_SPENT  
FROM  
    ORDERS O  
    JOIN CUSTOMERS C ON C.CUSTOMER_ID = O.CUSTOMER_ID  
GROUP BY  
    C.CUSTOMER_ID,  
    C.NAME  
ORDER BY  
    TOTAL_SPENT DESC LIMIT  
    1;
```





**FOR YOUR ATTENTION**

**THANK  
YOU**

Hope that Helps!!