Mini SQL Project: Online Bookstore using MySQL

Just completed a mini SQL project: Online Bookstore using MySQL!

In this project, I practiced and applied my SQL skills through:

Basic queries:

- Retrieve fiction books
- List customers from Canada
- Find books published after 1950
- Show orders placed in November 2023
- Calculate total stock & revenue

Advanced queries:

- Find the most frequently ordered book
- Remaining stock after fulfilling all orders
- Average price of books by genre
- Total quantity sold by author
- Top 3 most expensive fantasy books

Through this project, I improved my understanding of:

- Joins
- Aggregations (SUM, AVG)
- Group By & Having
- Ordering and limiting results

You can check out the complete SQL code here:

[GitHub Gist / Repository link]

Excited to get your feedback and suggestions!

#SQL #MySQL #DataAnalytics #LearningByDoing #PortfolioProject

SQL Queries Used in the Project

```
-- Basic Queries --

SELECT * FROM customers1;
SELECT * FROM books;
SELECT * FROM orders;

SELECT * FROM books WHERE genre='fiction';
SELECT * FROM books WHERE Published_year>1950;
SELECT * FROM customers1 WHERE country='Canada';
SELECT * FROM orders WHERE order_date BETWEEN '2023-11-01' AND '2023-11-30';
SELECT SUM(stock) AS total_books_available FROM books;
SELECT * FROM books ORDER BY price DESC LIMIT 1;
SELECT * FROM orders WHERE quantity>1;
SELECT * FROM orders WHERE Total_amount>20;
SELECT DISTINCT genre FROM books;
SELECT * FROM books ORDER BY stock ASC LIMIT 1;
SELECT SUM(Total_amount) AS total_revenue FROM orders;

-- Advanced Queries --

SELECT b.genre, SUM(o.quantity) AS Total_books_sold
FROM books b JOIN orders o ON b.book_id=o.book_id
GROUP BY b.genre ORDER BY Total_books_sold DESC;

SELECT AVG(price) AS fantasy_books_price_average FROM books WHERE genre='fantasy';

SELECT c.name, COUNT(o.order_id) AS order_count
FROM customers1 c JOIN orders o ON c.customer_id=o.customer_id
GROUP BY c.name HAVING order_count>2;

SELECT b.title, o.book_id, COUNT(o.order_id) AS most_frequently_book
FROM orders o JOIN books b ON b.book_id=o.book_id
GROUP BY book_id, b.title ORDER BY most_frequently_book DESC LIMIT 1;

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

SELECT b.author, SUM(o.quantity) AS total_quantity
FROM books b JOIN orders o ON b.book_id=o.book_id
GROUP BY b.author ORDER BY total_quantity DESC;

SELECT DISTINCT c.city, o.total_amount
FROM customers1 c JOIN orders o ON c.customer_id=o.customer_id
WHERE o.total_amount>=30;
```

```
SELECT c.customer_id, c.name, SUM(o.total_amount) AS most_spent
FROM customers1 c JOIN orders o ON c.customer_id=o.customer_id
GROUP BY c.name, c.customer_id ORDER BY most_spent DESC LIMIT 1;

SELECT b.book_id, b.title, b.stock,
COALESCE(SUM(o.quantity), 0) AS total_quantity,
b.stock - COALESCE(SUM(o.quantity), 0) AS remaining_stock
FROM books b LEFT JOIN orders o ON b.book_id = o.book_id
GROUP BY b.book_id, b.title, b.stock;
```