Online Book Store

Rakesh kumar yadav

Introduction

The main goal of this project is to apply SQL concepts to manage and analyze data from an online book store. The dataset includes information about books, customers, and orders, allowing us to perform both basic and advanced SQL queries

What This Project Covers:

- Retrieving, filtering, and sorting data using SELECT, WHERE, and ORDER BY
- Aggregating data with SUM, COUNT, AVG, and GROUP BY
- Joining multiple tables to gain deeper insights
- Solving real-world business problems such as tracking sales, analyzing customer behavior, and identifying best-selling products

Tools Used:

- MySQL
- CSV datasets for Books, Customers, and Orders

1) Retrieve all books in the "Fiction" genre:

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

#2) Find books published after the year 1950:

```
SELECT
    *
FROM
    books
WHERE
    Published_Year > 1949;
```

#3) List all customers from the Canada:

```
SELECT
    *
FROM
    customerss
WHERE
    country = 'canada';
```

#4>Show orders placed in November 2023:

```
SELECT
   *
FROM
   orderss
WHERE
        Order_Date  BETWEEN  '2023-11-1'
AND '2023-11-31';
```

5) Retrieve the total stock of books available:

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

6) Find the details of the most expensive book:

```
SELECT
    *
FROM
    books
ORDER BY price DESC
LIMIT 1;
```

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

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

#8> Retrieve all orders where the total amount exceeds $20:

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

9) List all genres available in the Books table

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

#10) Find the book with the lowest stock:

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

11) Calculate the total revenue generated from all orders:

```
SELECT
   SUM(Total_Amount)
FROM
   orderss;
```

#ADVANCED QUERY

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

```
SELECT
        b.Genre,    SUM(o.Quantity)    AS
Total_Books_sold
FROM
   Orderss o
      JOIN
   Books b ON o.book_id = b.book_id
GROUP BY b.Genre;
```

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

```
SELECT
    AVG(Price) AS avg_of_fantasy
FROM
    books
WHERE
    Genre = 'Fantasy';
```

#3)List customers who have placed at least 2 orders:

```
SELECT
                    o.customer_id,      c.name,
COUNT(o.Order_id) AS ORDER_COUNT
FROM
   orderss AS o
      JOIN
    customerss AS c ON c.Customer_ID =
o.Customer_ID
GROUP BY o.customer_id , c.name
HAVING COUNT(Order_id) >= 2;
```

-- 4) Find the most frequently ordered book:

```
SELECT
    o.Book_id, b.title, COUNT(o.order_id)
AS ORDER_COUNT
FROM
   orderss 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;
```

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

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

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

```
SELECT
   b.Author, SUM(o.quantity)
FROM
   books AS b
      JOIN
         orderss AS o ON o.Book_ID =
b.Book_ID
GROUP BY b.Author;
```

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

```
SELECT DISTINCT
    c.city, o.Total_Amount
FROM
    orderss AS o
        JOIN
    customerss AS c ON c.Customer_ID =
o.Customer_ID
WHERE
    Total_Amount > 30;
```

8) Find the customer who spent the most on orders:

```
SELECT
                    c.Customer_ID,      c.Name,
SUM(o.Total_Amount) AS total_spend
FROM
   orderss AS o
      JOIN
    customerss AS c ON c.Customer_ID =
o.Customer_ID
GROUP BY c.Customer_ID , c.Name
ORDER BY total_spend DESC
LIMIT 1;
```