SQL PROJECT REPORT

Bookstore Sales & Customer Analysis using SQL

Name: Pranjal Tiwari
Database: PostgreSQL
Project Type: SQL Data Analysis Project

---

1. Project Overview

This project focuses on analyzing a **bookstore database** using **SQL** to extract meaningful business insights related to **sales, customers, revenue, and inventory management**.

The dataset was provided in Excel (CSV) format and imported into PostgreSQL. Using structured SQL queries, various real-world business questions were answered to understand customer behavior, book demand, and stock performance.

---

2. Dataset Description

The project uses three datasets:

Books.csv – Contains information about books, genres, prices, and stock.

Customers.csv – Contains customer details and location.

Orders.csv – Contains order details, quantities, and total amount.

---

3. Database Schema Design

Books Table:

```
CREATE  TABLE Books (
   Book_ID  SERIAL PRIMARY KEY,
   Title  VARCHAR(100),
   Author  VARCHAR(100),
   Genre  VARCHAR(50),
   Published_year  INT,
   Price  NUMERIC(10,2),
   Stock  INT
);
```

Customers Table:

```
CREATE TABLE Customers (
   Customer_ID SERIAL PRIMARY KEY,
   Name VARCHAR(100),
   Email VARCHAR(100),
   Phone VARCHAR(15),
   City VARCHAR(50),
   Country VARCHAR(100)
);
```

Orders Table:

```
CREATE TABLE Orders (
   Order_Id SERIAL PRIMARY KEY,
   Customer_Id INT REFERENCES Customers(Customer_ID),
   Book_ID INT REFERENCES Books(Book_ID),
   Order_Date DATE,
   Quantity INT,
   Total_Amount NUMERIC(10,2)
);
```

Queries:

```
Select * FROM  Books ;

Select * FROM Customers;

Select * FROM Orders;
```

4. SQL Queries & Analysis

Q1. Retrieve all books in the Fiction genre .

Answer:

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

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 60 | Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
2	22	Multi-layered optimizing migration	Wesley Escobar	Fiction	1908	39.23	78
3	28	Expanded analyzing portal	Lisa Coffey	Fiction	1941	37.51	79
4	29	Quality-focused multi-tasking challenge	Katrina Underwood	Fiction	1905	31.12	100
5	31	Implemented encompassing conglomerat...	Melissa Taylor	Fiction	2010	21.23	44

Q2. Find books published after 1950 .

Answer:

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

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 292 | Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19
2	4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
3	5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	10.95	16
4	6	Advanced encompassing implementation	Bryan Morgan	Biography	1985	6.56	2
5	8	Persistent local encoding	Troy Cox	Science Fiction	2019	48.99	84

Q3. List all customers from Canada

Answer:

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

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 3 | Page No: 1

	customer_id [PK] integer	name character varying (100)	email character varying (100)	phone character varying (15)	city character varying (50)	country character varying (100)
1	38	Nicholas Harris	christine93@perkins.com	1234567928	Davistown	Canada
2	415	James Ramirez	robert54@hall.com	1234568305	Maxwelltown	Canada
3	468	David Hart	stokesrebecca@gmail.c...	1234568358	Thompsonfurt	Canada

Q4. Show orders placed in November 2023 .

Answer:

```
SELECT * FROM Orders
WHERE Order_Date >= '2023-11-01'
AND Order_Date < '2023-12-01';
```

Output:

Data Output | Messages | Notifications

	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_amount numeric (10,2)
1	4	433	343	2023-11-25	7	301.21
2	19	496	60	2023-11-17	9	316.26
3	75	291	375	2023-11-30	5	170.75
4	132	469	333	2023-11-22	7	194.32
5	137	474	471	2023-11-25	8	363.04

Q5. Retrieve the total stock of books available .

Answer:

```
SELECT SUM(Stock) AS Total_Stock
FROM Books;
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	total_stock bigint
1	25056

Q6. Find the most expensive book .

Answer:

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

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	340	Proactive system-worthy orchestrati...	Robert Scott	Mystery	1907	49.98	88

Q7. Show customers who ordered more than 1 quantity .

Answer:

```
SELECT c.Name AS Customer_Name, o.Quantity
FROM Customers c  JOIN Orders o
ON c.Customer_ID = o.Customer_ID
WHERE o.Quantity > 1;
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 438 | Page No: 1 of 1

	customer_name character varying (100)	quantity integer
1	Gary Blair	8
2	Steven Miller	10
3	Phillip Allen	6
4	Corey Wells	7
5	John Wood	7

Q8. Retrieve orders where total amount exceeds 200 .

Answers:

```
SELECT * FROM Orders
WHERE Total_Amount > 200;
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 150 | Page No: 1 of 1

	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_amount numeric (10,2)
1	2	137	301	2023-01-23	10	216.60
2	4	433	343	2023-11-25	7	301.21
3	6	439	119	2024-10-11	5	249.40
4	9	109	407	2024-01-04	9	379.71
5	15	127	479	2023-01-10	6	229.62

Q9. List all available book genres .

Answer:

```
SELECT DISTINCT Genre FROM Books;
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 7 | Page No: 1 of 1

	genre character varying (50)
1	Romance
2	Biography
3	Mystery
4	Fantasy
5	Fiction

Q10. Find the book with the lowest stock .

Answer:

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

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	44	Networked systemic implementati...	Ryan Frank	Science Fiction	1965	13.55	0

Q11. Calculate total revenue from all orders .

Answer:

```
SELECT SUM(Total_Amount) AS Total_Revenue
FROM Orders;
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	total_revenue numeric
1	75628.66

Advanced SQL Analysis

Q1. Total number of books sold per genre .

Answer:

```
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;
```

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 7 | Page No: 1

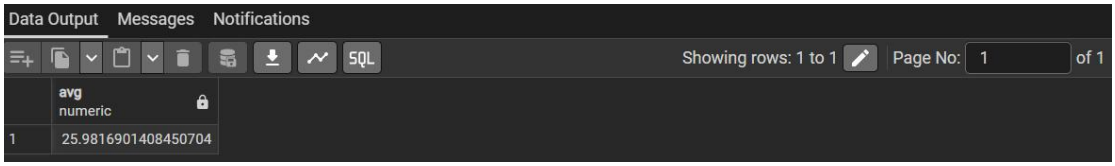
	genre character varying (50)	total_book_sold bigint
1	Romance	439
2	Biography	285
3	Mystery	504
4	Fantasy	446
5	Fiction	225

Q2. Average price of Fantasy genre books .

Answer:

```
SELECT AVG(Price) AS Avg_Price
FROM Books WHERE Genre = 'Fantasy';
```

Output:

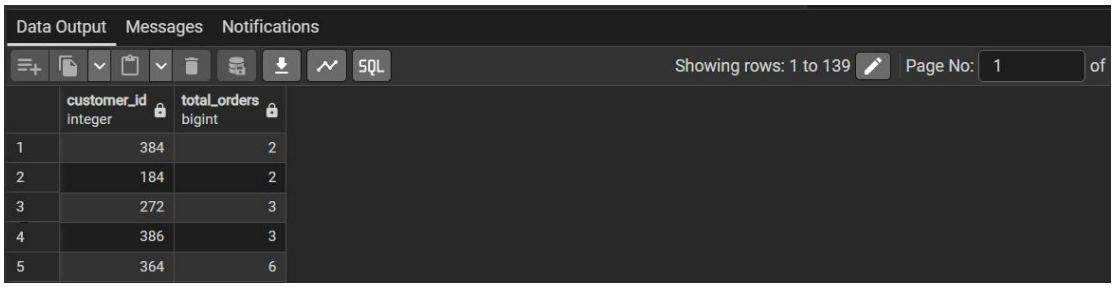


Q3. Customers with at least 2 orders .

Answer:

```
SELECT c.Customer_ID, c.Name,
COUNT(o.Order_Id) AS Total_Orders
FROM Customers c
JOIN Orders o ON c.Customer_ID = o.Customer_ID
GROUP BY c.Customer_ID, c.Name
HAVING COUNT(o.Order_Id) >= 2;
```

Output:

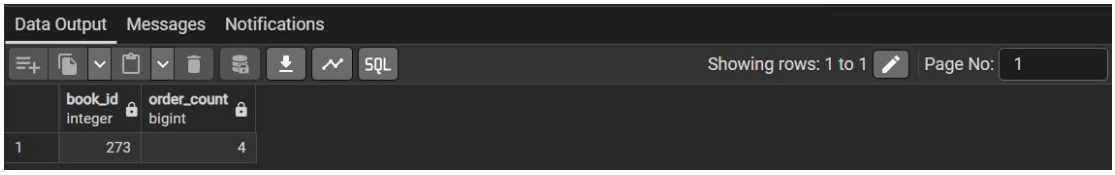


Q4. Most frequently ordered book .

Answer:

```
SELECT b.Title, COUNT(o.Order_Id) AS Order_Count
FROM Books b JOIN Orders o ON b.Book_ID = o.Book_ID
GROUP BY b.Title  ORDER BY Order_Count DESC
LIMIT 1;
```

Output:

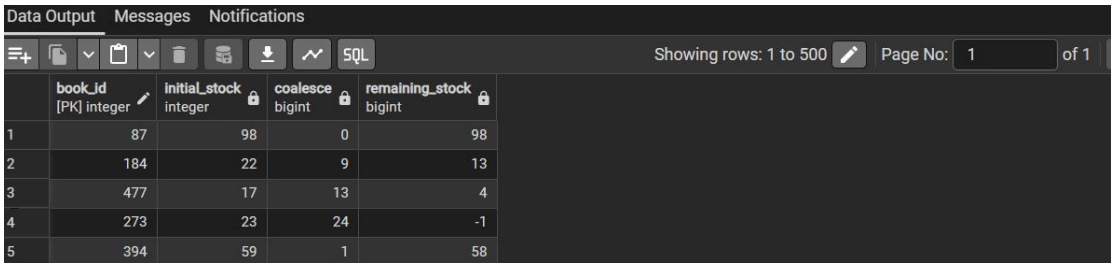


Q5. Remaining stock after fulfilling all orders .

Answer:

```
SELECT  b.Book_ID,
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.Stock;
```

Output:



Data Output | Messages | Notifications

Showing rows: 1 to 500 | Page No: 1 of 1

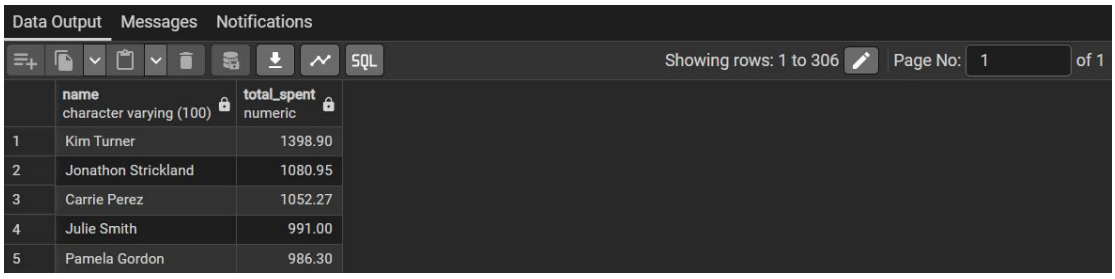
	book_id [PK] integer	initial_stock integer	coalesce bigint	remaining_stock bigint
1	87	98	0	98
2	184	22	9	13
3	477	17	13	4
4	273	23	24	-1
5	394	59	1	58

Q6. Find the customer who spent the most on orders .

Answer:

```
Select c.name,Sum(o.total_amount) As total_spent
from customers c
join orders o
on c.customer_id=o.customer_id
group by c.name
order by total_spent Desc;
```

Output:



Data Output | Messages | Notifications

Showing rows: 1 to 306 | Page No: 1 of 1

	name character varying (100)	total_spent numeric
1	Kim Turner	1398.90
2	Jonathon Strickland	1080.95
3	Carrie Perez	1052.27
4	Julie Smith	991.00
5	Pamela Gordon	986.30

---

5. Conclusion

This project demonstrates the practical use of SQL for data analysis, including database design, joins, aggregations, and business problem-solving. It reflects real-world scenarios such as sales analysis, customer behavior tracking, and inventory management.

Skills Used

- SQL (PostgreSQL)
- Joins & Subqueries
- Aggregation Functions
- Data Analysis
- Relational Database Design