PRESENTATION

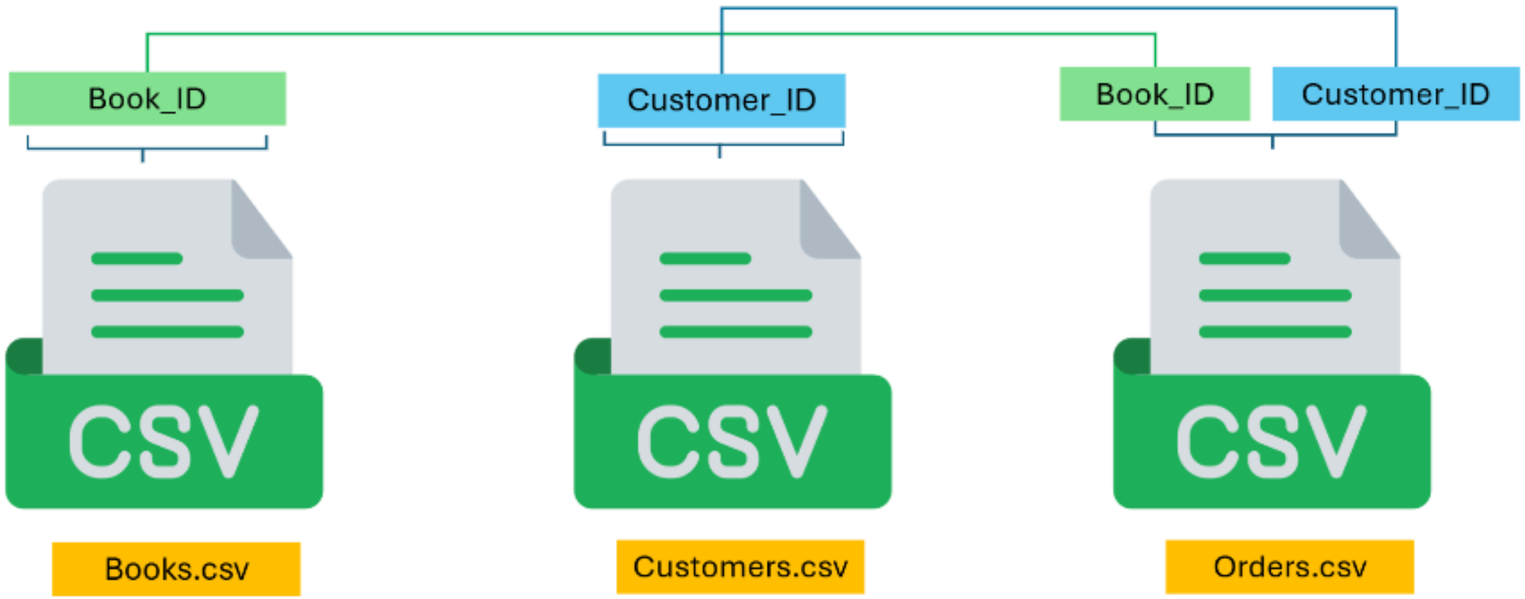
By Altamash Patil

PROJECT TITLE: ONLINE BOOK STORE

Hello, I'm [Altamash], and I'm excited to work on this SQL project analyzing online book store data. The goal of this project is to gain insights into customer behavior, sales trends, and book performance. By analyzing the data, we can identify areas of improvement and provide recommendations to stakeholders in the online book store industry.

3 CSV Files

Tables must have at least one common column with same column name and same data type



Basic 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

Advance 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

```sql
CREATE DATABASE OnlineBookstore;

use OnlineBookstore;
```

```sql
-- Create Tables
DROP TABLE IF EXISTS Books;
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
);
```

```sql
DROP TABLE IF EXISTS customers;
CREATE TABLE Customers (
    Customer_ID SERIAL PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Phone VARCHAR(15),
    City VARCHAR(50),
    Country VARCHAR(150)
);
```

1) RETRIEVE ALL BOOKS IN THE "FICTION" GENRE:

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

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
22	Multi-layered optimizing migration	Wesley Escobar	Fiction	1908	39.23	78
28	Expanded analyzing portal	Lisa Coffey	Fiction	1941	37.51	79
29	Quality-focused multi-tasking challenge	Katrina Underwood	Fiction	1905	31.12	100
31	Implemented encompassing conglomeration	Melissa Taylor	Fiction	2010	21.23	44
39	Optimized national process improvement	Megan Goodwin	Fiction	1978	10.99	42
40	Adaptive didactic interface	Natalie Gonzalez	Fiction	1923	25.97	94

2) FIND BOOKS PUBLISHED AFTER THE YEAR 1950:

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

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19
4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	10.95	16
6	Advanced encompassing implementation	Bryan Morgan	Biography	1985	6.56	2
8	Persistent local encoding	Troy Cox	Science Fiction	2019	48.99	84
9	Optimized interactive challenge	Colin Buckley	Fantasy	1987	14.33	70
10	Ergonomic national hub	Samantha Ruiz	Mystery	2015	24.63	25

3) LIST ALL CUSTOMERS FROM THE CANADA:

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

Customer_ID	Name	Email	Phone	City	Country
38	Nicholas Harris	christine93@perkins.com	1234567928	Davistown	Canada
415	James Ramirez	robert54@hall.com	1234568305	Maxwelltown	Canada
468	David Hart	stokesrebecca@gmail.com	1234568358	Thompsonfurt	Canada

4) SHOW ORDERS PLACED IN NOVEMBER 2023:

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

Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
4	433	343	2023-11-25	7	301.21
19	496	60	2023-11-17	9	316.26
75	291	375	2023-11-30	5	170.75
132	469	333	2023-11-22	7	194.32
137	474	471	2023-11-25	8	363.04
163	207	384	2023-11-23	3	101.76
182	129	293	2023-11-01	7	125.51

5) RETRIEVE THE TOTAL STOCK OF BOOKS AVAILABLE:

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

Total_Stock
25056

6) FIND THE DETAILS OF THE MOST EXPENSIVE BOOK:

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

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
340	Proactive system-worthy orchestration	Robert Scott	Mystery	1907	49.98	88
NULL	NULL	NULL	NULL	NULL	NULL	NULL

7) SHOW ALL CUSTOMERS WHO ORDERED MORE THAN 1 QUANTITY OF A BOOK:

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

Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
1	84	169	2023-05-26	8	188.56
2	137	301	2023-01-23	10	216.60
3	216	261	2024-05-27	6	85.50
4	433	343	2023-11-25	7	301.21
5	14	431	2023-07-26	7	136.36
6	439	119	2024-10-11	5	249.40
7	195	467	2023-10-23	6	82.92

8) RETRIEVE ALL ORDERS WHERE THE TOTAL AMOUNT EXCEEDS $20:

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

Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
1	84	169	2023-05-26	8	188.56
2	137	301	2023-01-23	10	216.60
3	216	261	2024-05-27	6	85.50
4	433	343	2023-11-25	7	301.21
5	14	431	2023-07-26	7	136.36
6	439	119	2024-10-11	5	249.40
7	195	467	2023-10-23	6	82.92

9) LIST ALL GENRES AVAILABLE IN THE BOOKS TABLE:

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

genre
Biography
Fantasy
Non-Fiction
Fiction
Romance
Science Fiction
Mystery

10) FIND THE BOOK WITH THE LOWEST STOCK:

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

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
44	Networked systemic implementation	Ryan Frank	Science Fiction	1965	13.55	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

11) CALCULATE THE TOTAL REVENUE GENERATED FROM ALL ORDERS:

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

Revenue
756628.66

1) RETRIEVE THE TOTAL NUMBER OF BOOKS SOLD FOR EACH GENRE:

```
SELECT
*
FROM
ORDERS;

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

Genre	Total_Books_sold
Biography	285
Fantasy	446
Science Fiction	447
Mystery	504
Romance	439
Non-Fiction	351
Fiction	225

2) FIND THE AVERAGE PRICE OF BOOKS IN THE "FANTASY" GENRE:

```
SELECT
AVG(price) AS Average_Price
FROM
    Books
WHERE
Genre = 'Fantasy' ;
```

Average_Price
25.981690

3) LIST CUSTOMERS WHO HAVE PLACED AT LEAST 2 ORDERS:

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

customer_id	name	ORDER_COUNT
84	Gary Blair	2
137	Steven Miller	2
216	Phillip Allen	2
14	John Wood	2
195	Dominique Turner	3
109	Jacob Kelley	2
94	Mr. David Cox	3

4) FIND THE MOST FREQUENTLY ORDERED BOOK:

```
SELECT
o.Book_id, b.title,
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;
```

Book_id	title	ORDER_COUNT
8	Robust tangible hardware	4

5) SHOW THE TOP 3 MOST EXPENSIVE BOOKS OF 'FANTASY' GENRE :

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

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
240	Stand-alone content-based hub	Lisa Ellis	Fantasy	1957	49.90	41
462	Innovative 3rdgeneration database	Allison Contreras	Fantasy	1988	49.23	62
238	Optimized even-keeled analyzer	Sherri Griffith	Fantasy	1975	48.97	72
NULL	NULL	NULL	NULL	NULL	NULL	NULL

6) 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;
```

author	Total_Books_Sold
Margaret Moore	8
John Davidson	13
Christopher Fuentes	6
Marissa Smith	16
Christopher Dixon	15
Tonya Saunders	21
Larry Hunt	6

21

7) LIST THE CITIES WHERE CUSTOMERS WHO SPENT OVER $30 ARE LOCATED:

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

city	total_amount
Lake Paul	188.56
North Keith	216.60
Kelseyfort	85.50
East David	301.21
Richardsonville	136.36
Ramosstad	249.40
Rogersborough	82.92

8) FIND THE CUSTOMER WHO SPENT THE MOST ON ORDERS:

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

customer_id	name	Total_Spent
457	Kim Turner	1398.90

THANK YOU