SQL PROJECT:

ONLINE BOOK STORE ANALYSIS

Efficient Data Analysis and Insights Using SQL Queries

PROJECT OVERVIEW

Objective:

- Create an SQL-based project to manage and analyze data for an online book store.
- Perform both basic and advanced queries to extract meaningful insights.

Datasets Used:

- **Books:** Contains information about book titles, genres, authors, prices, and stock levels.
- **Customer:** Stores customer details such as name, address, city, and country.
- **Order:** Includes information about customer orders, book quantities, and total order amounts.

Total Queries:

- Basic Queries: 11
- Advanced Queries: 9

Tools Used:

- PostgreSQL

BASIC QUERIES

- Retrieve all books in the "Fiction" genre:

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

	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 conglomeration	Melissa Taylor	Fiction	2010	21.23	44

BASIC QUERIES

- Find books published after the year 1950:

```
SELECT * FROM Books
WHERE Published_year>1950
LIMIT 5;
```

	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 implement…	Bryan Morgan	Biography	1985	6.56	2
5	8	Persistent local encoding	Troy Cox	Science Fiction	2019	48.99	84

BASIC QUERIES

- List all customers from Canada:

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

	customer_id [PK] integer	name character varying (100)	email character varying (100)	phone character varying (15)	city character varying (50)	country character varying (150)
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.com	1234568358	Thompsonfurt	Canada

BASIC QUERIES

- Show orders placed in November 2023:

```
SELECT * FROM Orders
WHERE order_date BETWEEN '2023-11-01' AND '2023-11-30'
LIMIT 5;
```

	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

BASIC QUERIES

- Retrieve the total stock of books available:

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

	total_stock bigint
1	25056

- Find the details of the most expensive book:

```
SELECT * FROM Books
ORDER BY Price DESC
LIMIT 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 orchestration	Robert Scott	Mystery	1907	49.98	88

BASIC QUERIES

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

```
SELECT * FROM Orders
WHERE quantity>1
LIMIT 5;
```

	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_amount numeric (10,2)
1	1	84	169	2023-05-26	8	188.56
2	2	137	301	2023-01-23	10	216.60
3	3	216	261	2024-05-27	6	85.50
4	4	433	343	2023-11-25	7	301.21
5	5	14	431	2023-07-26	7	136.36

- Calculate the total revenue generated from all orders:

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

	revenue numeric
1	75628.66

BASIC QUERIES

- Retrieve all orders where the total amount exceeds $20:

```
SELECT * FROM Orders
WHERE total_amount>20
LIMIT 5;
```

	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_amount numeric (10,2)
1	1	84	169	2023-05-26	8	188.56
2	2	137	301	2023-01-23	10	216.60
3	3	216	261	2024-05-27	6	85.50
4	4	433	343	2023-11-25	7	301.21
5	5	14	431	2023-07-26	7	136.36

BASIC QUERIES

- List all genres available in the Books table:

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

	genre character varying (50)
1	Romance
2	Biography
3	Mystery
4	Fantasy
5	Fiction
6	Non-Fiction
7	Science Fiction

- Find the book with the lowest stock:

```
SELECT * FROM Books
ORDER BY stock
LIMIT 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 implementation	Ryan Frank	Science Fiction	1965	13.55	0

ADVANCED QUERIES

- Retrieve the total number of books sold for each genre:

```
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 character varying (50)	total_books_sold bigint
1	Romance	439
2	Biography	285
3	Mystery	504
4	Fantasy	446
5	Fiction	225
6	Non-Fiction	351
7	Science Fiction	447

- Find the average price of books in the "Fantasy" genre:

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

	average_price numeric
1	25.9816901408450704

ADVANCED QUERIES

- 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
LIMIT 5;
```

	customer_id integer	name character varying (100)	order_count bigint
1	225	Christopher Mccullough	2
2	418	Kiara Blankenship MD	3
3	322	William Cameron	3
4	325	Emily Vargas	4
5	376	Justin Donaldson	2

- 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 integer	title character varying (100)	order_count bigint
1	88	Robust tangible hardware	4

ADVANCED QUERIES

- Show the top 3 most expensive books in the "Fantasy" genre:

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

	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	240	Stand-alone content-based hub	Lisa Ellis	Fantasy	1957	49.90	41
2	462	Innovative 3rdgeneration database	Allison Contreras	Fantasy	1988	49.23	62
3	238	Optimized even-keeled analyzer	Sherri Griffith	Fantasy	1975	48.97	72

ADVANCED QUERIES

- Retrieve the total quantity of books sold by each author:

```
SELECT b.author, b.title, SUM(o.quantity) AS Total_Books_Sold
FROM orders o
JOIN books b ON o.book_id=b.book_id
GROUP BY b.author, b.title
LIMIT 5;
```

	author character varying (100)	title character varying (100)	total_books_sold bigint
1	Samantha Stewart	Synchronized mission-critical process improvement	8
2	Brian Haney	Robust eco-centric capacity	9
3	Jared Cortez	Digitized cohesive emulation	10
4	Lori Taylor	Self-enabling contextually-based frame	4
5	Jared Reyes	Upgradable contextually-based hierarchy	3

ADVANCED QUERIES

- 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
LIMIT 5;
```

	city character varying (50)	total_amount numeric (10,2)
1	Taylorfurt	189.45
2	Leeport	141.39
3	Port Jasonview	149.12
4	Port Aaronstad	145.44
5	Matthewfurt	328.50

ADVANCED QUERIES

- 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 5;
```

	customer_id [PK] integer	name character varying (100)	total_spent numeric
1	457	Kim Turner	1398.90
2	174	Jonathon Strickland	1080.95
3	364	Carrie Perez	1052.27
4	405	Julie Smith	991.00
5	386	Pamela Gordon	986.30

- Calculate the stock remaining after fulfilling all orders:

```
SELECT b.book_id, b.title, b.stock, COALESCE(SUM(o.quantity),0) AS Order_quantity,
    b.stock- COALESCE(SUM(o.quantity),0) AS Remaining_Quantity
FROM books b
LEFT JOIN orders o ON b.book_id=o.book_id
GROUP BY b.book_id ORDER BY b.book_id
LIMIT 5;
```

	book_id [PK] integer	title character varying (100)	stock integer	order_quantity bigint	remaining_quantity bigint
1	1	Configurable modular throughput	100	3	97
2	2	Persevering reciprocal knowledge user	19	0	19
3	3	Streamlined coherent initiative	27	5	22
4	4	Customizable 24hour product	8	0	8
5	5	Adaptive 5thgeneration encoding	16	8	8

CHALLENGES AND SOLUTION

Challenges:

- **Challenge 1:** Handling NULL values when performing joins.
- **Solution:** Used COALESCE() to handle NULL values gracefully.
- **Challenge 2:** Query performance optimization.
- **Solution:** Applied INDEX on frequently queried columns to speed up retrieval.
- **Challenge 3:** Managing date ranges in queries
- **Solution:** Used BETWEEN for easy filtering by date range.

KEY INSIGHTS AND OUTCOMES

- **Revenue Generation:** Identified the total revenue from orders.
- **Top Customers:** Found high-spending customers and their locations.
- **Genre Performance:** Analyzed the total books sold by genre.
- **Stock Management:** Calculated remaining stock after fulfilling orders.
- **Frequently Ordered Books:** Determined the most popular books.

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion:

- The project successfully demonstrated SQL querying capabilities for managing and analyzing online book store data.
- Gained insights into customer behavior, sales trends, and stock management.

Future Enhancements:

- Add more complex queries with subqueries and window functions.
- Implement triggers for automatic stock updates.
- Introduce stored procedures for repetitive operations.

THANK YOU

bhaskarpal.official@gmail.com