ONLINE BOOK STORE ANALYSIS

1. **Introduction :** This report presents an end-to-end SQL-based analytical system built for an online bookstore. The objective is to improve inventory visibility, customer understanding, and sales analysis. The system uses structured tables for Books, Customers, and Orders, along with 20 analytical queries.

2. **Business Problem Statement.**

The online bookstore is struggling with three core issues:

- Poor visibility into inventory health, leading to stockouts of high-demand books and overstocking of low-demand ones.
- Limited understanding of customer purchasing behavior, which makes it difficult to identify loyal customers,
- high-value buyers, or customers who frequently buy specific genres.
- No centralized system to monitor sales performance, revenue generation, frequently ordered books, or author-wise contribution.

To solve these gaps, the bookstore needs a structured SQL-based analytical system that can:

- Track stock levels, identify the lowest-stock items, and calculate remaining stock after fulfilling orders.
- Analyze sales by genre, author, order frequency, and price to understand buying patterns.
- Identify revenue trends, high-spending customers, and orders exceeding certain thresholds.
- Provide insights into high-performing books, most expensive books, and average prices across genres.
- The objective of this project is to build an SQL-driven analytical model that delivers operational clarity,
- inventory optimization, and data-driven decision-making for the online bookstore.

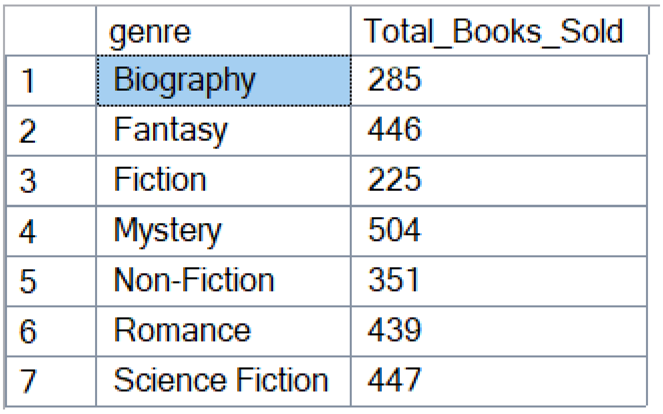
3. **Database Design Tables.**

- Books: Stores book details, genre, price, and stock.
- Customers: Holds customer demographic information.
- Orders: Tracks purchases, quantities, amounts, and dates

4. Key Insights Generated using sql queries

- **Genre-wise books sold**

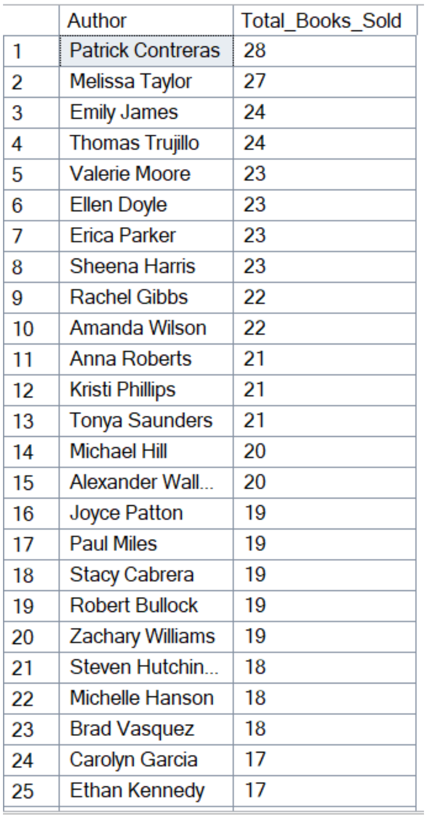
```
select b.genre , sum(o.quantity) Total_Books_Sold
from orders o join Books b
   on o.Book_ID = b.Book_ID
group by b.Genre
```



	genre	Total_Books_Sold
1	Biography	285
2	Fantasy	446
3	Fiction	225
4	Mystery	504
5	Non-Fiction	351
6	Romance	439
7	Science Fiction	447

- **Author-wise sales**

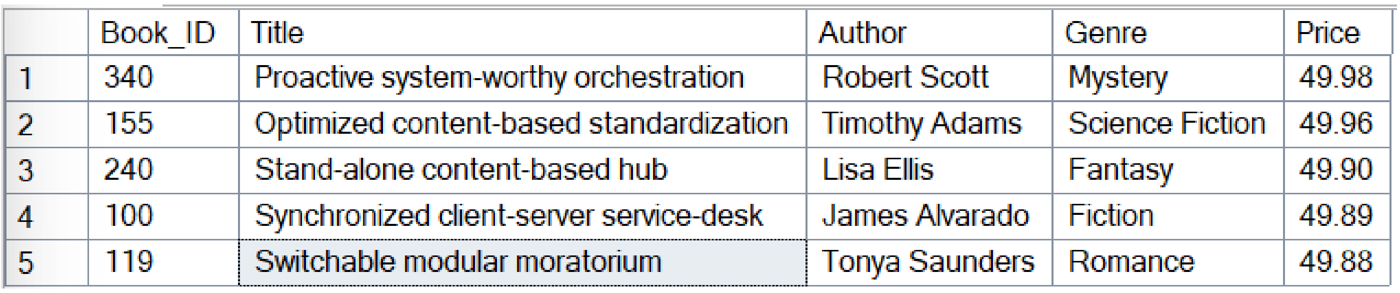
```
SELECT
     b.Author,
     SUM(o.Quantity) AS Total_Books_Sold
FROM Orders o
JOIN Books b ON o.Book_ID = b.Book_ID
GROUP BY b.Author
ORDER BY Total_Books_Sold DESC;
```



	Author	Total_Books_Sold
1	Patrick Contreras	28
2	Melissa Taylor	27
3	Emily James	24
4	Thomas Trujillo	24
5	Valerie Moore	23
6	Ellen Doyle	23
7	Erica Parker	23
8	Sheena Harris	23
9	Rachel Gibbs	22
10	Amanda Wilson	22
11	Anna Roberts	21
12	Kristi Phillips	21
13	Tonya Saunders	21
14	Michael Hill	20
15	Alexander Wall...	20
16	Joyce Patton	19
17	Paul Miles	19
18	Stacy Cabrera	19
19	Robert Bullock	19
20	Zachary Williams	19
21	Steven Hutchin...	18
22	Michelle Hanson	18
23	Brad Vasquez	18
24	Carolyn Garcia	17
25	Ethan Kennedy	17

- **Most Expensive Books:**

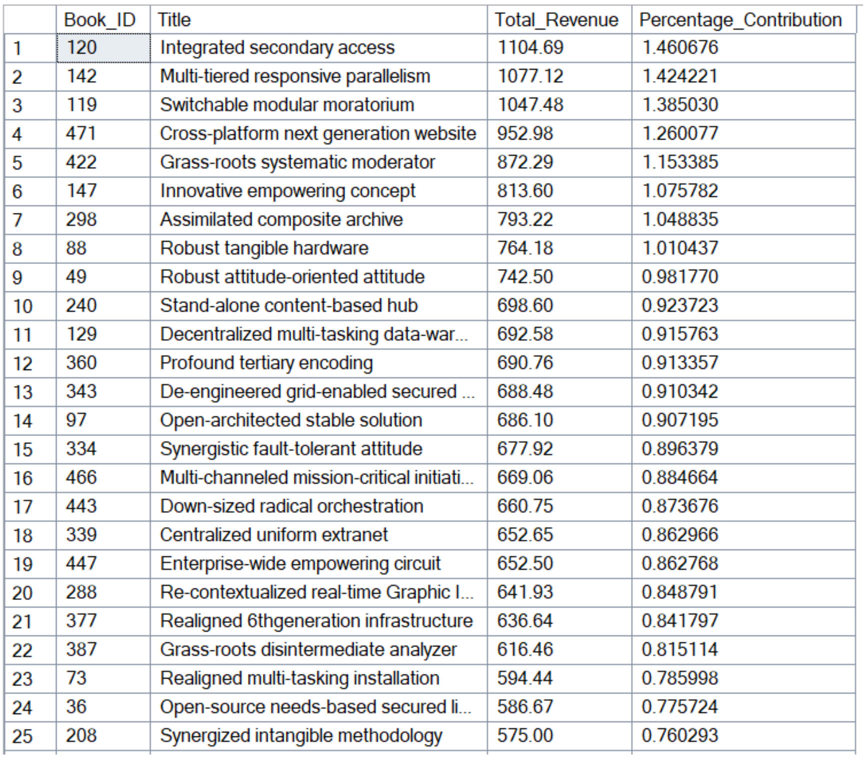
```
SELECT top 5
    Book_ID,
    Title,
    Author,
    Genre,
    Price
FROM Books
ORDER BY Price DESC
```



	Book_ID	Title	Author	Genre	Price
1	340	Proactive system-worthy orchestration	Robert Scott	Mystery	49.98
2	155	Optimized content-based standardization	Timothy Adams	Science Fiction	49.96
3	240	Stand-alone content-based hub	Lisa Ellis	Fantasy	49.90
4	100	Synchronized client-server service-desk	James Alvarado	Fiction	49.89
5	119	Switchable modular moratorium	Tonya Saunders	Romance	49.88

- **Revenue Contribution by Each Book**

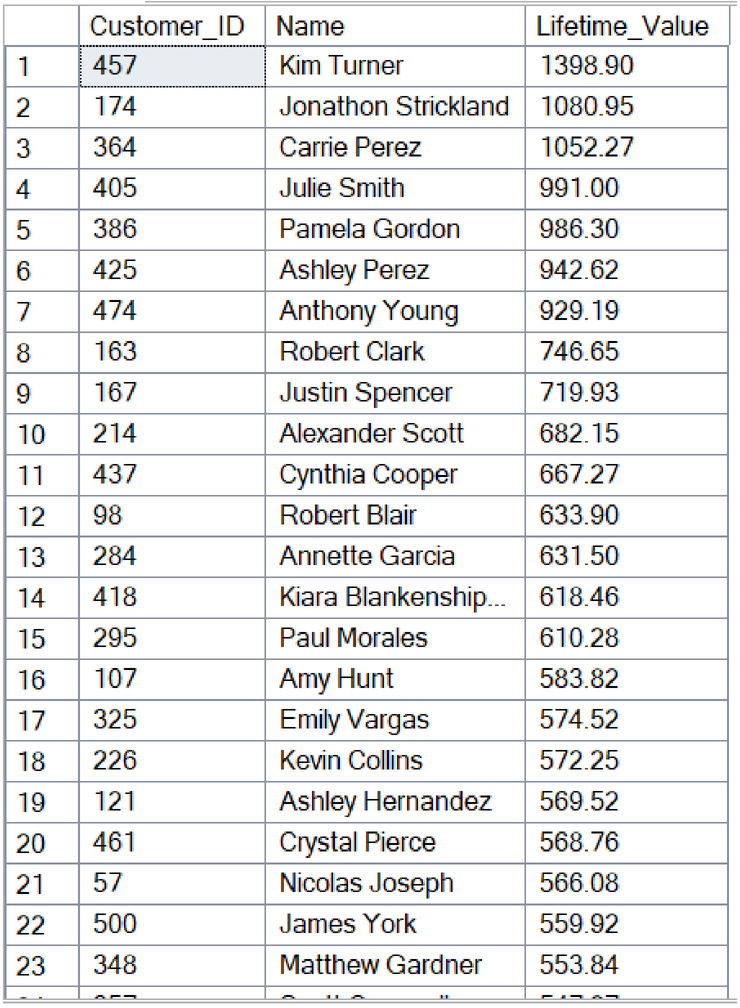
```
SELECT
    b.Book_ID,
    b.Title,
    SUM(o.Total_Amount) AS Total_Revenue,
    SUM(o.Total_Amount) * 100.0 / SUM(SUM(o.Total_Amount)) OVER
() AS Percentage_Contribution
FROM Orders o
JOIN Books b ON o.Book_ID = b.Book_ID
GROUP BY b.Book_ID, b.Title
ORDER BY Total_Revenue DESC;
```



	Book_ID	Title	Total_Revenue	Percentage_Contribution
1	120	Integrated secondary access	1104.69	1.460676
2	142	Multi-tiered responsive parallelism	1077.12	1.424221
3	119	Switchable modular moratorium	1047.48	1.385030
4	471	Cross-platform next generation website	952.98	1.260077
5	422	Grass-roots systematic moderator	872.29	1.153385
6	147	Innovative empowering concept	813.60	1.075782
7	298	Assimilated composite archive	793.22	1.048835
8	88	Robust tangible hardware	764.18	1.010437
9	49	Robust attitude-oriented attitude	742.50	0.981770
10	240	Stand-alone content-based hub	698.60	0.923723
11	129	Decentralized multi-tasking data-war...	692.58	0.915763
12	360	Profound tertiary encoding	690.76	0.913357
13	343	De-engineered grid-enabled secured ...	688.48	0.910342
14	97	Open-architected stable solution	686.10	0.907195
15	334	Synergistic fault-tolerant attitude	677.92	0.896379
16	466	Multi-channeled mission-critical initiati...	669.06	0.884664
17	443	Down-sized radical orchestration	660.75	0.873676
18	339	Centralized uniform extranet	652.65	0.862966
19	447	Enterprise-wide empowering circuit	652.50	0.862768
20	288	Re-contextualized real-time Graphic I...	641.93	0.848791
21	377	Realigned 6thgeneration infrastructure	636.64	0.841797
22	387	Grass-roots disintermediate analyzer	616.46	0.815114
23	73	Realigned multi-tasking installation	594.44	0.785998
24	36	Open-source needs-based secured li...	586.67	0.775724
25	208	Synergized intangible methodology	575.00	0.760293

- **Customer Lifetime Value**

```
SELECT
    c.Customer_ID,
    c.Name,
    SUM(o.Total_Amount) AS Lifetime_Value
FROM Orders o
JOIN Customers c ON o.Customer_ID = c.Customer_ID
GROUP BY c.Customer_ID, c.Name
ORDER BY Lifetime_Value DESC;
```

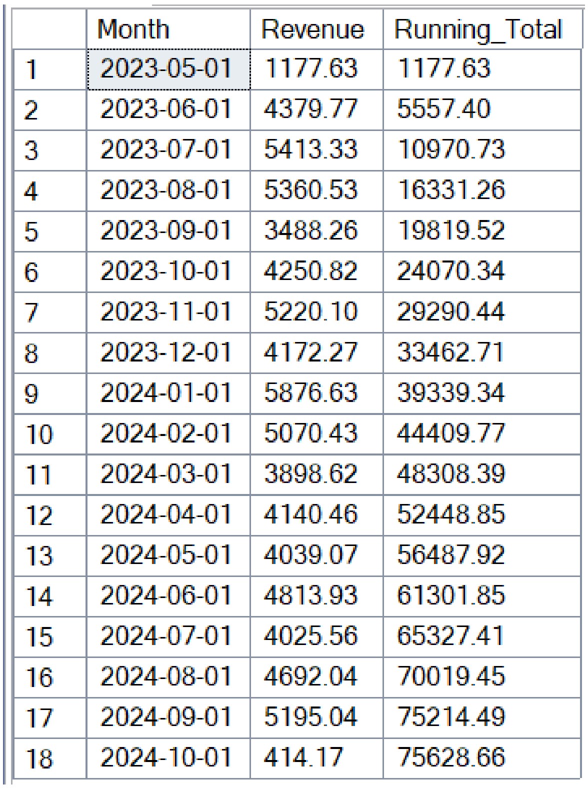


	Customer_ID	Name	Lifetime_Value
1	457	Kim Turner	1398.90
2	174	Jonathon Strickland	1080.95
3	364	Carrie Perez	1052.27
4	405	Julie Smith	991.00
5	386	Pamela Gordon	986.30
6	425	Ashley Perez	942.62
7	474	Anthony Young	929.19
8	163	Robert Clark	746.65
9	167	Justin Spencer	719.93
10	214	Alexander Scott	682.15
11	437	Cynthia Cooper	667.27
12	98	Robert Blair	633.90
13	284	Annette Garcia	631.50
14	418	Kiara Blankenship...	618.46
15	295	Paul Morales	610.28
16	107	Amy Hunt	583.82
17	325	Emily Vargas	574.52
18	226	Kevin Collins	572.25
19	121	Ashley Hernandez	569.52
20	461	Crystal Pierce	568.76
21	57	Nicolas Joseph	566.08
22	500	James York	559.92
23	348	Matthew Gardner	553.84

- **Monthly Revenue + Running Total**

```
WITH MonthlyRevenue AS (
    SELECT
        DATEFROMPARTS(YEAR(Order_Date), MONTH(Order_Date), 1) AS
Month,
        SUM(Total_Amount) AS Revenue
    FROM Orders
    GROUP BY YEAR(Order_Date), MONTH(Order_Date)
)
SELECT
    Month,
    Revenue,
    SUM(Revenue) OVER (ORDER BY Month
                        ROWS BETWEEN UNBOUNDED PRECEDING AND
CURRENT ROW) AS Running_Total
```

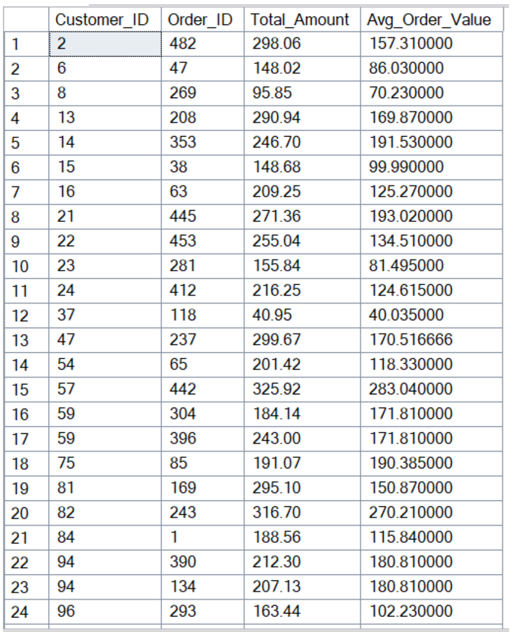
```
FROM MonthlyRevenue
ORDER BY Month;
```



	Month	Revenue	Running_Total
1	2023-05-01	1177.63	1177.63
2	2023-06-01	4379.77	5557.40
3	2023-07-01	5413.33	10970.73
4	2023-08-01	5360.53	16331.26
5	2023-09-01	3488.26	19819.52
6	2023-10-01	4250.82	24070.34
7	2023-11-01	5220.10	29290.44
8	2023-12-01	4172.27	33462.71
9	2024-01-01	5876.63	39339.34
10	2024-02-01	5070.43	44409.77
11	2024-03-01	3898.62	48308.39
12	2024-04-01	4140.46	52448.85
13	2024-05-01	4039.07	56487.92
14	2024-06-01	4813.93	61301.85
15	2024-07-01	4025.56	65327.41
16	2024-08-01	4692.04	70019.45
17	2024-09-01	5195.04	75214.49
18	2024-10-01	414.17	75628.66

- **Customers With Above-Average Spending (Compared to Their Own Avg)**

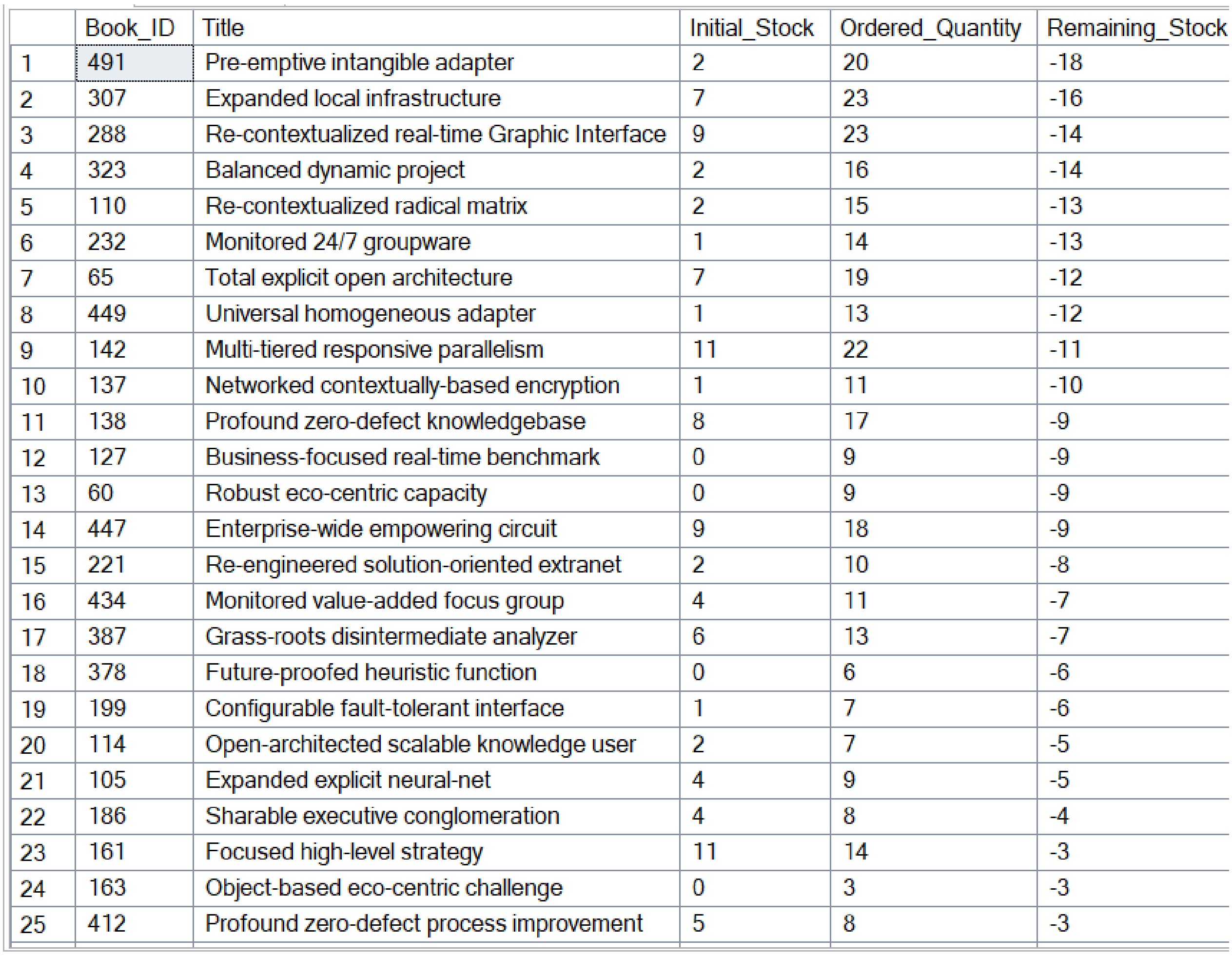
```
WITH OrderWithAvg AS (
    SELECT
        Customer_ID,
        Order_ID,
        Total_Amount,
        AVG(Total_Amount) OVER (PARTITION BY Customer_ID) AS
Avg_Order_Value
    FROM Orders
)
SELECT *
FROM OrderWithAvg
WHERE Total_Amount > Avg_Order_Value;
```



	Customer_ID	Order_ID	Total_Amount	Avg_Order_Value
1	2	482	298.06	157.310000
2	6	47	148.02	86.030000
3	8	269	95.85	70.230000
4	13	208	290.94	169.870000
5	14	353	246.70	191.530000
6	15	38	148.68	99.990000
7	16	63	209.25	125.270000
8	21	445	271.36	193.020000
9	22	453	255.04	134.510000
10	23	281	155.84	81.495000
11	24	412	216.25	124.615000
12	37	118	40.95	40.035000
13	47	237	299.67	170.516666
14	54	65	201.42	118.330000
15	57	442	325.92	283.040000
16	59	304	184.14	171.810000
17	59	396	243.00	171.810000
18	75	85	191.07	190.385000
19	81	169	295.10	150.870000
20	82	243	316.70	270.210000
21	84	1	188.56	115.840000
22	94	390	212.30	180.810000
23	94	134	207.13	180.810000
24	96	293	163.44	102.230000

- **Remaining Stock After Fulfilling Orders.**

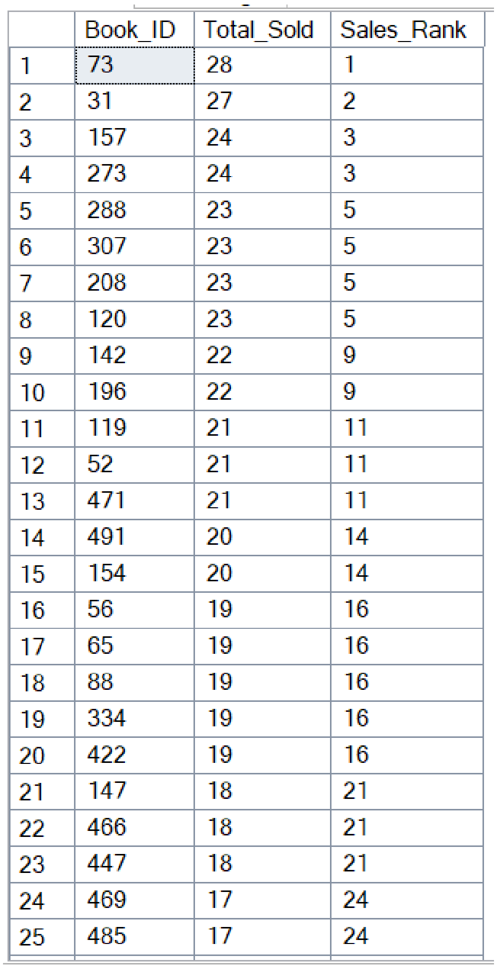
```
SELECT
    b.Book_ID,
    b.Title,
    b.Stock AS Initial_Stock,
    COALESCE(SUM(o.Quantity), 0) AS Ordered_Quantity,
    b.Stock - COALESCE(SUM(o.Quantity), 0) AS Remaining_Stock
FROM Books b
LEFT JOIN Orders o ON b.Book_ID = o.Book_ID
GROUP BY b.Book_ID, b.Title, b.Stock
ORDER BY Remaining_Stock ASC;
```



	Book_ID	Title	Initial_Stock	Ordered_Quantity	Remaining_Stock
1	491	Pre-emptive intangible adapter	2	20	-18
2	307	Expanded local infrastructure	7	23	-16
3	288	Re-contextualized real-time Graphic Interface	9	23	-14
4	323	Balanced dynamic project	2	16	-14
5	110	Re-contextualized radical matrix	2	15	-13
6	232	Monitored 24/7 groupware	1	14	-13
7	65	Total explicit open architecture	7	19	-12
8	449	Universal homogeneous adapter	1	13	-12
9	142	Multi-tiered responsive parallelism	11	22	-11
10	137	Networked contextually-based encryption	1	11	-10
11	138	Profound zero-defect knowledgebase	8	17	-9
12	127	Business-focused real-time benchmark	0	9	-9
13	60	Robust eco-centric capacity	0	9	-9
14	447	Enterprise-wide empowering circuit	9	18	-9
15	221	Re-engineered solution-oriented extranet	2	10	-8
16	434	Monitored value-added focus group	4	11	-7
17	387	Grass-roots disintermediate analyzer	6	13	-7
18	378	Future-proofed heuristic function	0	6	-6
19	199	Configurable fault-tolerant interface	1	7	-6
20	114	Open-architected scalable knowledge user	2	7	-5
21	105	Expanded explicit neural-net	4	9	-5
22	186	Sharable executive conglomeration	4	8	-4
23	161	Focused high-level strategy	11	14	-3
24	163	Object-based eco-centric challenge	0	3	-3
25	412	Profound zero-defect process improvement	5	8	-3

- **Best-Selling Books Ranking**

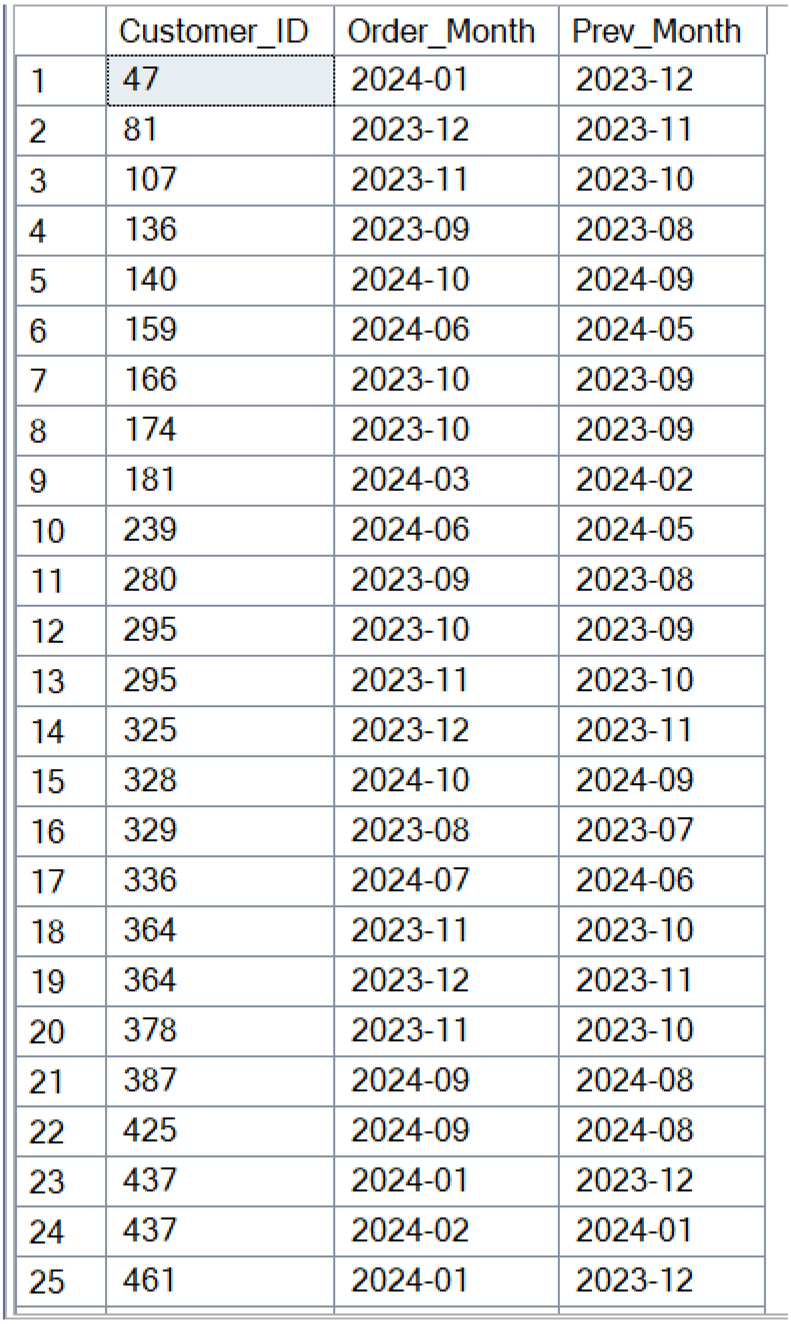
```
WITH BookSales AS (
    SELECT
        Book_ID,
        SUM(Quantity) AS Total_Sold,
        RANK() OVER (ORDER BY SUM(Quantity) DESC) AS Sales_Rank
    FROM Orders
    GROUP BY Book_ID
)
SELECT *
FROM BookSales
ORDER BY Sales_Rank
```



	Book_ID	Total_Sold	Sales_Rank
1	73	28	1
2	31	27	2
3	157	24	3
4	273	24	3
5	288	23	5
6	307	23	5
7	208	23	5
8	120	23	5
9	142	22	9
10	196	22	9
11	119	21	11
12	52	21	11
13	471	21	11
14	491	20	14
15	154	20	14
16	56	19	16
17	65	19	16
18	88	19	16
19	334	19	16
20	422	19	16
21	147	18	21
22	466	18	21
23	447	18	21
24	469	17	24
25	485	17	24

- **Customers With Consecutive Monthly Purchases**

```
WITH MonthlyOrders AS (
    SELECT
        Customer_ID,
        FORMAT(Order_Date, 'yyyy-MM') AS Order_Month,
        LAG(FORMAT(Order_Date, 'yyyy-MM')) OVER (PARTITION BY
Customer_ID ORDER BY Order_Date) AS Prev_Month
    FROM Orders
)
SELECT *
FROM MonthlyOrders
WHERE DATEADD(MONTH, 1, CAST(Prev_Month + '-01' AS DATE))
      = CAST(Order_Month + '-01' AS DATE);
```



	Customer_ID	Order_Month	Prev_Month
1	47	2024-01	2023-12
2	81	2023-12	2023-11
3	107	2023-11	2023-10
4	136	2023-09	2023-08
5	140	2024-10	2024-09
6	159	2024-06	2024-05
7	166	2023-10	2023-09
8	174	2023-10	2023-09
9	181	2024-03	2024-02
10	239	2024-06	2024-05
11	280	2023-09	2023-08
12	295	2023-10	2023-09
13	295	2023-11	2023-10
14	325	2023-12	2023-11
15	328	2024-10	2024-09
16	329	2023-08	2023-07
17	336	2024-07	2024-06
18	364	2023-11	2023-10
19	364	2023-12	2023-11
20	378	2023-11	2023-10
21	387	2024-09	2024-08
22	425	2024-09	2024-08
23	437	2024-01	2023-12
24	437	2024-02	2024-01
25	461	2024-01	2023-12

business recommendations :

- Stop overstocking slow-moving genres and reallocate budget to genres with consistently higher sales.
- Push promotions for high-margin authors/books instead of wasting effort on cheap low-ROI titles.
- Reprice underperforming expensive books or bundle them — dead inventory is burning cash.
- Prioritize books with the highest revenue contribution when planning marketing and discounts.

- Target customers with high lifetime value with loyalty perks; don’t waste budget on low-spending segments.
- Run campaigns at the start of each month since monthly revenue patterns show predictable peaks.
- Reduce churn by analyzing customers who spend below average and give them targeted nudges.
- Restock only the books that actually sell — the remaining stock report tells you where you're losing money.
- Double down on the top-selling books — keep them always in stock and promote them aggressively.
- Reward customers with consecutive monthly purchases to lock them in before competitors do.