



# **Bookstore Management & SQL Analytics**

**Project context:**

The Bookstore Management project is designed to simulate a real-world retail bookstore environment and demonstrate how SQL analytics can provide valuable business insights. The analysis is based on a fictional bookstore dataset, with the goal of managing inventory, tracking customer behavior, and monitoring sales performance. By examining customer locations, order trends, and revenue patterns, this project highlights how structured SQL queries can support decision-making in retail operations.

**Tables used in analysis: Books, Orders, and Customers.**

Through SQL queries, key business-critical questions such as revenue tracking, customer segmentation, genre performance, and stock availability are answered, providing actionable insights to improve bookstore management and enhance customer satisfaction.

**Tables Used:**

The analysis is built upon a simulated relational database consisting of three tables. These tables provide the raw materials for the analysis, representing the fundamental components of the retail operation.

**• Books**

- Book Id
- Title
- Author
- Genre
- Published\_Year
- Price
- Stock

**• Orders**

- Order\_Id
- Customer\_ID
- Book\_ID
- Order\_Date
- Quantity
- Total\_Amount

**• Customers**

- Customer\_Id
- name
- Email
- Phone
- City
- Country

## Question 1

**Problem statement:** Retrieve all books in the "Fiction" genre.

**Analysis approach:** Identify books classified as *Fiction* for targeted promotion and genre segmentation.

### Identifying relevant tables and columns:

- Table: Books
- Columns: book\_id, title, author, genre, published\_year, price, stock

### Calculation logic:

- WHERE filter → Genre = 'Fiction'

### SQL query:

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

### Output:

book_id	title	author	genre	published_year	price	stock
4	Customizable 24hour product	Christopher Andrews	Fiction	2020	435	8
22	Multi-layered optimizing migration	Wesley Escobar	Fiction	1908	392	78
28	Expanded analyzing portal	Lisa Coffey	Fiction	1941	375	79

### Recommendations:

1. **Targeted promotions** – Run seasonal or thematic campaigns for Fiction books to boost sales.
2. **Author partnerships** – Partner with top-performing Fiction authors for special releases.
3. **Stock management** – Monitor Fiction inventory closely as it is a high-demand genre.

## Question 2

**Problem statement:** Find books published after 1950.

**Analysis approach:** Identify modern titles for marketing campaigns and highlight newer literature.

**Identifying relevant tables and columns:**

- Table: Books
- Columns: book\_id, title, author, published\_year

**Calculation logic:**

- WHERE filter → Published\_year > 1950

**Sorting the Results:**

- ORDER BY published\_year ASC/DESC (optional, for chronology).

**SQL query:**

SELECT \* FROM Books WHERE Published\_year > 1950;

**Output:**

book_id	title	author	genre	published_year	price	stock
2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	358	19
4	Customizable 24hour product	Christopher Andrews	Fiction	2020	435	8
5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	109	16

**Recommendations:**

1. **New arrivals section** – Highlight books published after 1950 as “modern literature.”
2. **Pricing strategy** – Premium pricing can be applied to recent high-demand titles.
3. **Marketing campaigns** – Feature post-1950 books in *digital ads* for younger readers.

## Question 3

**Problem statement:** List customers from Canada.

**Analysis approach:** Segment customers based on country for regional campaigns.

**Identifying relevant tables and columns:**

- Table: Customers
- Columns: customer\_id, name, city, country

**Calculation logic:**

- WHERE filter → country = 'Canada'

**Sorting the Results:**

- ORDER BY name ASC for better readability.

**SQL query:**

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

**Output:**

customer_id	name	email	phone
38	Nicholas Harris	christine93@perkins.com	1234567928
415	James Ramirez	robert54@hall.com	1234568305
468	David Hart	stokesrebecca@gmail.com	1234568358

**Recommendations:**

1. **Regional discounts** – Offer country-specific discounts to Canadian readers.
2. **Localized shipping** – Negotiate better delivery logistics in Canada.
3. **Community engagement** – Collaborate with Canadian book clubs for higher outreach.

**Question 4**

**Problem statement:** Retrieve orders placed in November 2023.

**Analysis approach:** Analyze seasonal demand trends and peak shopping periods.

**Identifying relevant tables and columns:**

- Table: Orders Columns: order\_id, book\_id, customer\_id, quantity, total\_amount, order\_date

**Calculation logic:**

- WHERE filter → order\_date BETWEEN '01-11-2023' AND '30-11-2023'

**Sorting the Results:**

- ORDER BY order\_date ASC to show transactions in order.

### SQL query:

SELECT \* FROM Orders WHERE order\_date BETWEEN '01-11-2023' AND '30-11-2023';

### Output:

order_id	customer_id	book_id	order_date	quantity	total_amount
1	84	169	26-05-2023	8	1880
2	137	301	23-01-2023	10	2160
3	216	261	27-05-2024	6	852

### Recommendations:

1. **Promote holiday campaigns** – Peak activity in November suggests strong seasonal sales.
2. **Stock planning** – Increase inventory in late October to meet November demand.
3. **Marketing tie-ins** – Align with Black Friday/Cyber Monday offers.

## Question 5

**Problem statement:** Find total stock available.

**Analysis approach:** Summarize current book inventory for warehouse planning.

### Identifying relevant tables and columns:

- Table: Books
- Columns: stock

### Calculation logic:

- Aggregation → SUM(stock)

### SQL query:

SELECT SUM(stock) AS Total\_Stock FROM Books;

### Output:

Total_Stock
25,056

### Recommendations:

1. **Warehouse allocation** – Use total stock to optimize space and reduce storage costs.
2. **Procurement planning** – Track reorder points for books running low.
3. **Sales forecasting** – Link stock levels with seasonal demand.

## Question 6

**Problem statement:** Identify the most expensive book.

**Analysis approach:** Determine high-value items that may require premium marketing.

**Identifying relevant tables and columns:**

- Table: Books
- Columns: title, price, author

**Calculation logic:**

- ORDER BY price DESC
- LIMIT 1

**Sorting the Results:**

- ORDER BY price DESC LIMIT 1 → returns top-priced item.

**SQL query:**

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

**Output:**

book_id	title	author	genre	published_year	price	stock
346	Object-based asynchronous moratorium	Jennifer Oliver	Fantasy	1965	997	23

**Recommendations:**

1. **Premium marketing** – Promote as a “collector’s edition” or “exclusive release.”
2. **Bundling strategy** – Pair with mid-priced books to increase sales.
3. **Monitor pricing** – Ensure competitive positioning against similar high-end titles.

## Question 7

**Problem statement:** Retrieve customers who ordered more than 1 quantity in a single order.

**Analysis approach:** Identify loyal or bulk-buying customers.

**Identifying relevant tables and columns:**

- Table: Orders
- Columns: order\_id, customer\_id, quantity

**Calculation logic:**

- WHERE filter → quantity > 1

**Sorting the Results:**

- ORDER BY quantity DESC to show largest bulk buyers first.

**SQL query:**

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

**Output:**

order_id	customer_id	book_id	order_date	quantity	total_amount
1	84	169	26-05-2023	8	1880
2	137	301	23-01-2023	10	2160
3	216	261	27-05-2024	6	852
4	433	343	25-11-2023	7	3010
5	14	431	26-07-2023	7	1358

**Recommendations:**

1. **Loyalty programs** – Reward customers who consistently buy in bulk.
2. **Cross-selling** – Suggest complementary books during checkout.
3. **Discount tiers** – Offer “Buy 2, Get 1 Free” schemes to encourage larger orders.



## Question 8

**Problem statement:** Retrieve orders where total amount > 20.

**Analysis approach:** Identify higher-value transactions to analyze spending patterns.

**Identifying relevant tables and columns:**

- Table: Orders
- Columns: order\_id, book\_id, customer\_id, quantity, total\_amount, order\_date

**Calculation logic:**

- WHERE filter → total\_amount > 20

**Sorting the Results:**

- ORDER BY total\_amount DESC to highlight biggest spenders first.

**SQL query:**

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

**Output:**

order_id	customer_id	book_id	order_date	quantity	total_amount
1	84	169	26-05-2023	8	1880
2	137	301	23-01-2023	10	2160
3	216	261	27-05-2024	6	852
4	433	343	25-11-2023	7	3010
5	14	431	26-07-2023	7	1358
6	439	119	11-10-2024	5	2490
7	195	467	23-10-2023	6	828

**Recommendations:**

1. **VIP program** – Identify and reward high-spending customers with loyalty perks.
2. **Personalized marketing** – Send targeted offers to frequent high-value buyers.
3. **Fraud monitoring** – Flag unusually high transactions for validation.

## Question 9

**Problem statement:** List all unique book genres.

**Analysis approach:** Understand genre diversity and support analytics dashboards/filters.

**Identifying relevant tables and columns:**

- Table: Books
- Columns: genre

**Calculation logic:**

- DISTINCT function on genre

**Sorting the Results:**

- ORDER BY genre ASC for clean alphabetical listing.

**SQL query:**

SELECT DISTINCT genre FROM Books;

**Output:**

Genre
Romance
Biography
Mystery
Fantasy
Fiction
Non-Fiction
Science Fiction

**Recommendations:**

1. **Genre filters** – Build genre-based search filters for online bookstore.
2. **Category promotions** – Run “Fantasy Week” or “Mystery Monday” deals.
3. **Demand forecasting** – Track popularity shifts across genres.

**Question 10**

**Problem statement:** Identify the book with the lowest stock.

**Analysis approach:** Find items at risk of going out of stock for timely restocking.

**Identifying relevant tables and columns:**

- Table: Books
- Columns: book\_id, title, stock

**Calculation logic:**

- ORDER BY stock ASC
- LIMIT 1

**Sorting the Results:**

- ORDER BY stock ASC LIMIT 1 → returns lowest-stock item.

**SQL query:**

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

**Output:**

book_id	title	author	genre	published_year	price	stock
44	Networked systemic implementation	Ryan Frank	Science Fiction	1965	135	0

**Recommendations:**

1. **Immediate restock** – Prevent lost sales by replenishing low-stock books.
2. **Stock threshold alerts** – Automate alerts when stock drops below 10.
3. **Demand prioritization** – Prioritize reordering popular low-stock books.

**Question 11**

**Problem statement:** Calculate total revenue.

**Analysis approach:** Measure the financial performance of bookstore sales.

**Identifying relevant tables and columns:**

- Table: Orders
- Columns: total\_amount

**Calculation logic:**

- Aggregation → SUM(total\_amount)

**SQL query:**

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

**Output:**

Revenue
938675

**Recommendations:**

1. **Revenue benchmarking** – Compare performance against previous months/years.
2. **Growth strategy** – Allocate resources to top-performing categories.
3. **Profitability check** – Pair revenue with cost data for margin analysis.

**Question 12**

**Problem statement:** Calculate the number of books sold by genre.

**Analysis approach:** Identify the best-selling genres to guide marketing and inventory strategy.

**Identifying relevant tables and columns:**

- Table: Orders, Books
- Columns: Orders.quantity, Books.genre

**Calculation logic:**

- JOIN Orders and Books on book\_id
- SUM of quantity grouped by genre

**Sorting the Results:**

- ORDER BY Total\_Books\_sold DESC to highlight top genres.

**SQL query:**

```
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 ORDER BY Total_Books_sold DESC;
```

**Output:**

genre	total_books_sold
Romance	439
Biography	285
Mystery	504

Fantasy	446
Fiction	225
Non-Fiction	351
Science Fiction	447

### Recommendations:

1. **Double down on top genres** – Increase inventory and promotions for Fiction & Fantasy.
2. **Curated bundles** – Create cross-genre bundles to promote lower-performing categories.
3. **Seasonal marketing** – Tie genre promotions to holidays (e.g., Fantasy for Halloween).

### Question 13

**Problem statement:** Find the average price of Fantasy books.

**Analysis approach:** Establish pricing benchmarks within the Fantasy category.

### Identifying relevant tables and columns:

- Table: Books
- Columns: price, genre

### Calculation logic:

- WHERE filter → genre = 'Fantasy'
- Aggregate → AVG(price)

### SQL query:

```
SELECT AVG(price) AS Average_Price FROM Books
```

```
WHERE Genre = 'Fantasy';
```

### Output:

Average_Price
334.98

### Recommendations:

1. **Standardize pricing** – Keep Fantasy books within a predictable price range.
2. **Premium tiering** – Introduce higher-priced collector editions.

### 3. **Competitive check** – Compare with other genres to balance price positioning.

#### Question 14

**Problem statement:** Identify customers with at least 2 orders.

**Analysis approach:** Highlight repeat customers for loyalty programs.

**Identifying relevant tables and columns:**

- Table: Orders, Customers
- Columns: Orders.customer\_id, Orders.order\_id, Customers.name

**Calculation logic:**

- JOIN Orders and Customers on customer\_id
- COUNT of orders grouped by customer\_id
- HAVING condition → COUNT(order\_id) >= 2

**Sorting the Results:**

- ORDER BY ORDER\_COUNT DESC

**SQL query:**

```
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
ORDER BY ORDER_COUNT DESC;
```

**Output:**

customer_id	name	order_count
225	Christopher Mccullough	2
418	Kiara Blankenship MD	3
322	William Cameron	3
325	Emily Vargas	4
376	Justin Donaldson	2
486	Melanie Kelly	2
461	Crystal Pierce	3
2	Crystal Clements	2
149	Jason Robinson	3
173	Victoria Dixon	2
120	Rita Wallace	2

### Recommendations:

1. **Loyalty rewards** – Provide coupons or discounts to repeat customers.
2. **Personalized offers** – Recommend books based on previous orders.
3. **Retention tracking** – Monitor order frequency trends for customer churn prevention.

### Question 15

**Problem statement:** Find the most frequently ordered book.

**Analysis approach:** Identify the bookstore's best-seller for focused marketing.

#### Identifying relevant tables and columns:

- Table: Orders, Books
- Columns: Orders.book\_id, Books.title, Orders.order\_id

#### Calculation logic:

- JOIN Orders and Books on book\_id
- COUNT of order\_id grouped by book\_id
- ORDER BY ORDER\_COUNT DESC
- LIMIT 1

#### Sorting the Results:

- ORDER BY ORDER\_COUNT DESC LIMIT 1

#### SQL query:

```
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;
```

#### Output:

book_id	title	order_count
88	Robust tangible hardware	4

### Recommendations:

1. **Best-seller promotion** – Feature in front-page banners and top picks.

2. **Restock priority** – Ensure consistent availability of the best-selling book.
3. **Spin-off opportunities** – Explore sequels or author collaborations.

## Question 16

**Problem statement:** Retrieve the top 3 most expensive Fantasy books.

**Analysis approach:** Identify premium titles in the Fantasy genre for curated collections.

**Identifying relevant tables and columns:**

- Table: Books
- Columns: title, genre, price

**Calculation logic:**

- WHERE filter → genre = 'Fantasy'
- ORDER BY price DESC
- LIMIT 3

**Sorting the Results:**

- ORDER BY price DESC LIMIT 3

**SQL query:**

```
SELECT * FROM Books WHERE genre = 'Fantasy'
```

```
ORDER BY price DESC LIMIT 3;
```

**Output:**

book_id	title	author	genre	published_year	price	stock
346	Object-based asynchronous moratorium	Jennifer Oliver	Fantasy	1965	997	23
243	Automated systemic toolset	Tiffany Conley	Fantasy	1953	887	65
316	Multi-tiered client-server methodology	Rebecca Chavez	Fantasy	1992	881	74

**Recommendations:**

1. **Curated premium section** – Highlight these as “Luxury Fantasy Picks.”
2. **Bundling offers** – Bundle premium Fantasy books with mid-range ones.
3. **Collector editions** – Market these as high-value for niche collectors.



## Question 17

**Problem statement:** Calculate books sold by each author.

**Analysis approach:** Evaluate author performance for partnerships and promotions.

**Identifying relevant tables and columns:**

- Table: Orders, Books
- Columns: Orders.quantity, Books.author

**Calculation logic:**

- JOIN Orders and Books on book\_id
- SUM of quantity grouped by author

**Sorting the Results:**

- ORDER BY Total\_Books\_Sold DESC

**SQL query:**

```
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;
```

**Output:**

author	total_books_sold
Jared Cortez	10
Tracy Parker	11
Taylor Wang	9
Cathy Knight	6
Bianca Matthews	3
Douglas Malone	6
James Alvarado	9
Betty Cross	6
Michael Hill	20

**Recommendations:**

1. **Author promotions** – Partner with top authors for exclusive book launches.
2. **Royalty negotiations** – Use sales data to set fair contracts.
3. **Cross-author bundles** – Package works from multiple authors for marketing.

## Question 18

**Problem statement:** Identify cities where customers spent more than 30.

**Analysis approach:** Detect high-revenue geographic markets.

**Identifying relevant tables and columns:**

- Table: Orders, Customers
- Columns: Orders.total\_amount, Customers.city

**Calculation logic:**

- JOIN Orders and Customers on customer\_id
- WHERE filter → total\_amount > 30
- DISTINCT cities to avoid duplicates

**Sorting the Results:**

- ORDER BY city ASC

**SQL query:**

SELECT DISTINCT c.city, o.total\_amount FROM Orders o

JOIN Customers c ON o.customer\_id = c.customer\_id

WHERE o.total\_amount > 30;

**Output:**

city	total_amount
Gonzalestown	2412
Elizabethshire	2530
West Blake	1260
Lake Brittany	1113
Barbarahaven	223
Brandimouth	399
Rodriguezmouth	5130
North Keith	2160
North Michael	209
Lake Robert	2904
Lake Anthony	2048
Port Josephmouth	1272
Lake Hannahton	978
East Wyattchester	2150
Parkerborough	6972

**Recommendations:**

1. **Regional campaigns** – Focus ads and promotions in high-spending cities.
2. **Local events** – Partner with bookstores in Toronto, Vancouver, Montreal.

### 3. **Delivery optimization** – Improve shipping times in these key markets.

#### Question 19

**Problem statement:** Identify the top customer by total spending.

**Analysis approach:** Recognize high-value customers for retention strategies.

#### Identifying relevant tables and columns:

- Table: Orders, Customers
- Columns: Orders.total\_amount, Customers.name

#### Calculation logic:

- JOIN Orders and Customers on customer\_id
- SUM of total\_amount grouped by customer
- ORDER BY Total\_Spent DESC
- LIMIT 1

#### Sorting the Results:

- ORDER BY Total\_Spent DESC LIMIT 1

#### SQL query:

```
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;
```

#### Output:

customer_id	name	total_spent
425	Ashley Perez	14474

#### Recommendations:

1. **Exclusive rewards** – Provide early access or discounts to top spenders.
2. **Customer testimonials** – Use high-value customers as brand advocates.
3. **Retention focus** – Assign relationship managers for VIP customers.

## Question 20

**Problem statement:** Calculate remaining stock after fulfilling all orders.

**Analysis approach:** Provide real-time inventory insights post-sales.

**Identifying relevant tables and columns:**

- Table: Books, Orders
- Columns: Books.stock, Orders.quantity

**Calculation logic:**

- LEFT JOIN Books and Orders
- COALESCE for NULL values (books without orders)
- Remaining stock = Books.stock - SUM(Orders.quantity)

**Sorting the Results:**

- ORDER BY book\_id ASC

**SQL query:**

```
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, b.title, b.stock ORDER BY b.book_id;
```

**Output:**

book_id	title	stock	order_quantity	remaining_quantity
1	Configurable modular throughput	100	3	97
2	Persevering reciprocal knowledge user	19	0	19
3	Streamlined coherent initiative	27	5	22
4	Customizable 24hour product	8	0	8
5	Adaptive 5thgeneration encoding	16	8	8
6	Advanced encompassing implementation	2	0	2
7	Open-architected exuding structure	95	5	90
8	Persistent local encoding	84	3	81
9	Optimized interactive challenge	70	0	70
10	Ergonomic national hub	25	1	24
11	Secured zero tolerance time-frame	10	5	5
12	Polarized optimal array	63	0	63
13	Adaptive 5thgeneration orchestration	99	9	90

14	Re-engineered demand-driven parallelism	95	0	95
15	User-friendly motivating strategy	58	0	58

### Recommendations:

1. **Automated stock alerts** – Trigger alerts when remaining stock < 5.
2. **Dynamic reordering** – Use sales trends to automate procurement.
3. **Safety stock buffer** – Maintain extra units of high-demand books.