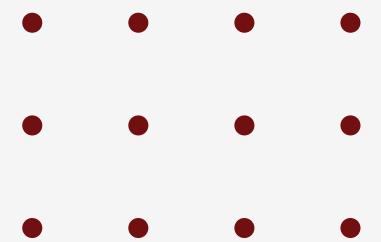


Bookstore Analytics & Insights



-chitra singh

Overview



Bookstore analytics uses data to understand customer preferences, product trends, inventory health, and marketing performance. SQL helps answer questions like:

- **Which books sell the most?**
- **Which customers generate the most profit?**



Project Goals:



Identify best-selling books and revenue drivers



Monitor and alert on low stock inventory



Segment customers by behavior using RFM metrics



Measure ROI on marketing and customer acquisition



Present actionable insights to stakeholders

- • •
- • •
- • •



SQL (MySQL / PostgreSQL / SQLite)



**Joins, Subqueries,
Aggregations**



**Common Table
Expressions (CTEs)**

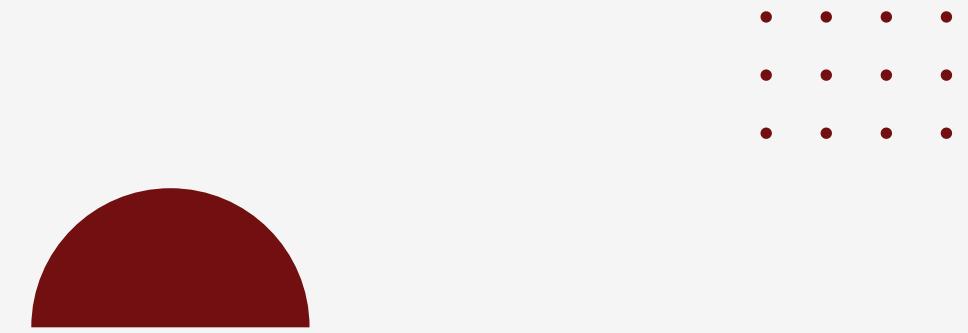


Tools & Techniques:



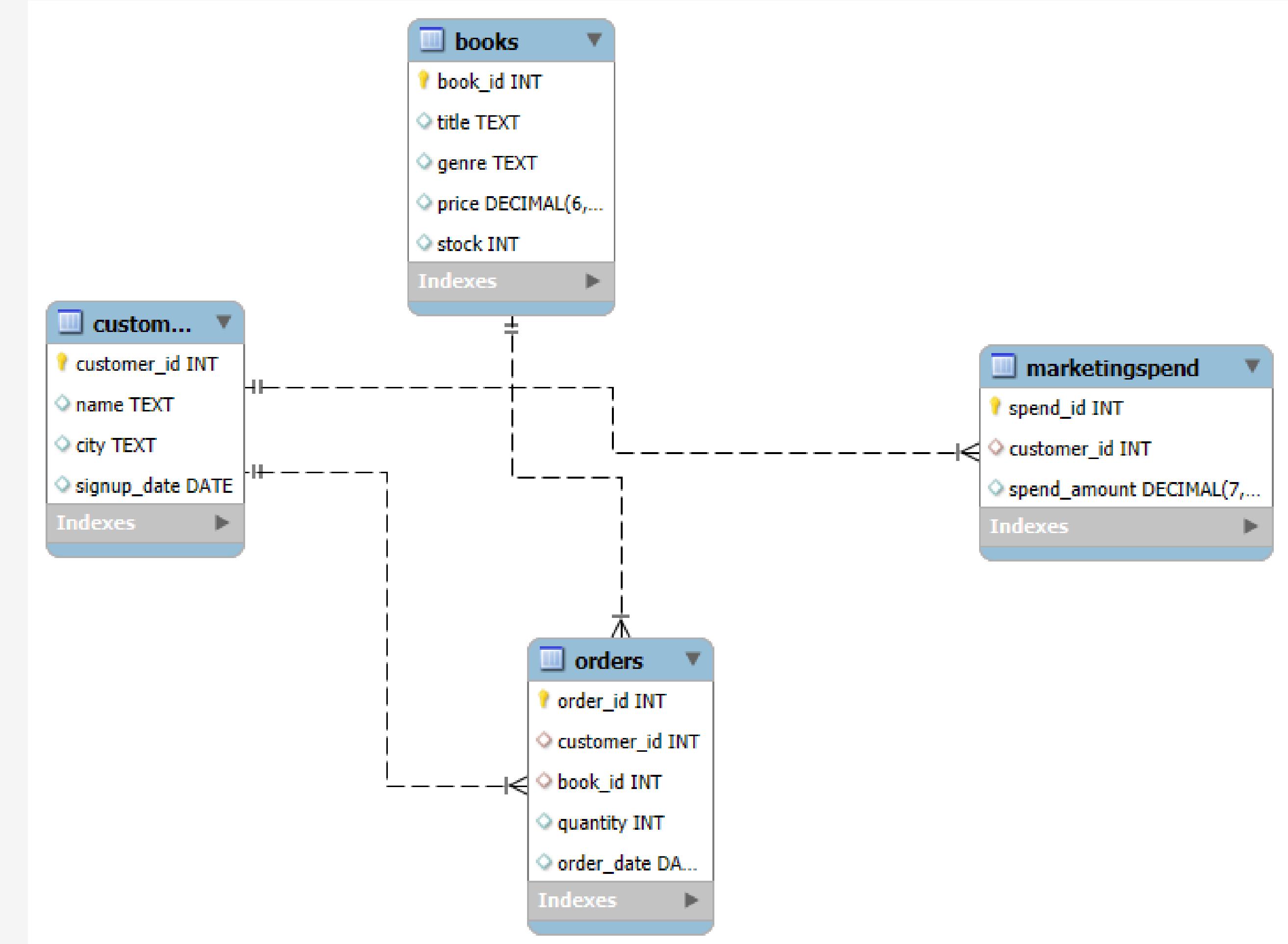
Table

	Description
Books	Book details and stock
Customers	Customer info including city & signup date
Orders	Purchase data linking customers and books
MarketingSpend	Marketing cost per customer



Dataset Overview:

EER diagram



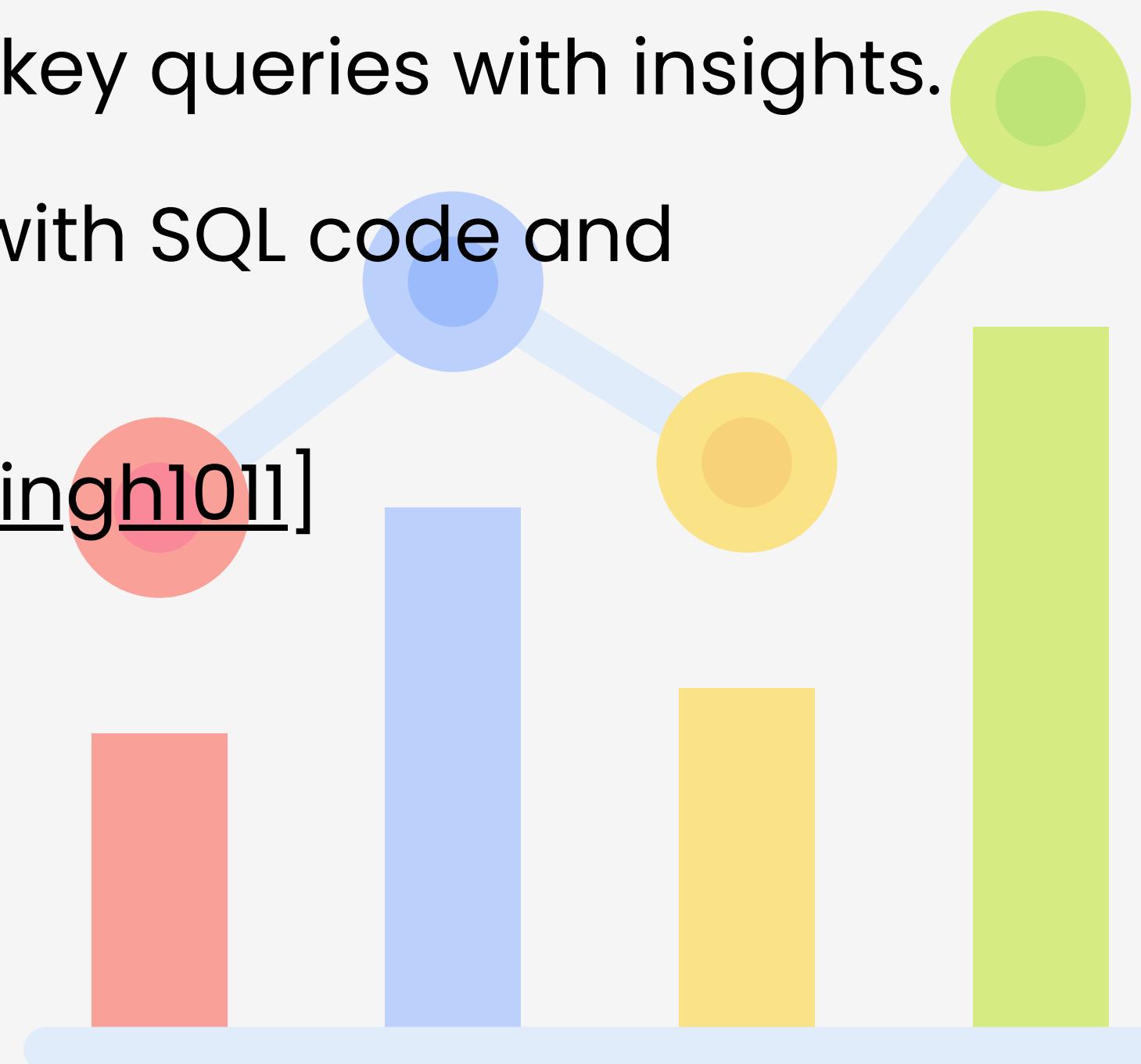
Analytical SQL Queries

- This project contains 24 analytical queries answering key bookstore business questions
- In this presentation, we are showcasing 14 key queries with insights.
- To explore the complete set of 24 queries with SQL code and outputs, please visit my GitHub repository.



GitHub Link: [github.com/chitrasingh1011]

• • •
• • •
• • •
• • •



Total number of books available in the Books table

```
30 •   SELECT COUNT(book_id)  
31      FROM books;  
32
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	
COUNT(book_id)					
▶	10				

The book with the highest price

```
79 •   SELECT title, price  
80      FROM books  
81      ORDER BY price DESC  
82      LIMIT 1;
```

Result Grid		Filter Rows:	Export:
title	price		
▶ AI for Beginners	45.00		



Titles of all books along with their stock, but only those books where the stock is less than 15

```
38 •   SELECT  
39       title, stock  
40   FROM Books  
41 WHERE stock < 15;
```

result Grid | Filter Rows:

title	stock
Fantasy World Chronides	10
Space Explorers	12



Average order value across all orders.

```
21 •   SELECT  
22       ROUND(SUM(quantity * b.price) * 1.0 / COUNT(DISTINCT o.order_id), 2) AS avg_order_value  
23   FROM Orders o  
24   JOIN Books b ON o.book_id = b.book_id;  
25
```

result Grid | Filter Rows: Export: Wrap Cell Content:

avg_order_value
45.50

Top 10 pairs of books that are most frequently bought together by the same customer. Show the book IDs and the number of times they were purchased together

SELECT

```
    o1.book_id AS book_1,  
    o2.book_id AS book_2,  
    COUNT(*) AS times_bought_together
```

FROM Orders o1

JOIN Orders o2

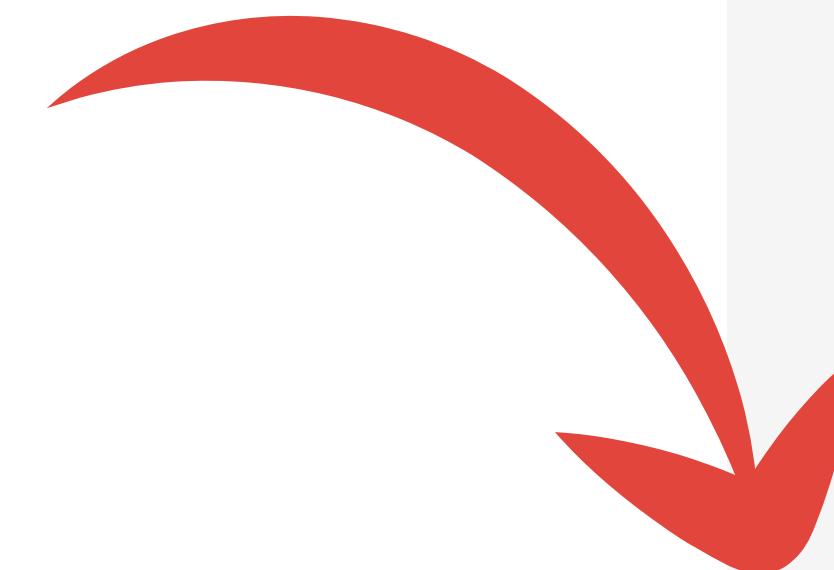
ON o1.customer_id = o2.customer_id AND o1.order_id != o2.order_id

WHERE o1.book_id < o2.book_id

GROUP BY book_1, book_2

ORDER BY times_bought_together DESC

LIMIT 10;



book_1	book_2	times_bought_together
1	3	2
2	4	1
6	7	1
7	9	1
2	6	1
2	7	1
4	7	1
4	9	1
5	8	1
1	9	1

Each book title along with the total units sold and the total revenue earned from that book, and order the result by revenue (highest first)

```
87 • SELECT
88     b.title,
89     SUM(o.quantity) AS total_units_sold,
90     SUM(o.quantity * b.price) AS total_revenue
91 FROM Orders o
92 JOIN Books b ON o.book_id = b.book_id
93 GROUP BY b.title
94 ORDER BY total_revenue DESC;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

title	total_units_sold	total_revenue
Data Science 101	7	209.93
Space Explorers	5	160.00
The Art of SQL	4	138.00
Learn Python the Hard Way	3	120.00
Fantasy World Chronicles	3	67.50
Cooking with Love	3	56.25
Mystery at the Bookstore	3	45.00
AI for Beginners	1	45.00

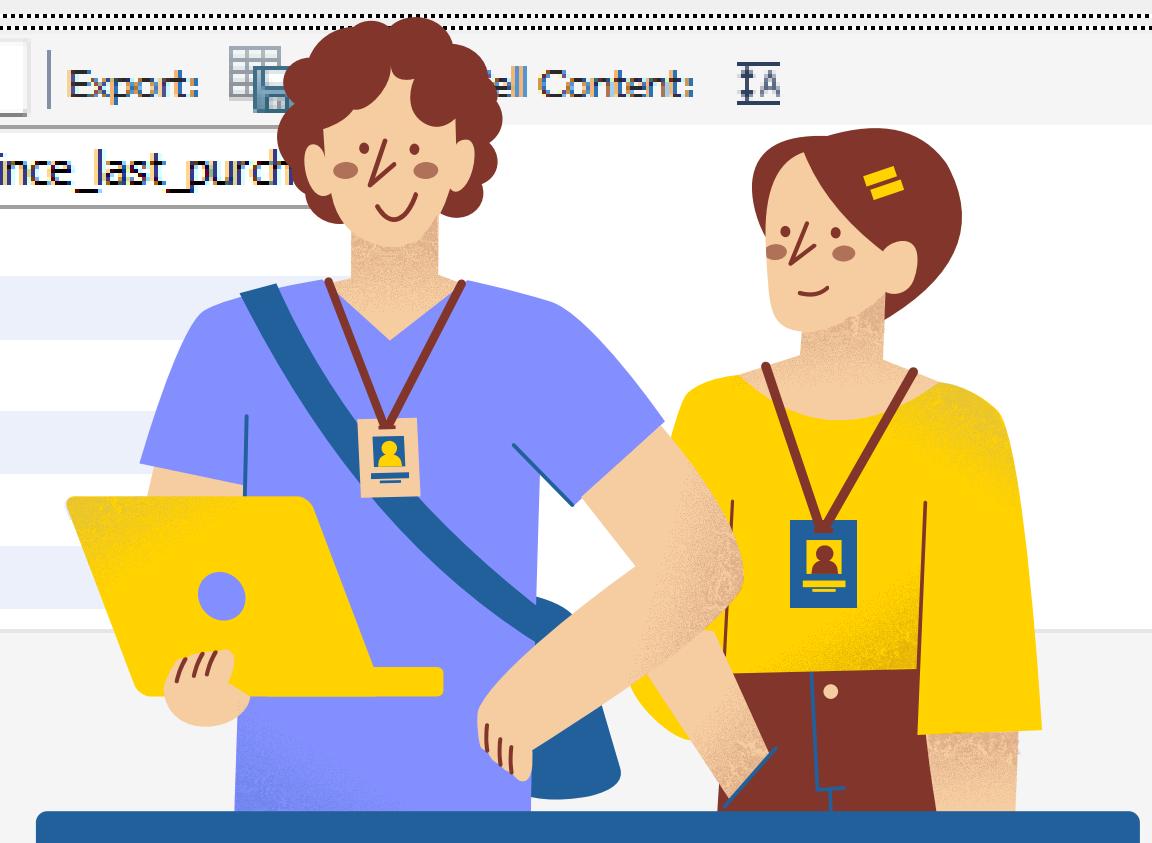
List customers whose last purchase was more than a year ago

```
165 • SELECT  
166     c.customer_id,  
167     c.name,  
168     MAX(o.order_date) AS last_purchase,  
169     DATEDIFF('2025-07-01', MAX(o.order_date)) AS days_since_last_purchase  
170   FROM Customers c  
171   JOIN Orders o ON c.customer_id = o.customer_id  
172   GROUP BY c.customer_id  
173   HAVING days_since_last_purchase > 365;  
174
```



result Grid | Filter Rows: Export: Cell Content:

customer_id	name	last_purchase	days_since_last_purchase
1	Alice	2024-06-06	390
2	Bob	2024-06-05	391
3	Charlie	2024-06-04	392
4	Diana	2024-06-04	392
5	Evan	2024-06-05	391
6	Fiona	2024-06-12	384



customers who live in New York

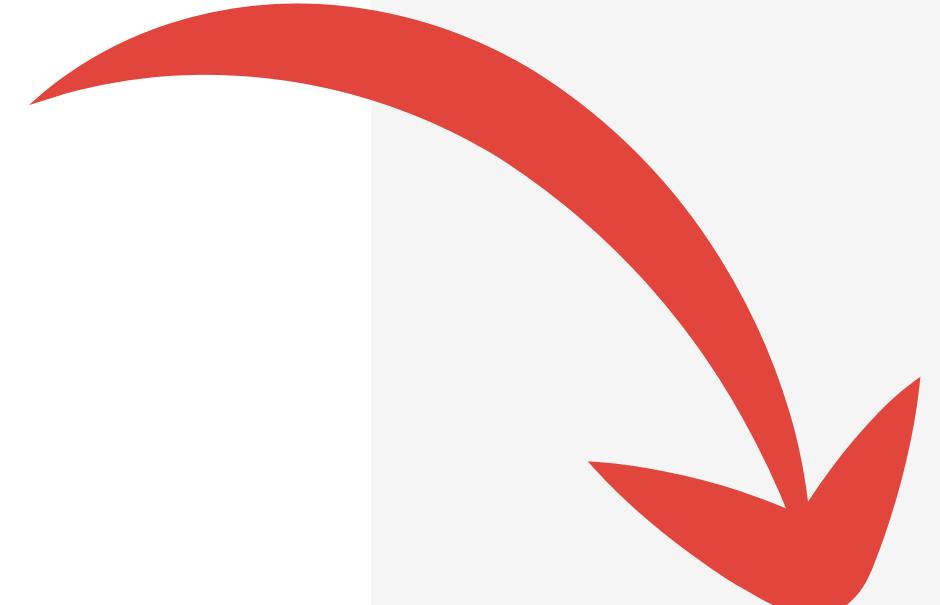
```
7 • SELECT name  
8   FROM customers  
9   WHERE city = 'New York';
```

result Grid | Filter Rows: Export:

name
Alice
Diana
Ian

RFM (Recency, Frequency, Monetary) metrics for each customer

```
WITH customer_metrics AS (
    SELECT
        c.customer_id,
        c.name,
        MAX(o.order_date) AS last_order,
        COUNT(o.order_id) AS frequency,
        SUM(o.quantity * b.price) AS monetary
    FROM Customers c
    JOIN Orders o ON c.customer_id = o.customer_id
    JOIN Books b ON o.book_id = b.book_id
    GROUP BY c.customer_id
)
SELECT *,
    DATEDIFF('2024-07-01', last_order) AS recency_days
FROM customer_metrics;
```



customer_id	name	last_order	frequency	monetary	recency_days
1	Alice	2024-06-06	3	104.97	25
2	Bob	2024-06-05	2	74.50	26
3	Charlie	2024-06-04	1	89.97	27
4	Diana	2024-06-04	1	45.00	27
5	Evan	2024-06-05	1	69.00	26
6	Fiona	2024-06-12	3	98.25	19
7	George	2024-06-12	3	181.50	19
8	Hannah	2024-06-10	2	50.90	21
9	Ian	2024-06-11	2	125.99	20
10	Jane	2024-06-11	2	70.00	20

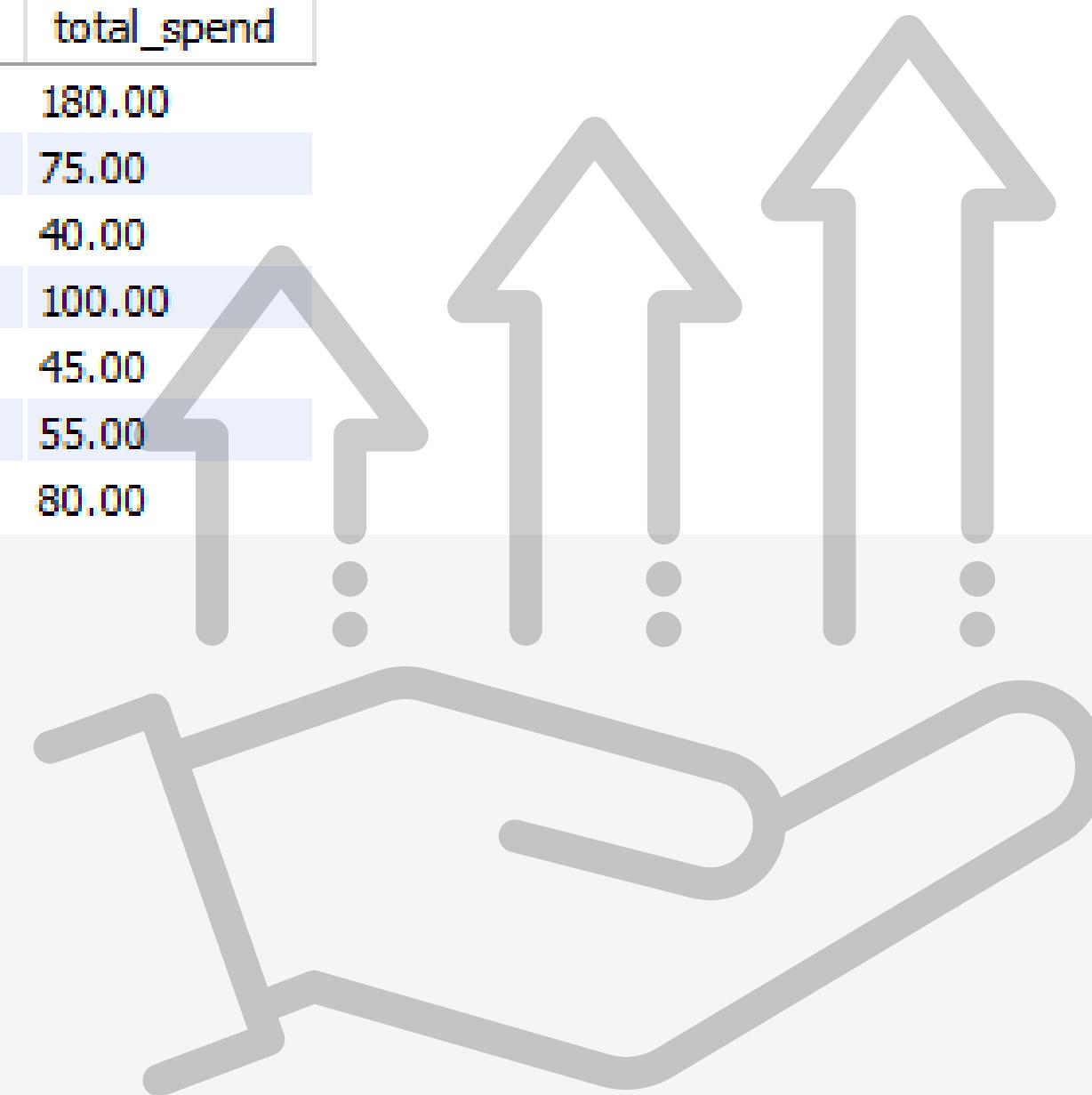


Total marketing spend for each city

```
30 •   SELECT c.city, SUM(m.spend_amount) AS total_spend  
31     FROM customers AS c  
32     JOIN marketingspend AS m ON c.customer_id = m.customer_id  
33     GROUP BY c.city;  
34
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

city	total_spend
New York	180.00
San Francisco	75.00
Austin	40.00
Chicago	100.00
Boston	45.00
Seattle	55.00
San Diego	80.00



customers who ordered the most expensive book

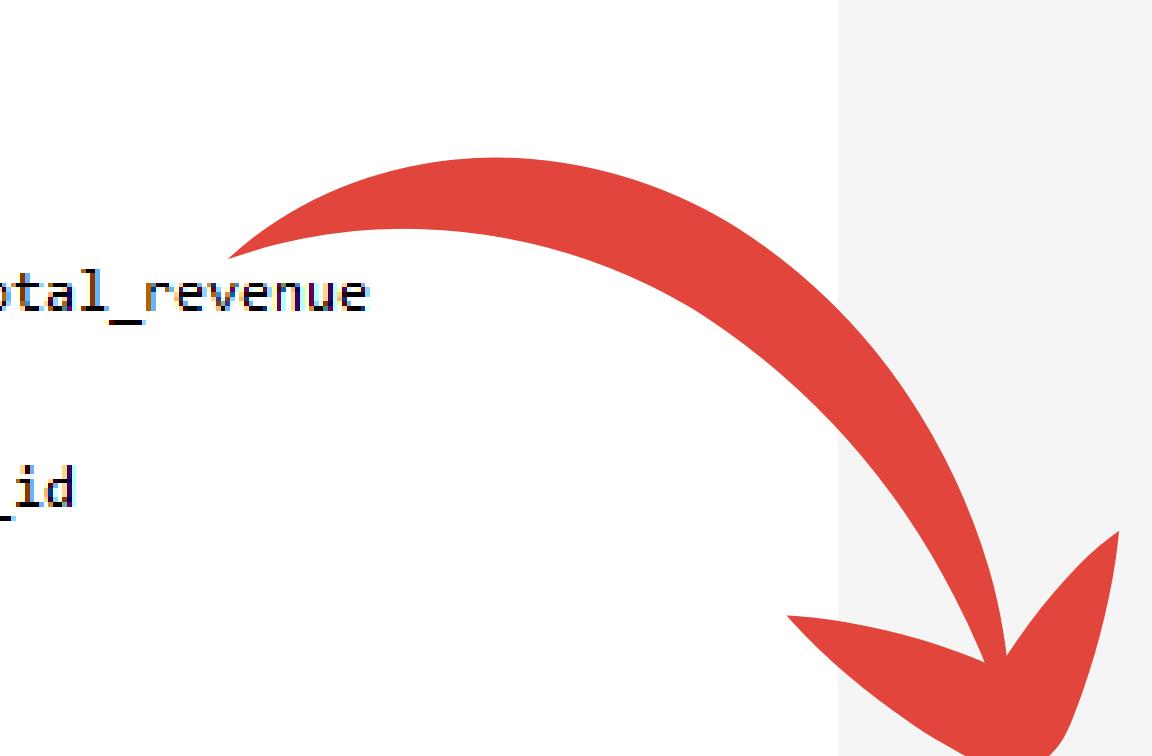
```
.02 •   SELECT c.name  
.03     FROM Customers AS c  
.04     JOIN Orders AS o ON c.customer_id = o.customer_id  
.05     JOIN Books AS b ON o.book_id = b.book_id  
.06     WHERE b.price = (SELECT MAX(price) FROM Books);
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

name
Fiona

For each customer, display their customer ID, name, total revenue from orders, marketing spend, and profit

```
WITH customer_spend AS (
    SELECT
        o.customer_id,
        SUM(o.quantity * b.price) AS total_revenue
    FROM Orders o
    JOIN Books b ON o.book_id = b.book_id
    GROUP BY o.customer_id
)
SELECT
    c.customer_id,
    c.name,
    ms.spend_amount,
    cs.total_revenue,
    (cs.total_revenue - ms.spend_amount) AS profit
FROM Customers c
JOIN MarketingSpend ms ON c.customer_id = ms.customer_id
JOIN customer_spend cs ON c.customer_id = cs.customer_id;
```



customer_id	name	spend_amount	total_revenue	profit
1	Alice	50.00	104.97	54.97
2	Bob	75.00	74.50	-0.50
3	Charlie	40.00	89.97	49.97
4	Diana	60.00	45.00	-15.00
5	Evan	35.00	69.00	34.00
6	Fiona	45.00	98.25	53.25
7	George	55.00	181.50	126.50
8	Hannah	65.00	50.90	-14.10
9	Ian	70.00	125.99	55.99
10	Jane	80.00	70.00	-10.00



customers who have spent more than the average marketing spend

```
29 •   SELECT c.name  
30      FROM Customers AS c  
31      JOIN MarketingSpend AS m  
32          ON c.customer_id = m.customer_id  
33  ⚡ WHERE m.spend_amount > (  
34      SELECT AVG(spend_amount)  
35      FROM MarketingSpend  
36  );
```

Result Grid | Filter Rows: Export: Wrap Cell Counter

name
Bob
Diana
Hannah
Ian
Jane



Top 2 books by total quantity sold

```
5 •   SELECT b.title, SUM(o.quantity) AS total_sold  
6      FROM Orders o  
7      JOIN Books b ON o.book_id = b.book_id  
8      GROUP BY b.title  
9      ORDER BY total_sold DESC  
10     LIMIT 2;
```

Result Grid | Filter Rows: Export: Wrap Cell Counter

title	total_sold
Data Science 101	7
Space Explorers	5

Insights & Recommendations



Customer Insights



- Most active customers: Alice, Ian, Jane (high revenue & frequent orders)
- Some customers haven't purchased in over a year → risk of churn
- Top spenders sometimes correlate with higher marketing spend

Recommendations:

- Implement loyalty programs for high-value customers
- Run re-engagement campaigns for inactive customers
- Personalize promotions for repeat buyers to increase retention



Sales & Revenue Analysis



- Best-selling books: “Data Science 101”, “The Art of SQL”
- Highest revenue genres: Education, Technology
- Average order value can inform pricing and bundling strategies
- Revenue is concentrated among certain months (seasonal trend)

Recommendations:

- Stock more high-demand books and popular genres
- Bundle slower-selling books with popular titles
- Use revenue trends to plan marketing campaigns seasonally



Inventory Management

- Books like “Fantasy World Chronicles” and “Space Explorers” are low on stock
- Some books never sold → may need promotions or discounts
- High-priced books within a genre might deter buyers

Recommendations:

- Set automated low-stock alerts
- Promote unsold books via discounts or bundles
- Reassess pricing strategy for books above genre average



Marketing & ROI

- Marketing spend is not always proportional to revenue
- Customers with moderate marketing spend sometimes generate higher ROI

Recommendations:

- Optimize marketing budget by targeting cost-effective channels
- Focus on cities or segments with high ROI
- Track profit per customer to evaluate marketing efficiency

