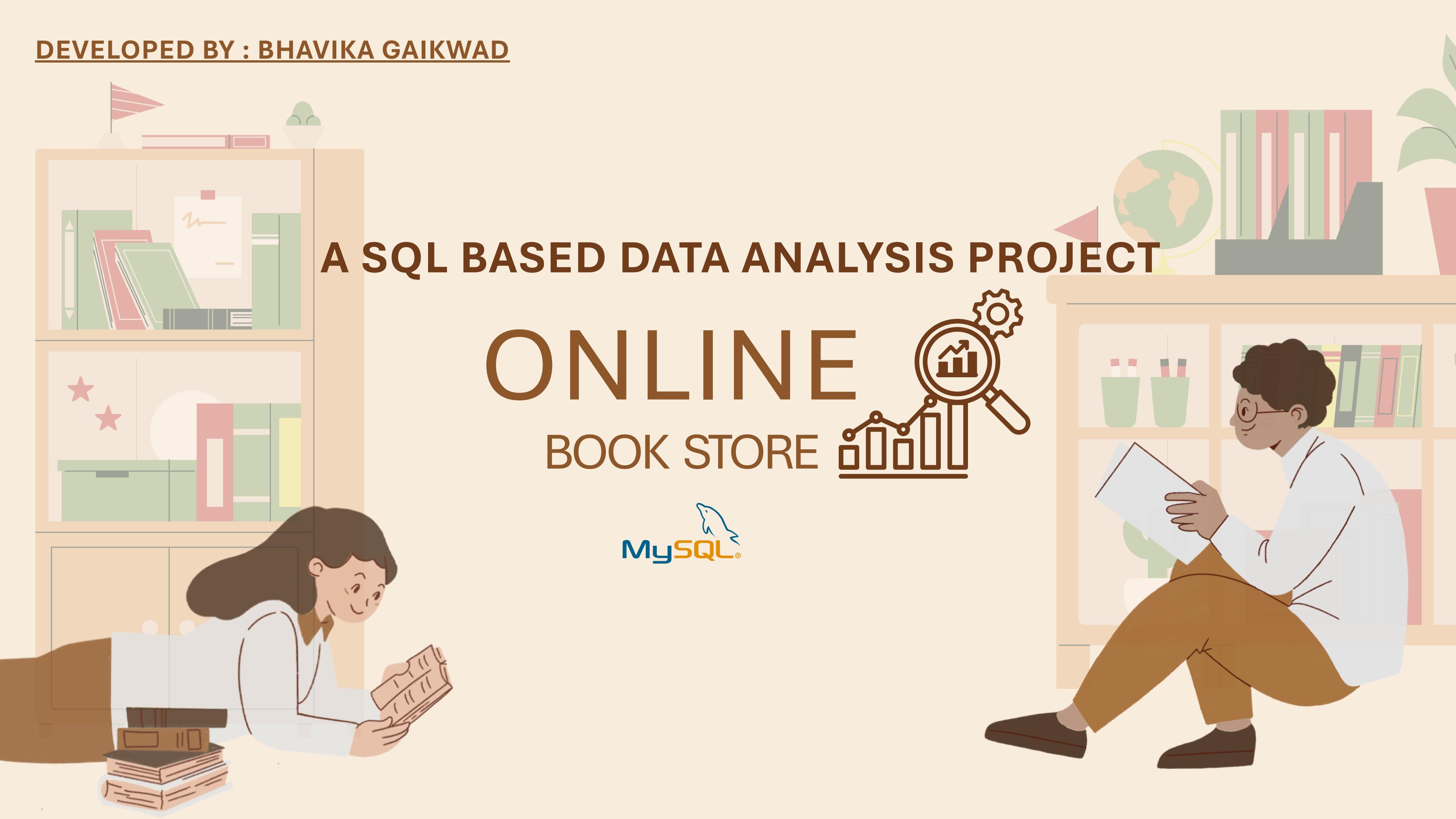


DEVELOPED BY : BHAVIKA GAIKWAD

A SQL BASED DATA ANALYSIS PROJECT

ONLINE BOOK STORE

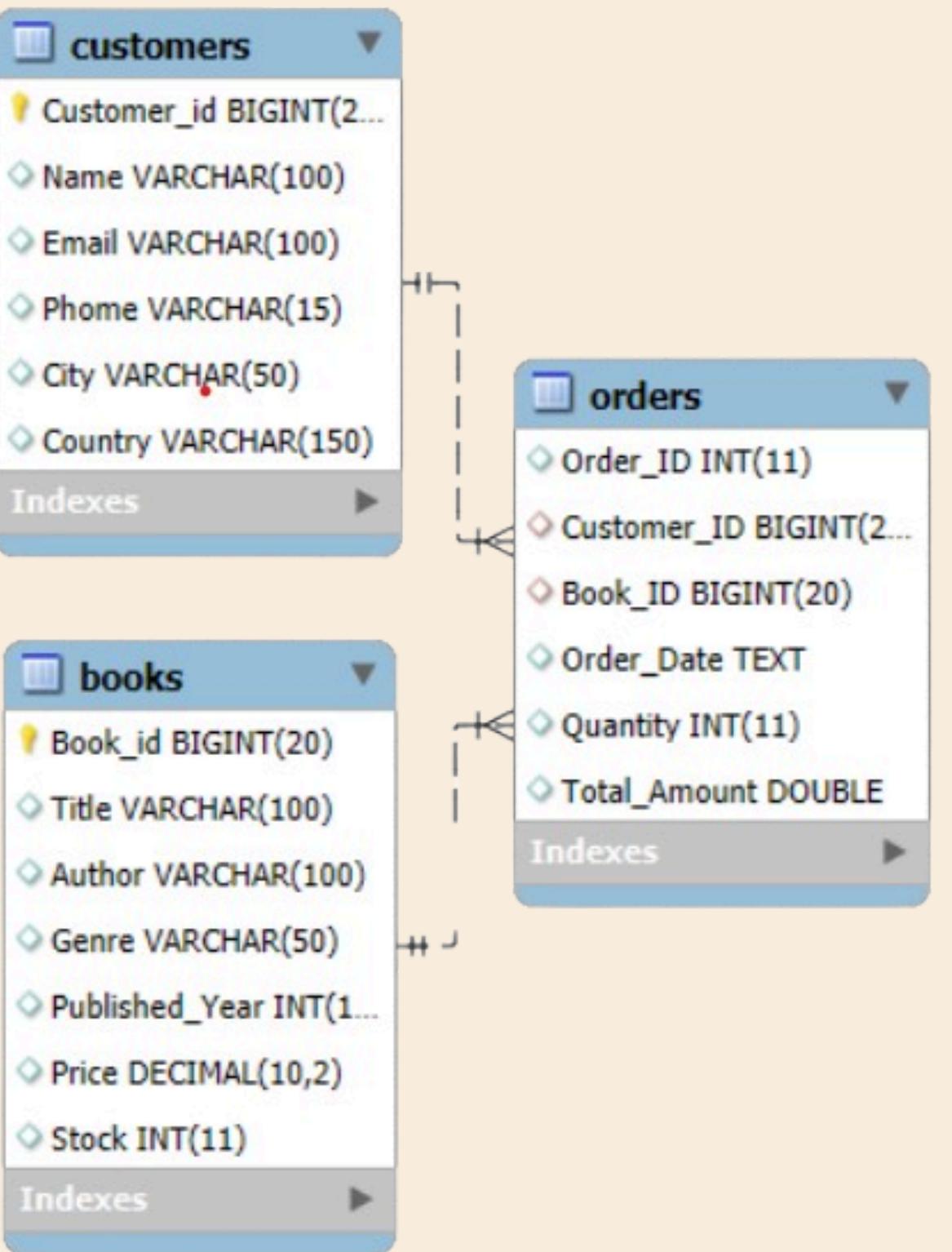


ABSTRACT

The Online Book Store Management System is a MySQL-based database project developed to manage books, customers, and orders efficiently. The system uses relational database design with primary and foreign keys to ensure data accuracy and integrity. SQL queries, joins, aggregations, and window functions are implemented to analyze sales trends, customer behavior, and order performance. This project demonstrates practical application of database concepts and analytical querying in a real-world business scenario.

ER-DIAGRAM

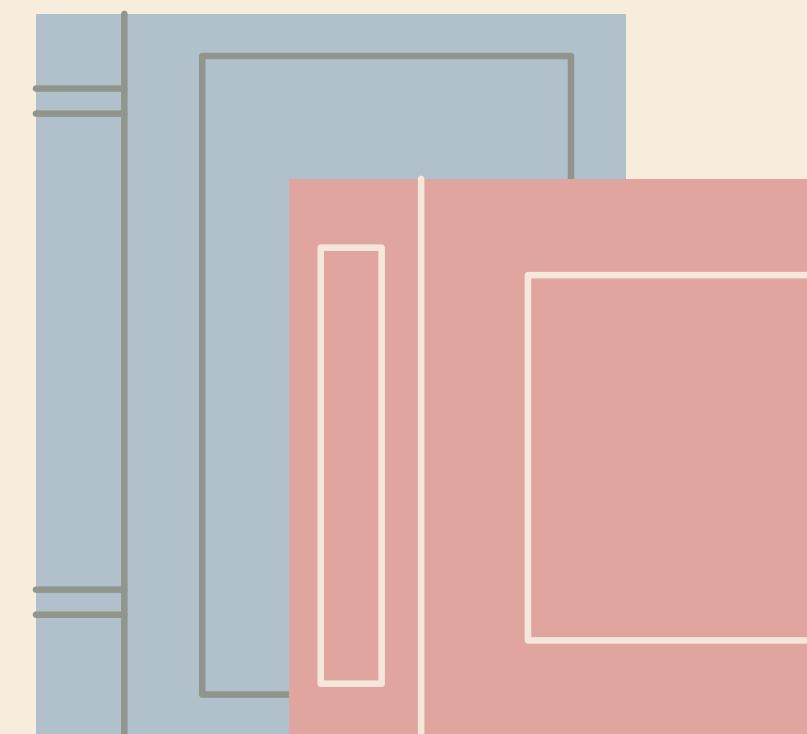
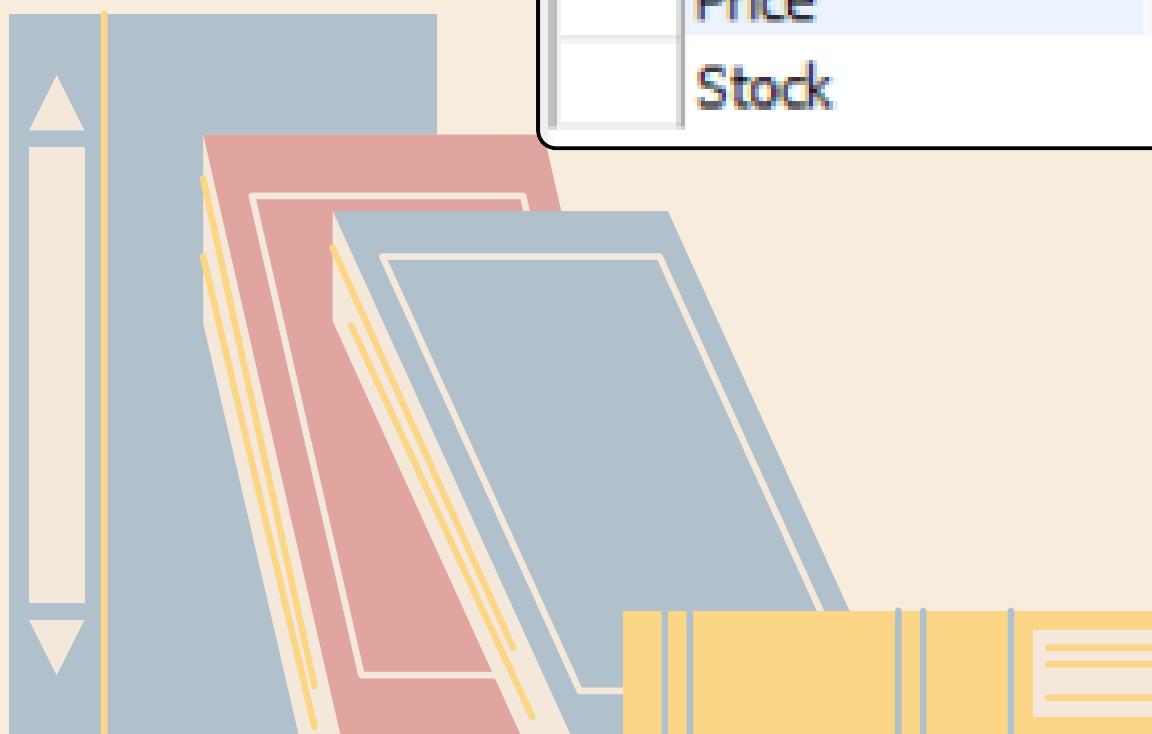
SHOWS THE STRUCTURE OF THE DATABASE



STRUCTURE OF THE TABLE

```
describe books;
```

	Field	Type	Null	Key	Default	Extra
▶	Book_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
	Title	varchar(100)	YES		NULL	
	Author	varchar(100)	YES		NULL	
	Genre	varchar(50)	YES		NULL	
	Published_Year	int(11)	YES		NULL	
	Price	decimal(10,2)	YES		NULL	
	Stock	int(11)	YES		NULL	



STRUCTURE OF THE TABLE

```
describe customers;
```

	Field	Type	Null	Key	Default	Extra
▶	Customer_id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
	Name	varchar(100)	YES		NULL	
	Email	varchar(100)	YES		NULL	
	Phone	varchar(15)	YES		NULL	
	City	varchar(50)	YES		NULL	
	Country	varchar(150)	YES		NULL	

STRUCTURE OF THE TABLE

describe orders;

Result Grid | Filter Rows: Export: Wrap Cell Content

	Field	Type	Null	Key	Default	Extra
▶	Order_ID	int(11)	YES		NULL	
	Customer_ID	bigint(20) unsigned	YES	MUL	NULL	
	Book_ID	bigint(20) unsigned	YES	MUL	NULL	
	Order_Date	text	YES		NULL	
	Quantity	int(11)	YES		NULL	
	Total_Amount	double	YES		NULL	

CONTENTS OF TABLE: BOOKS

`select * from Books;`

THIS TABLE CONTAINS 500 RECORDS

	Book_id	Title	Author	Genre	Published_Year	Price	Stock
▶	1	Configurable modular throughput	Joseph Crane	Biography	1949	21.34	100
	2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19
	3	Streamlined coherent initiative	Derrick Howard	Non-Fiction	1913	15.75	27
	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
	7	Open-architected exuding structure	Jacqueline Young	Romance	1927	43.63	95
	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
	11	Secured zero tolerance time-frame	Denise Barnes	Fantasy	1998	35.95	10
	12	Polarized optimal array	Destiny Scott	Non-Fiction	1989	27.43	63
	13	Adaptive 5thgeneration orchestration	Jadyn Miller	Romance	1913	14.04	99
	14	Re-engineered demand-driven parallelism	Jeremy Hayes	Science Fiction	1933	6.04	95

Books 7 ×

Customers 8

Orders 9

CONTENTS OF TABLE: CUSTOMERS

```
select * from Customers;
```

THIS TABLE CONTAINS 500 RECORDS

	Customer_id	Name	Email	Phone	City	Country
▶	1	Deborah Griffith	balljoseph@wright-keith.net	1234567891	South Craigfort	Denmark
	2	Crystal Clements	kimberlybennett@curtis.com	1234567892	East Derekberg	Nicaragua
	3	Susan Fuller	beanmichael@burnett-stewart.net	1234567893	Austinbury	Equatorial Guinea
	4	Jamie Ramirez	amandahood@warren.com	1234567894	Dianemouth	Slovenia
	5	Marcus Murphy	connerjohn@yahoo.com	1234567895	Smithbury	Guinea-Bissau
	6	Stephen Vasquez	ricemiguel@yahoo.com	1234567896	Hamiltonstad	Rwanda
	7	Susan Hicks	jeffrey91@yahoo.com	1234567897	East Rebecca	Montenegro
	8	Matthew Johnson	austinkenneth@manning.net	1234567898	Kirstenborough	Israel
	9	Matthew Williams	jeffrey41@diaz.com	1234567899	Rebeccafurt	Somalia
	10	Ronald Osborn	staciekelley@heath.com	1234567900	Lake Benjamin	Cameroon
	11	Thomas Garcia	rmiller@gmail.com	1234567901	West Monicabury	Argentina
	12	Jennifer Murray	wilsonbrittany@hotmail.com	1234567902	South Ashley...	Zimbabwe
	13	Kristine Kim	sarahwilliams@hotmail.com	1234567903	Lake Robert	Nigeria
	14	John Wood	johnsonalexander@gmail.com	1234567904	Richardsonville	New Caledonia

Customers 8 ×

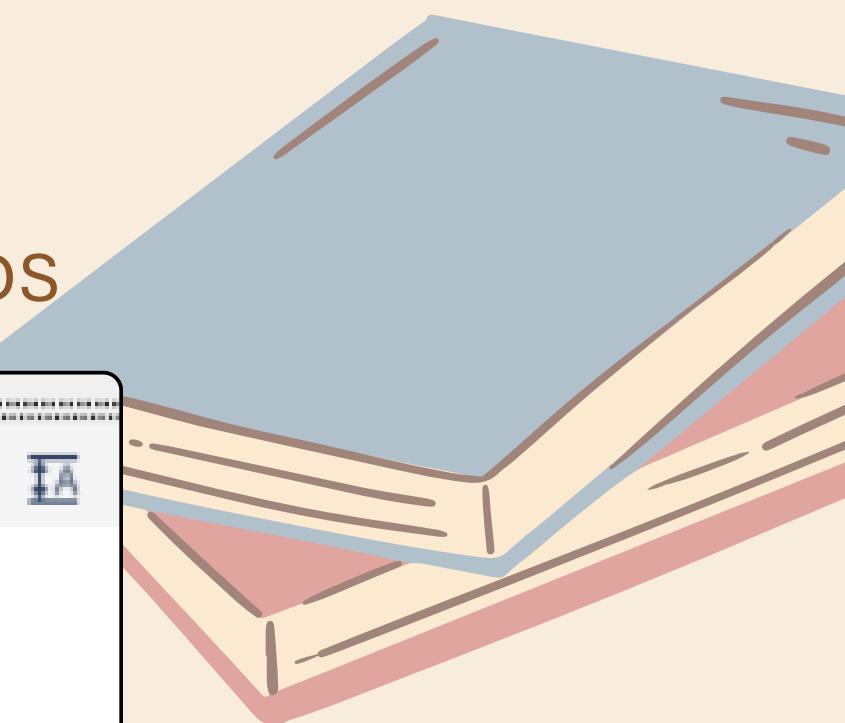
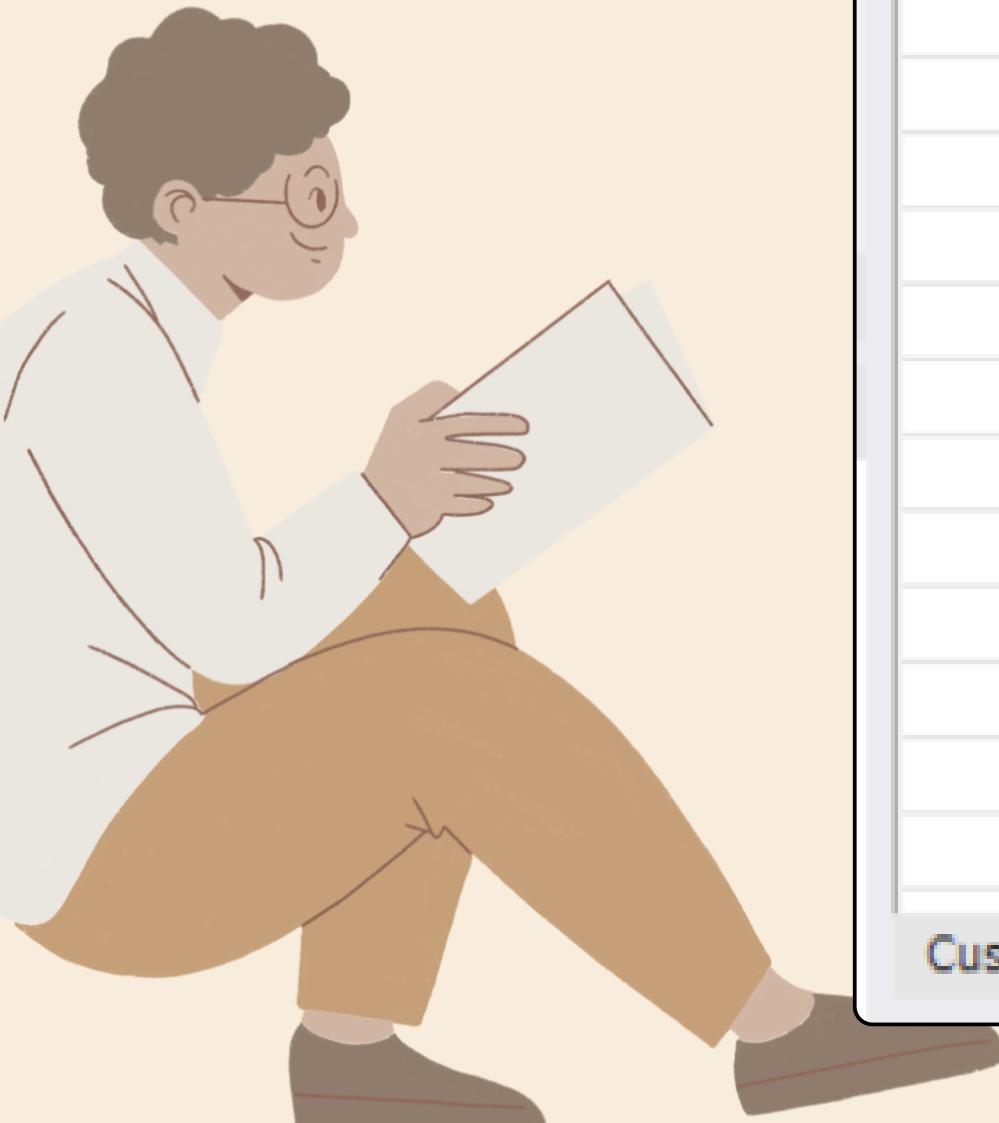
Orders 9



CONTENTS OF TABLE: ORDERS

`select * from Orders;`

THIS TABLE CONTAINS 500 RECORDS

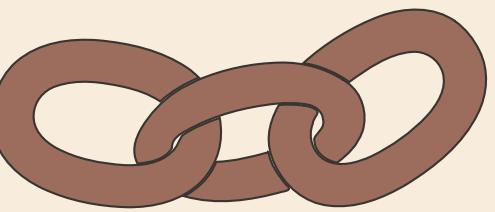


	Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
▶	1	84	169	26-05-2023	8	188.56
	2	137	301	23-01-2023	10	216.6
	3	216	261	27-05-2024	6	85.5
	4	433	343	25-11-2023	7	301.21
	5	14	431	26-07-2023	7	136.36
	6	439	119	11-10-2024	5	249.4
	7	195	467	23-10-2023	6	82.92
	8	32	159	07-05-2024	4	144.84
	9	109	407	04-01-2024	9	379.71
	10	94	122	09-07-2024	4	123
	11	131	206	16-10-2023	1	38.01
	12	454	3	17-06-2024	2	31.5
	13	420	180	08-06-2023	5	125.45
	14	454	319	24-08-2023	2	85.22

Customers 8 Orders 9 ×



JOINS



A join is used to combine rows from two or more tables based on a related column, allowing you to retrieve data from multiple tables in a single query.

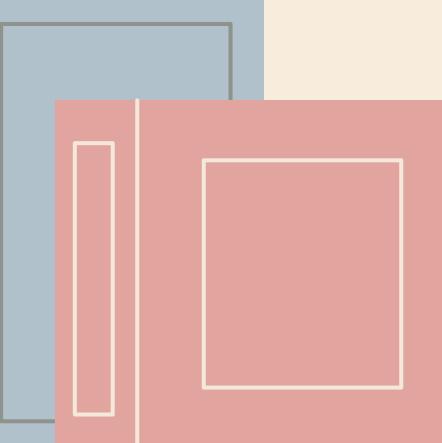
• FINDING THE BESTSELLERS

```
-- 4. Find the most frequently ordered book;(added)
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;
```

Result Grid			
	book_id	title	order_count
▶	273	Devolved zero administration process improvem...	4

CONCLUSION: THIS QUERY IDENTIFIES THE MOST FREQUENTLY ORDERED BOOK, REVEALING THE HIGHEST CUSTOMER DEMAND AND HELPING THE BOOKSTORE PRIORITIZE POPULAR TITLES FOR BETTER SALES AND INVENTORY PLANNING.





• PROFITABLE CUSTOMERS

```
-- 8. Find the customer who spent the most on orders;  
select c.name,sum(o.total_amount) as total_spent  
from customers c join orders o on c.customer_id=o.customer_id  
group by c.name  
order by total_spent desc limit 1 ;
```



A screenshot of a database query results interface. At the top, there are buttons for "Result Grid" (selected), "Filter Rows:", "Export:", and "Wrap Cell Content:". The result grid shows a single row of data:

	name	total_spent
▶	Kim Turner	1398.9

CONCLUSION: THIS QUERY IDENTIFIES THE HIGHEST-SPENDING CUSTOMER IN THE ONLINE BOOKSTORE BY CALCULATING TOTAL PURCHASE VALUE, PROVIDING KEY INSIGHT FOR LOYALTY PROGRAMS, PERSONALIZED OFFERS, AND CUSTOMER RETENTION STRATEGIES.

- LEAST SPENDING CUSTOMERS FOR GIVING OFFERS AND DISCOUNT

```
-- least spending customers for giving offers and discount  
SELECT c.customer_id, c.name, c.city,  
SUM(o.total_amount) AS total_spent  
FROM customers c JOIN orders o  
ON c.customer_id = o.customer_id  
GROUP BY c.customer_id, c.name, c.city  
ORDER BY total_spent ASC LIMIT 5;
```



	customer_id	name	city	total_spent
▶	100	Karla Guerrero	Dustinview	9.83
	7	Susan Hicks	East Rebecca	10.07
	101	Brooke Kline	New James	10.17
	475	Joseph Boyd	Derekview	11.31
	392	Dustin Ramirez	Ryanhaven	11.99

Result 14 ×

CONCLUSION: THIS QUERY IDENTIFIES THE BOTTOM FIVE LEAST-SPENDING CUSTOMERS, OFFERING VALUABLE INSIGHT FOR DESIGNING TARGETED DISCOUNTS, PROMOTIONAL OFFERS, AND ENGAGEMENT STRATEGIES TO ENCOURAGE HIGHER PURCHASING ACTIVITY.

- STOCK REMAINING RESULT FOR STOCK REFILLING

```
-- 9 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;
```

CONCLUSION:

THIS QUERY EVALUATES CURRENT INVENTORY STATUS BY COMPARING AVAILABLE STOCK WITH ORDERED QUANTITIES, HELPING IDENTIFY REMAINING STOCK LEVELS AND SUPPORTING EFFECTIVE INVENTORY CONTROL AND TIMELY RESTOCKING DECISIONS.



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

Result 1 ×

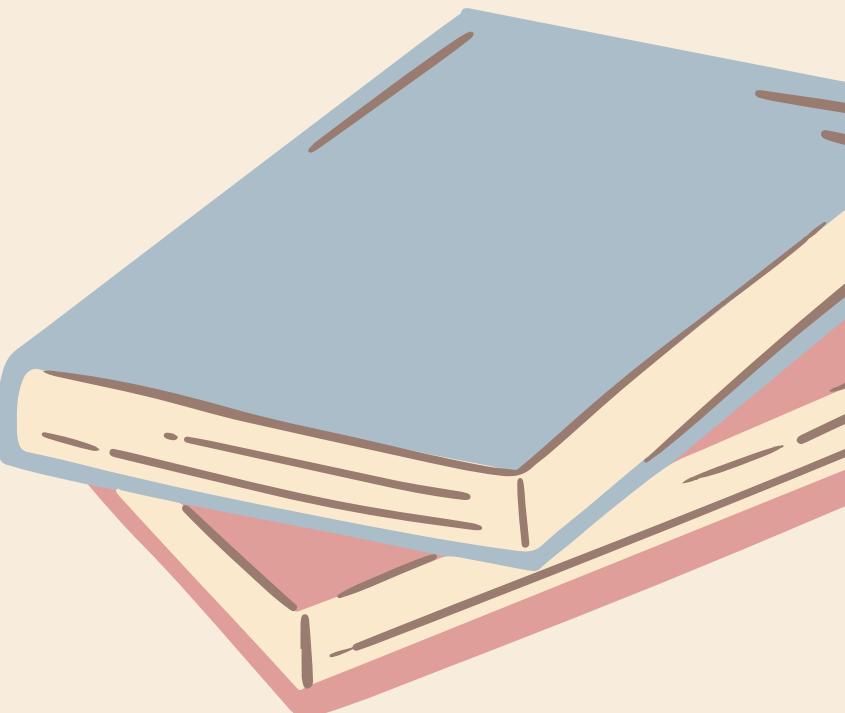
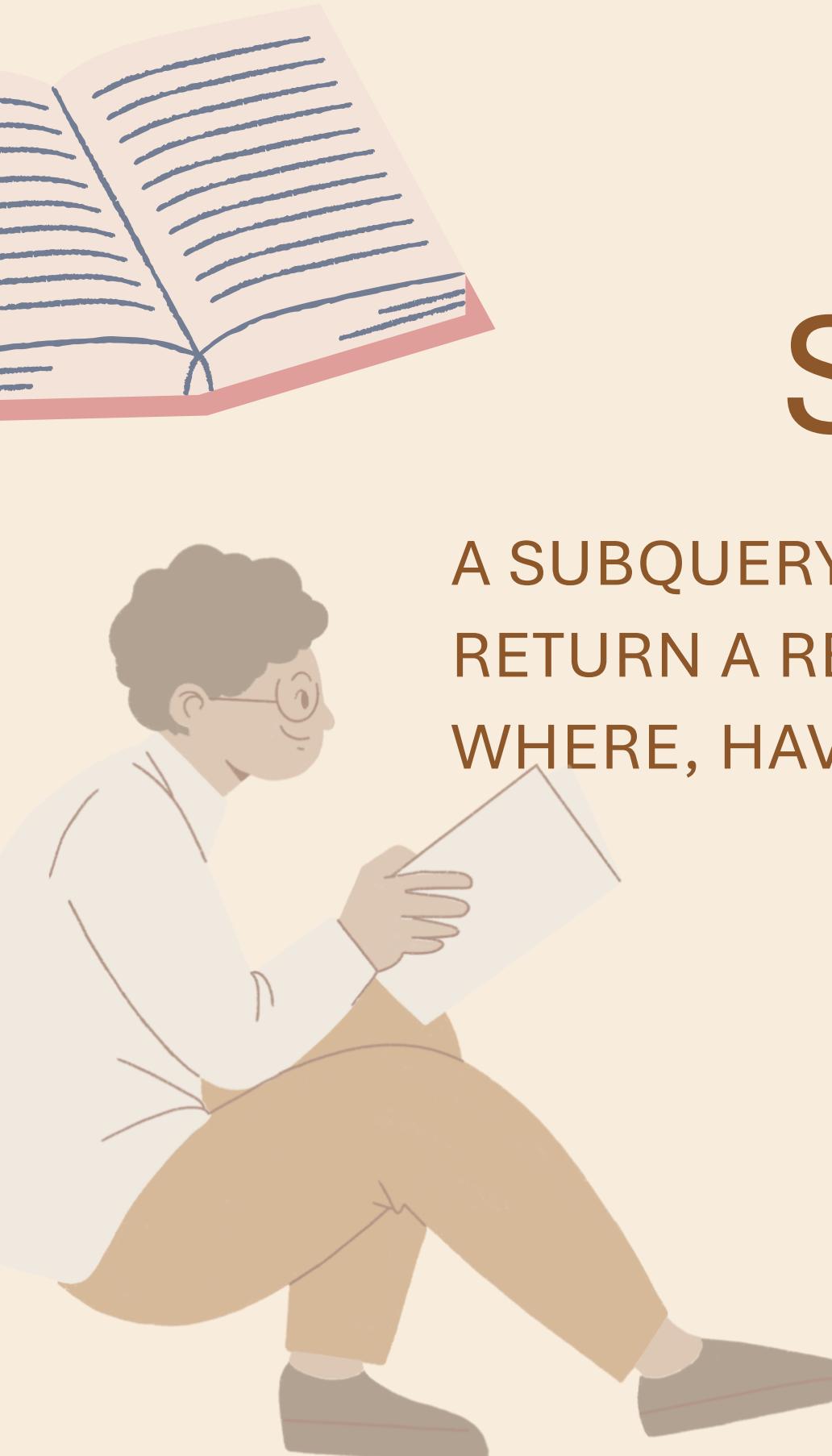
Output

Action Output

#	Time	Action	Message
1	15:00:44	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;	Error Code: 1046. No database selected
2	15:00:50	use onlinebookstore	0 row(s) affected
3	15:01:00	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;	500 row(s) returned

SUBQUERY {} {}

A SUBQUERY IS A QUERY NESTED INSIDE ANOTHER QUERY, USED TO RETURN A RESULT THAT IS THEN USED BY THE OUTER QUERY, OFTEN IN A WHERE, HAVING, OR FROM CLAUSE.



- DETECT LOW-PERFORMING INVENTORY.



```
select book_id, title from books where book_id not in  
(select distinct book_id from orders);
```

	book_id	title
▶	2	Persevering reciprocal knowledge user
	4	Customizable 24hour product
	6	Advanced encompassing implementation
	9	Optimized interactive challenge
	12	Polarized optimal array
	14	Re-engineered demand-driven parallelism
	15	User-friendly motivating strategy
	20	Face-to-face systematic throughput
	22	Multi-layered optimizing migration
	23	Reverse-engineered context-sensitive hardware
	24	Ergonomic incremental hub
	27	Team-oriented human-resource access
	30	Multi-layered global open system
	37	Up-sized tertiary archive



CONCLUSION:
THIS QUERY IDENTIFIES BOOKS THAT HAVE NOT RECEIVED ANY ORDERS, HIGHLIGHTING LOW-DEMAND OR INACTIVE TITLES AND SUPPORTING DECISIONS RELATED TO PROMOTIONS, PRICING STRATEGIES, OR INVENTORY OPTIMIZATION.

- TARGET CUSTOMERS ELIGIBLE FOR DISCOUNTS.

```
select c.customer_id, c.name, o.total_amount  
from customers c  
join orders o on c.customer_id = o.customer_id  
where o.total_amount =  
(select MIN(total_amount) from orders);
```

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

	customer_id	name	total_amount
▶	420	Andrew Murray	5.37

CONCLUSION:

THIS QUERY IDENTIFIES THE CUSTOMER ASSOCIATED WITH THE LOWEST-VALUE ORDER, PROVIDING INSIGHT INTO MINIMAL PURCHASING BEHAVIOR AND HELPING THE BUSINESS PLAN ENTRY-LEVEL OFFERS OR INCENTIVES TO INCREASE CUSTOMER SPENDING.



VIEW



A view is a virtual table created by a SELECT query that can encapsulate complex logic, allowing users to query the view as if it were a real table. It does not store data itself but presents a stored result of a query.



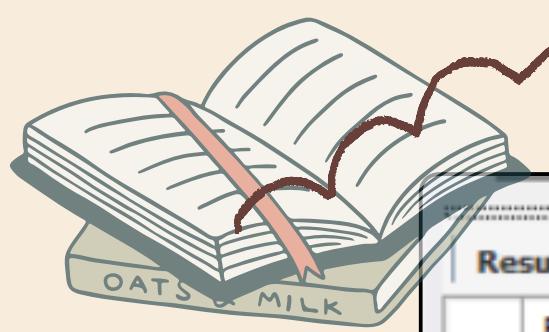
- SUMMARIZES TOTAL QUANTITY SOLD AND TOTAL REVENUE PER BOOK.

```
-- Summarizes total quantity sold and total revenue per book.  
create view vw_book_sales_summary as  
select b.Book_ID, b.Title,  
       SUM(o.Quantity) as Total_Quantity_Sold,  
       SUM(o.Total_Amount) as Total_Revenue  
from Books b  
join Orders o  
on b.Book_ID = o.Book_ID  
group by b.Book_ID, b.Title;
```

HOW TO USE

```
select * from vw_book_sales_summary;
```





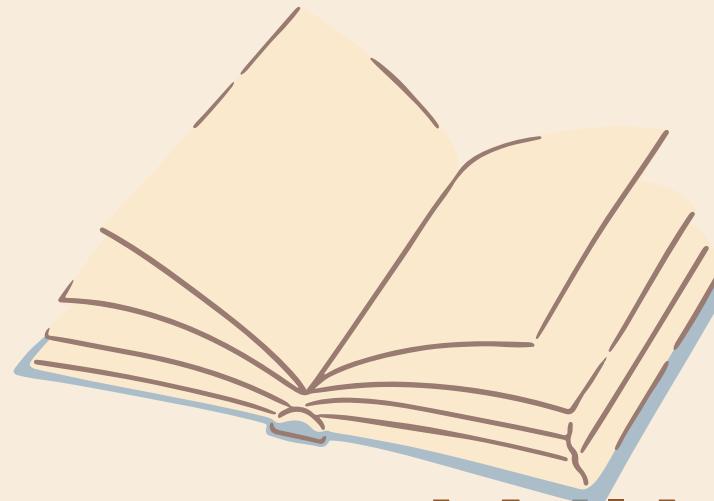
Result Grid | Filter Rows: Export: Wrap Cell Content:

	Book_ID	Title	Total_Quantity_Sold	Total_Revenue
▶	1	Configurable modular throughput	3	64.02
	3	Streamlined coherent initiative	5	78.75
	5	Adaptive 5thgeneration encoding	8	87.60000000000001
	7	Open-architected exuding structure	5	218.15
	8	Persistent local encoding	3	146.97
	10	Ergonomic national hub	1	24.63
	11	Secured zero tolerance time-frame	5	179.75
	13	Adaptive 5thgeneration or Adaptive 5thgeneration orchestration	36	36
	16	Vision-oriented tangible project	1	10.07
	17	Reduced secondary core	9	48.33
	18	Adaptive 4thgeneration concept	14	552.5799999999999
	19	Progressive asymmetric Internet solu...	1	11.31
	21	Down-sized static interface	3	141.39
	25	Devolved mobile conglomeration	6	51.3
	26	Multi-channelled multi-tasking capability	11	231.5499999999998
	28	Expanded analyzing portal	2	75.02
	29	Quality-focused multi-tasking challenge	4	124.48
	31	Implemented encompassing conglom...	27	573.21
	32	Synergistic dedicated concept	10	215.6
	33	Team-oriented empowering synergy	15	525.9000000000001
	34	Stand-alone multimedia throughput	2	32.98
	35	Grass-roots solution-oriented adapter	9	286.29
	36	Open-source needs-based secured line	17	586.6700000000001
	38	Pre-emptive asynchronous leverage	10	164.7
	40	Adaptive didactic interface	1	25.97
	42	Pre-emptive interactive focus group	5	241.75
	43	Function-based zero-defect initiative	4	189.56

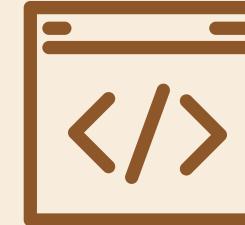
vw_book_sales_summary2 ×



CONCLUSION: THIS VIEW
SUMMARIZES BOOK-WISE SALES
AND REVENUE, ENABLING QUICK
PERFORMANCE TRACKING AND
INFORMED BUSINESS DECISIONS.



WINDOWS FUNCTIONS



A window function is a SQL function that performs calculations across a set of related rows, called a window, while preserving each row's detail in the result set. Unlike aggregate functions, window functions do not collapse rows; instead, they return a calculated value for every row, commonly used in the SELECT and ORDER BY clauses for ranking, running totals, and analytical comparisons.



• TO RANK CUSTOMERS ACCORDING TO THEIR SPENT

```
-- To rank customers in ranking when amounts are equal.
```

```
SELECT customer_id, sum(total_amount) as total_spent,  
rank() over ( order by sum(total_amount) desc) as rank_no from orders group by customer_id;
```



	customer_id	total_spent	rank_no
▶	457	1398.9	1
	174	1080.949999999998	2
	364	1052.27	3
	405	991	4
	386	986.300000000001	5
	425	942.62	6
	474	929.189999999999	7
	163	746.650000000001	8
	167	719.93	9
	214	682.15	10
	437	667.27	11
	98	633.900000000001	12
	284	631.5	13
	418	618.46	14

CONCLUSION:

THIS QUERY RANKS CUSTOMERS BASED ON TOTAL SPENDING, HELPING IDENTIFY TOP AND LOW SPENDERS FOR TARGETED REWARDS, PROMOTIONS, AND CUSTOMER SEGMENTATION STRATEGIES.

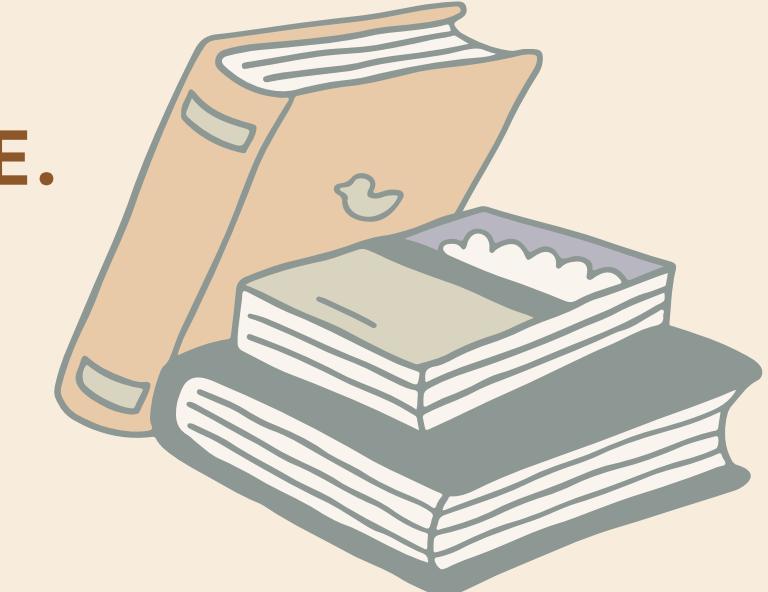
- TO RANK BOOKS WITHOUT GAPS EVEN IF PRICES ARE SAME.

-- To rank books without gaps even if prices are same.

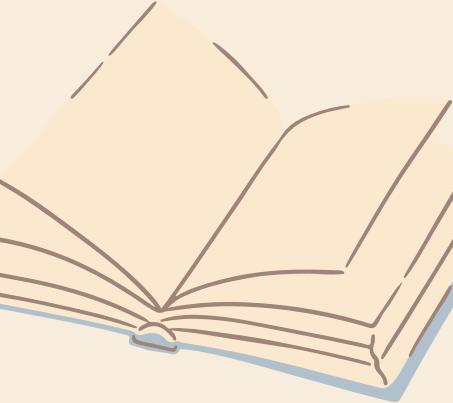
```
SELECT book_id, title, price,  
dense_rank() over ( order by price desc ) as price_rank from books;
```



book_id	title	price	price_rank
340	Proactive system-worthy orchestration	49.98	1
155	Optimized content-based standardization	49.96	2
240	Stand-alone content-based hub	49.90	3
100	Synchronized client-server service-desk	49.89	4
119	Switchable modular moratorium	49.88	5
260	Business-focused methodical application	49.59	6
95	Function-based heuristic analyzer	49.53	7
49	Robust attitude-oriented attitude	49.50	8
141	Exclusive hybrid monitoring	49.42	9
360	Profound tertiary encoding	49.34	10
462	Innovative 3rdgeneration database	49.23	11
488	Optimized attitude-oriented synergy	49.15	12
416	Expanded maximized portal	49.10	13
332	Quality-focused mobile migration	49.08	14

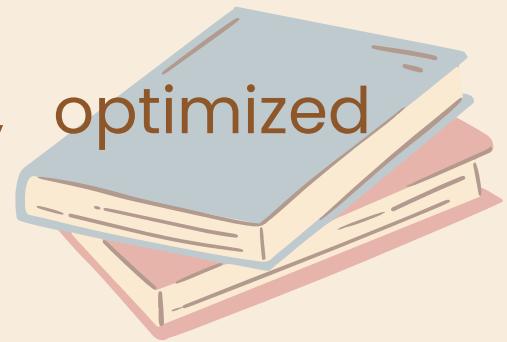


CONCLUSION:
THIS QUERY RANKS BOOKS
BASED ON PRICE LEVELS,
ENABLING EASY
IDENTIFICATION OF PREMIUM
AND LOW-PRICED TITLES FOR
PRICING ANALYSIS AND
PRODUCT POSITIONING.



FINAL CONCLUSION

- Sales analysis provided clear visibility into customer purchasing behavior and overall order trends, enabling better forecasting of demand.
- Identification of top-selling books and genres helped understand customer preferences and highlighted high-performing products.
- Revenue contribution analysis revealed key books and categories that significantly impacted overall sales, supporting strategic product prioritization.
- Customer spending patterns enabled segmentation of loyal and low-value customers, helping design targeted offers and promotional strategies.
- Order history analysis supported operational planning by identifying peak transaction periods and improving order management efficiency.
- SQL views and window functions enabled reusable, dynamic reporting for sales, customer performance, and revenue insights.
- Data-driven inventory and customer insights contribute to improved decision-making, optimized operations, and enhanced customer satisfaction.





THANK YOU

EXPLORING THE INSIGHTS BEHIND THE DATA.