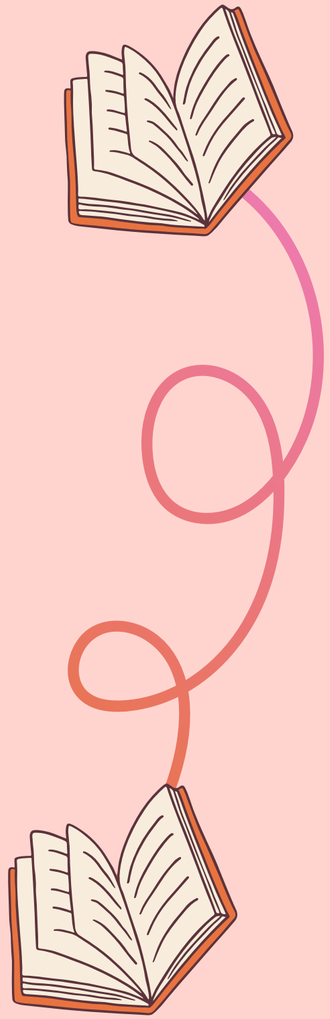
BOOK STORE ANALYSIS

TOOL: MY SQL

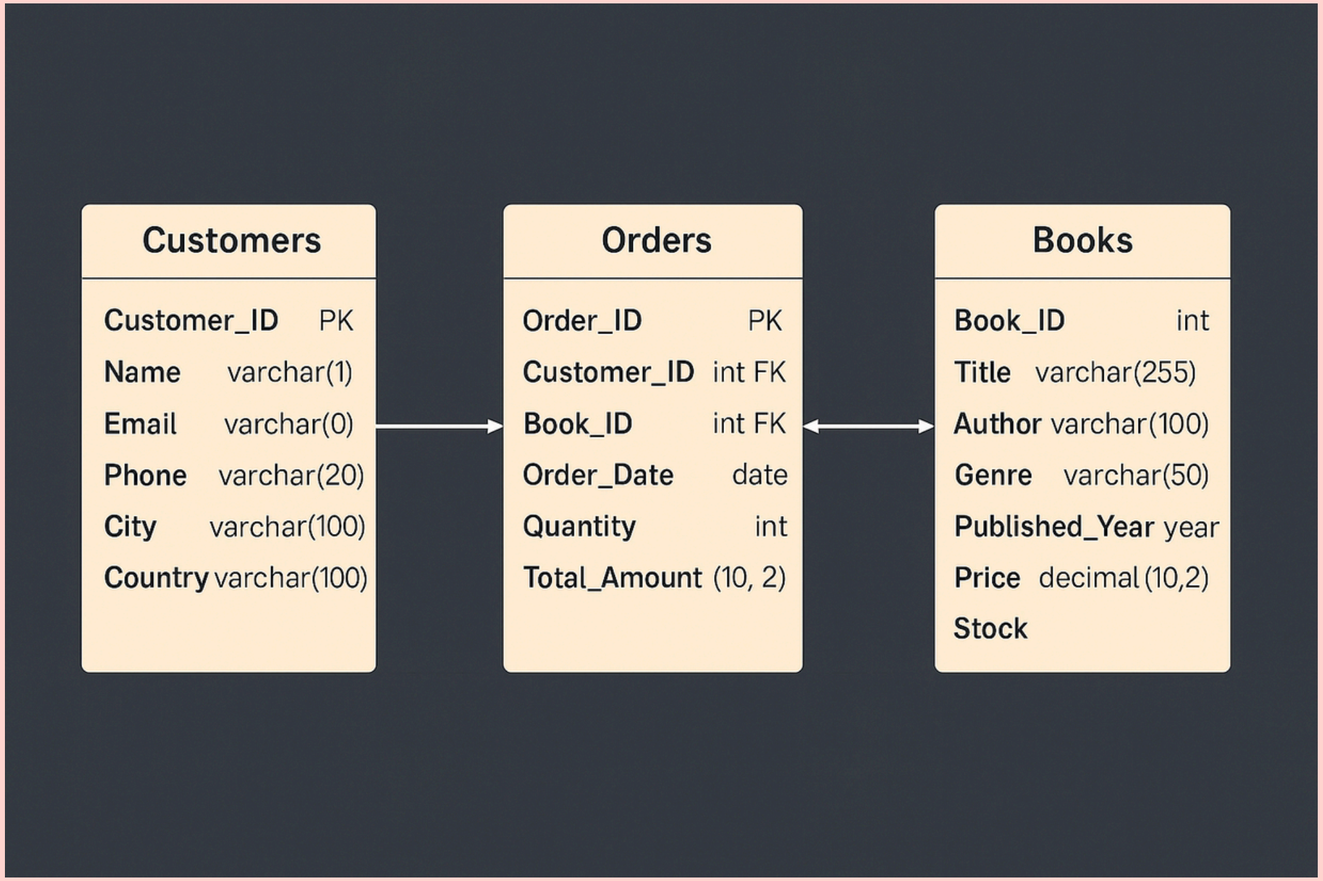
Hello, I'm Mansi Kala.
In this SQL project, I analyzed bookstore sales data to answer key business questions. My work includes identifying top genres, bestselling books, customer purchase patterns, and author-wise sales performance.
I also explored trends like high-value orders, frequently ordered books, and city-wise customer distribution.

Tools Used: MySQL

The next slides showcase my queries, insights, and takeaways from this analysis.



SCHEMA OVERVIEW



QUESTIONS

- Show orders placed in November 2023
- Books published after the year 1950
- All genres available in the Books table
- Retrieve the total number of books sold for each genre
- Find the average price of books in the "Fantasy" genre
- List customers who have placed at least 2 orders
- Find the most frequently ordered book
- Show the top 3 most expensive books of 'Fantasy' Genre
- Retrieve the total quantity of books sold by each author
- List the cities where customers who spent over $30 are located
- Find the customer who spent the most on orders

SHOW ORDERS PLACED IN NOVEMBER 2023

```sql
select* from orders where order_date between '2023-11-01' and '2023-11-30';
```

Result Grid | Filter Rows: | Edit: | Export/I

Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
182	129	293	2023-11-01	7	125.51
245	386	97	2023-11-01	9	411.66
429	449	146	2023-11-01	7	101.50
432	420	168	2023-11-04	3	42.39
257	123	403	2023-11-06	1	15.01
322	270	112	2023-11-08	2	16.04
414	23	234	2023-11-10	1	7.15
231	22	384	2023-11-11	1	33.92
288	6	128	2023-11-13	1	24.04
252	405	387	2023-11-15	5	237.10
19	496	60	2023-11-17	9	316.26
213	325	447	2023-11-17	7	253.75
307	368	133	2023-11-17	1	20.96
389	485	391	2023-11-18	2	66.84

BOOKS PUBLISHED AFTER THE YEAR 1950

```
select* from books where published_year >1950;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
166	Customizable discrete Graphical User In...	Rebecca Alexander	Romance	1951	11.02	56
174	Pre-emptive executive knowledge user	Rebecca Mann	Mystery	1951	37.83	18
432	Horizontal disintermediate alliance	Rodney Ward	Non-Fiction	1951	8.84	55
43	Function-based zero-defect initiative	Daniel Nunez	Romance	1952	47.39	61
150	Phased logistical open system	Jenna Henderson	Biography	1952	31.95	32
62	Re-contextualized real-time strategy	Nicole Lynch	Fiction	1953	26.34	23
156	Synergistic grid-enabled website	Brandon Black	Fiction	1953	31.68	34
243	Automated systemic toolset	Tiffany Conley	Fantasy	1953	8.87	65
457	Configurable disintermediate extranet	Melissa Lewis	Mystery	1953	28.22	2
167	User-friendly radical standardization	Leon Davis	Science Fiction	1954	36.02	55
193	Customer-focused tertiary methodology	Justin Garcia	Fantasy	1954	29.54	100
242	Business-focused responsive parallelism	Amy Reyes	Mystery	1954	33.79	38
37	Up-sized tertiary archive	Todd Kennedy	Fantasy	1955	13.08	3
49	Robust attitude-oriented attitude	Zachary Hayes	Biography	1955	49.50	15

ALL GENRES AVAILABLE IN THE BOOKS TABLE

```
select distinct genre from books;
```

Result Grid

genre
Biography
Fantasy
Non-Fiction
Fiction
Romance
Science Fiction
Mystery

RETRIEVE THE TOTAL NUMBER OF BOOKS SOLD FOR EACH GENRE

```sql
select books.genre, sum(orders.quantity) as total_books
from books
join orders
on books.book_id=orders.book_id
group by books.genre;
```

Result Grid | Filter Rows:

genre	total_books
Fiction	225
Biography	285
Non-Fiction	336
Romance	434
Fantasy	446
Science Fiction	447
Mystery	487

FIND THE AVERAGE PRICE OF BOOKS IN THE "FANTASY" GENRE

```
select avg(price) from books where genre='fantasy';
```

Result Grid

	avg(price)
▶	25.981690

LIST CUSTOMERS WHO HAVE PLACED AT LEAST 2 ORDERS

```sql
select CUSTOMER_ID, COUNT(ORDER_ID) AS ORDER_COUNT
FROM ORDERS
GROUP BY CUSTOMER_ID
HAVING COUNT(ORDER_ID)>=2;
```

Result Grid | Filter Rows:

CUSTOMER_ID	ORDER_COUNT
84	2
137	2
216	2
14	2
109	2
131	2
454	2
377	2

FIND THE MOST FREQUENTLY ORDERED BOOK

```
SELECT BOOKS.TITLE, SUM(ORDERS.QUANTITY) AS TOTALQTY
FROM ORDERS
JOIN BOOKS
ON ORDERS.BOOK_ID= BOOKS.BOOK_ID
GROUP BY BOOKS.TITLE
ORDER BY TOTALQTY DESC
LIMIT 1;
```

Result Grid | Filter Rows:

TITLE	TOTALQTY
Realigned multi-tasking installation	28

SHOW THE TOP 3 MOST EXPENSIVE BOOKS OF 'FANTASY' GENRE

```sql
SELECT TITLE , BOOK_ID
FROM books
WHERE GENRE ='FANTASY'
ORDER BY PRICE DESC
LIMIT 3;
```

Result Grid | Filter Rows:

TITLE	BOOK_ID
Stand-alone content-based hub	240
Innovative 3rdgeneration database	462
Optimized even-keeled analyzer	238

RETRIEVE THE TOTAL QUANTITY OF BOOKS SOLD BY EACH AUTHOR

```
SELECT BOOKS.AUTHOR, SUM(ORDERS.QUANTITY) AS TOTALQTY
FROM BOOKS
JOIN ORDERS
ON BOOKS.BOOK_ID=ORDERS.BOOK_ID
GROUP BY BOOKS.AUTHOR;
```

Result Grid | Filter Rows:

AUTHOR	TOTALQTY
Margaret Moore	8
John Davidson	13
Christopher Fuentes	6
Marissa Smith	16
Christopher Dixon	15
Tonya Saunders	21
Larry Hunt	6
Brandon Foster	4

LIST THE CITIES WHERE CUSTOMERS WHO SPENT OVER $30 ARE LOCATED

```
SELECT DISTINCT CUSTOMERS.CITY, TOTAL_AMOUNT
FROM CUSTOMERS
JOIN ORDERS
ON CUSTOMERS.Customer_ID = ORDERS.Customer_ID
WHERE TOTAL_AMOUNT>30;
```

Result Grid | Filter Rows: TAYLC

CITY	TOTAL_AMOUNT
Taylorfurt	189.45
New Taylorstad	221.68
Taylorfurt	100.96

FIND THE CUSTOMER WHO SPENT THE MOST ON ORDERS

```sql
SELECT CUSTOMERS.NAME ,  SUM(ORDERS.TOTAL_AMOUNT) AS AMT_SPENT
FROM CUSTOMERS
JOIN ORDERS
ON CUSTOMERS.Customer_ID= ORDERS.Customer_ID
GROUP BY CUSTOMERS.NAME
ORDER BY SUM(ORDERS.TOTAL_AMOUNT) DESC
LIMIT 1;
```

Result Grid | Filter Rov

	NAME	AMT_SPENT
▶	Kim Turner	1398.90

THANK YOU FOR VIEWING MY PROJECT!