LIBRARY MANAGEMENT SYSTEM

MYSQL PROJECT

Presented by: Akanksha Pawar

OBJECTIVE

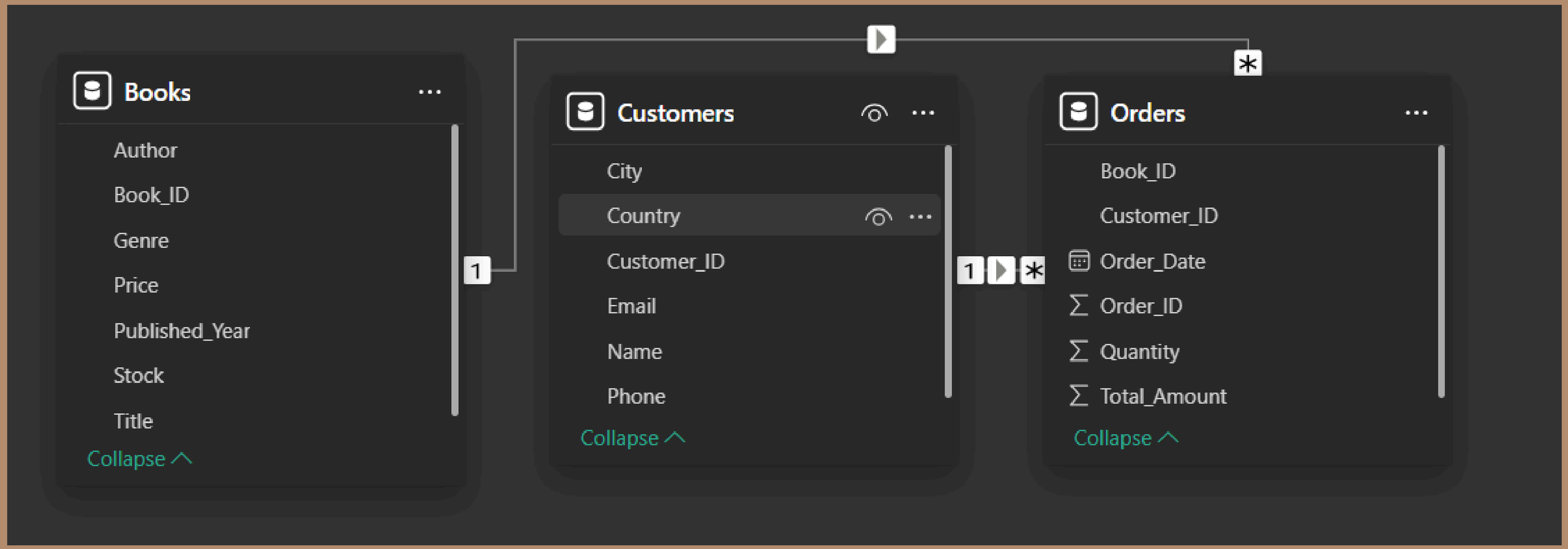
THIS SQL PROJECT AIMS TO ENHANCE DATABASE QUERYING SKILLS BY PERFORMING OPERATIONS USING VARIOUS SQL FUNCTIONS, INCLUDING AGGREGATE FUNCTIONS, JOINS, SUBQUERIES, STRING FUNCTIONS, AND DATETIME FUNCTIONS. THROUGH HANDS-ON EXECUTION OF THESE QUERIES, THE PROJECT EXPLORES DATA MANIPULATION, RETRIEVAL, AND TRANSFORMATION TECHNIQUES. THE OBJECTIVE IS TO DEVELOP A STRONG UNDERSTANDING OF SQL CONCEPTS AND THEIR PRACTICAL APPLICATIONS IN DATA ANALYSIS.

DATABASE DESCRIPTION

Tables Overview:

- Books Table: Contains details about books, including book name, price, and stock availability.
- Customers Table: Stores customer information such as customer name, city, and contact details.
- Orders Table: Records order details, including order date, quantity, and price.

MODEL DIAGRAM

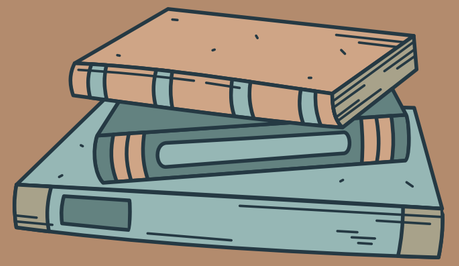


1) RETRIEVE ALL BOOKS IN THE "FICTION" GENRE

```
SELECT * from
Books WHERE genre = 'Fiction' ;
```

2) FIND BOOKS PUBLISHED AFTER THE YEAR 1950

```
SELECT * from
Books WHERE published_year > 1950;
```

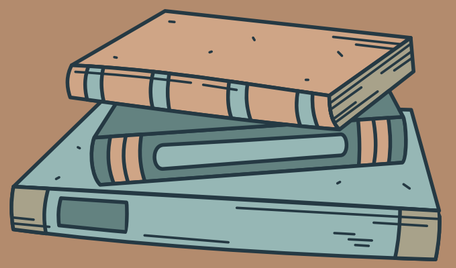


3)LIST ALL CUSTOMERS FROM THE CANADA

```
SELECT * from
Customers WHERE country  = 'Canada';
```

4) RETRIEVE THE TOTAL STOCK OF BOOKS AVAILABLE

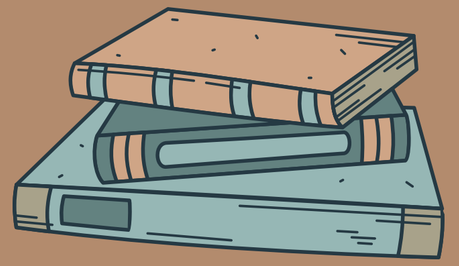
```
SELECT COUNT(stock)
as totalStock from books;
```



5) SHOW ORDERS PLACED IN NOVEMBER 2023 (USING BETWEEN OPERATOR AND YEAR,MONTH FUNCTION)

```
SELECT * from orders
WHERE Order_date BETWEEN '2023-11-01'
AND  '2023-11-30' ;;
```

```
SELECT * from orders WHERE
YEAR(Order_date)= 2023 AND
MONTH(Order_date)= 11;
```

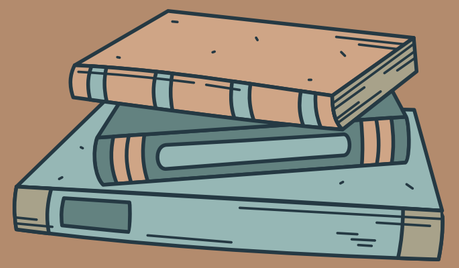


6)FIND THE DETAILS OF THE MOST EXPENSIVE BOOK

```
SELECT * from books WHERE price = (select
MAX(price) from books);
```

7)RETRIEVE ALL ORDERS WHERE THE TOTAL AMOUNT EXCEEDS $20

```
SELECT * from order
WHERE total_amount > 20;
```

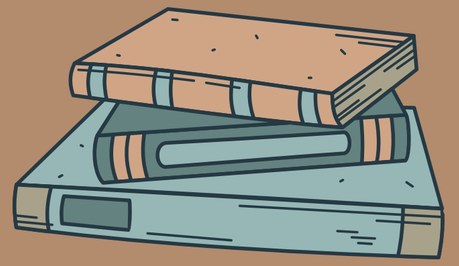


8)LIST ALL GENRES AVAILABLE IN THE BOOKS TABLE

```
SELECT  Distinct(genre) from books;
```

9)FIND THE BOOK WITH THE LOWEST STOCK

```
SELECT * from books
WHERE stock = (SELECT
MIN(stock) from books);
```

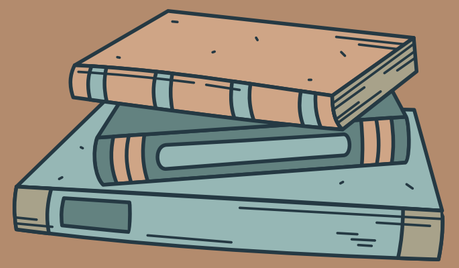


10)CALCULATE THE TOTAL REVENUE GENERATED FROM ALL ORDERS

```
SELECT  SUM(total_amount) AS
total_revenue from orders;
```

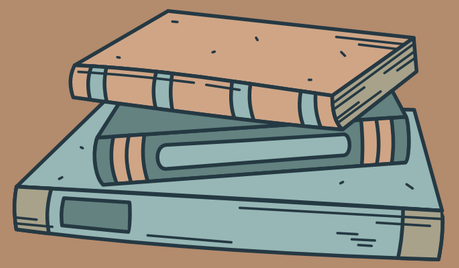
11)FIND THE AVERAGE PRICE OF BOOKS IN THE "FANTASY" GENRE

```
SELECT genre, AVG(Price) AS
Average_Price from books
WHERE  genre = 'Fantasy';
```



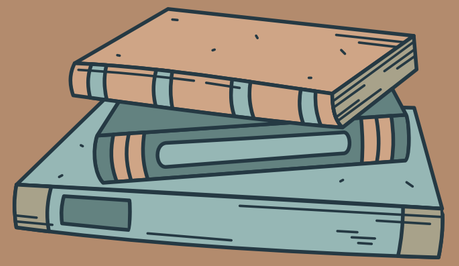
12)SELECT SUM(TOTAL_AMOUNT) AS TOTAL_REVENUE FROM ORDERS;

```
SELECT c.Customer_id , c.name ,
o.order_id,o.book_id ,o.order_date , o.quantity
FROM customers as c join orders AS o
ON c.customer_id = o.customer_id
WHERE  quantity >1;
```



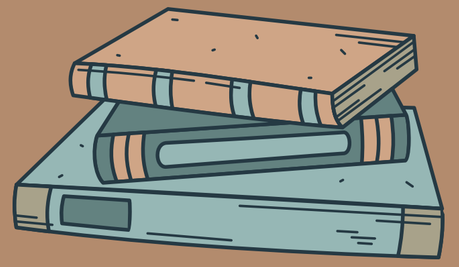
13)RETRIEVE THE TOTAL NUMBER OF BOOKS SOLD FOR EACH GENRE

```
SELECT b.genre , SUM(o.quantity) AS
total_books_sold FROM books as b JOIN orders as o
 ON b.book_id = o.book_id
 GROUP BY genre
ORDER BY total_books_sold DESC;
```



14)FIND THE MOST FREQUENTLY ORDERED BOOK

```
SELECT b.title , COUNT(order_id) AS order_count
FROM books b JOIN orders o
 ON b.book_id = o.book_id
 GROUP BY o.book_id ,b.title
 ORDER BY order_count DESC  LIMIT 1;
```

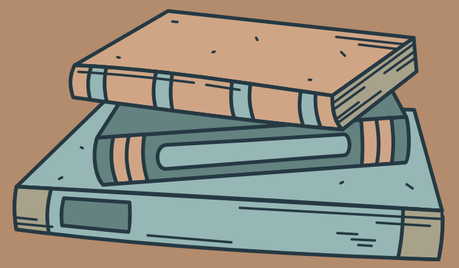


15)LIST CUSTOMERS WHO HAVE PLACED AT LEAST 2 ORDERS

```
SELECT Customer_id ,COUNT(order_id) AS order_count
FROM orders GROUP BY customer_id HAVING
COUNT(order_id) >=2;
```

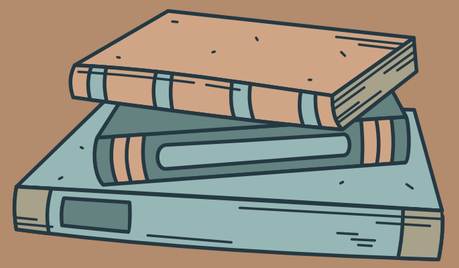
16)SHOW THE TOP 3 MOST EXPENSIVE BOOKS OF 'FANTASY' GENRE

```
SELECT book_id, title, Price  FROM books
WHERE genre = 'Fantasy'
ORDER BY price DESC LIMIT 3 ;
```



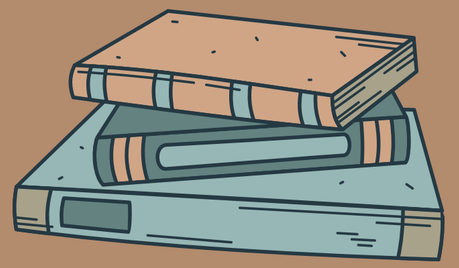
17)RETRIEVE THE TOTAL QUANTITY OF BOOKS SOLD BY EACH AUTHOR

```
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 b.author ;
```



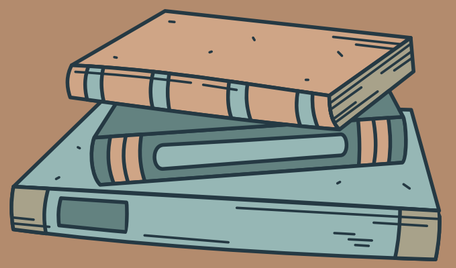
18)LIST THE CITIES WHERE CUSTOMERS WHO SPENT OVER $3
ARE LOCATED

```
SELECT c.customer_id , c.city,c.name ,o.total_amount
FROM customers c JOIN orders o
ON c.customer_id= o.customer_id
WHERE o.total_a
ORDER BY o.total_amount ;
```



19)FIND THE CUSTOMER WHO SPENT THE MOST ON ORDERS

```
SELECT c.customer_id,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.customer_id,c.name
ORDER BY total_spent DESC LIMIT 1;
```



CONCLUSION

- GAINED HANDS-ON EXPERIENCE IN SQL QUERYING FOR DATA MANIPULATION, RETRIEVAL, AND TRANSFORMATION.
- APPLIED VARIOUS SQL FUNCTIONS, INCLUDING AGGREGATE FUNCTIONS, JOINS, SUBQUERIES, STRING FUNCTIONS, AND DATETIME FUNCTIONS.
- EXPLORED DATA RELATIONSHIPS ACROSS THE BOOKS, CUSTOMERS, AND ORDERS TABLES TO EXTRACT MEANINGFUL INSIGHTS.
- STRENGTHENED UNDERSTANDING OF SQL CONCEPTS AND THEIR REAL-WORLD APPLICATIONS IN DATA ANALYSIS.
- DEVELOPED SKILLS IN EFFICIENT DATABASE MANAGEMENT AND QUERY OPTIMIZATION..