

GROUP PROJECT

Presented by Group 9

OUR TEAM

1. Tecla Gatwiri 222203
2. Daniella Awuor 222202
3. Mumtaz Yussuf 220386
4. Sylvia Waithira Muigai 181038
5. Chelsea Lorna 220734

TASKS

1. Database Design and Setup
2. Java Application Development
3. Statistical Analysis
4. Documentation & Presentation

DATABASE DESIGN AND SETUP

1 • `SELECT * FROM library_borrowing.members;`

Result Grid			
member_id	first_name	last_name	email
M001	James	Smith	james1@library.com
M002	David	Wei	david2@library.com
M003	Lucy	Ndegwa	lucy3@library.com
M004	Chen	Wei	chen4@library.com
M005	Lucy	Ali	lucy5@library.com
M006	Sam	Wei	sam6@library.com
M007	Brian	Smith	brian7@library.com
M008	Omar	Achieng	omar8@library.com
M009	Carlos	Mworia	carlos9@library.com
M010	Fatima	Mworia	fatima10@library.com
M011	Ali	Achieng	ali11@library.com
M012	Mary	Ali	mary12@library.com
M013	James	Murithi	james13@library.com
M014	Omar	Mugambi	omar14@library.com
M015	Carlos	Ndegwa	carlos15@library.com
M016	Sam	Mwani	sam16@library.com

We find out that the SELECT is the SQL command used to retrieve data from a database.

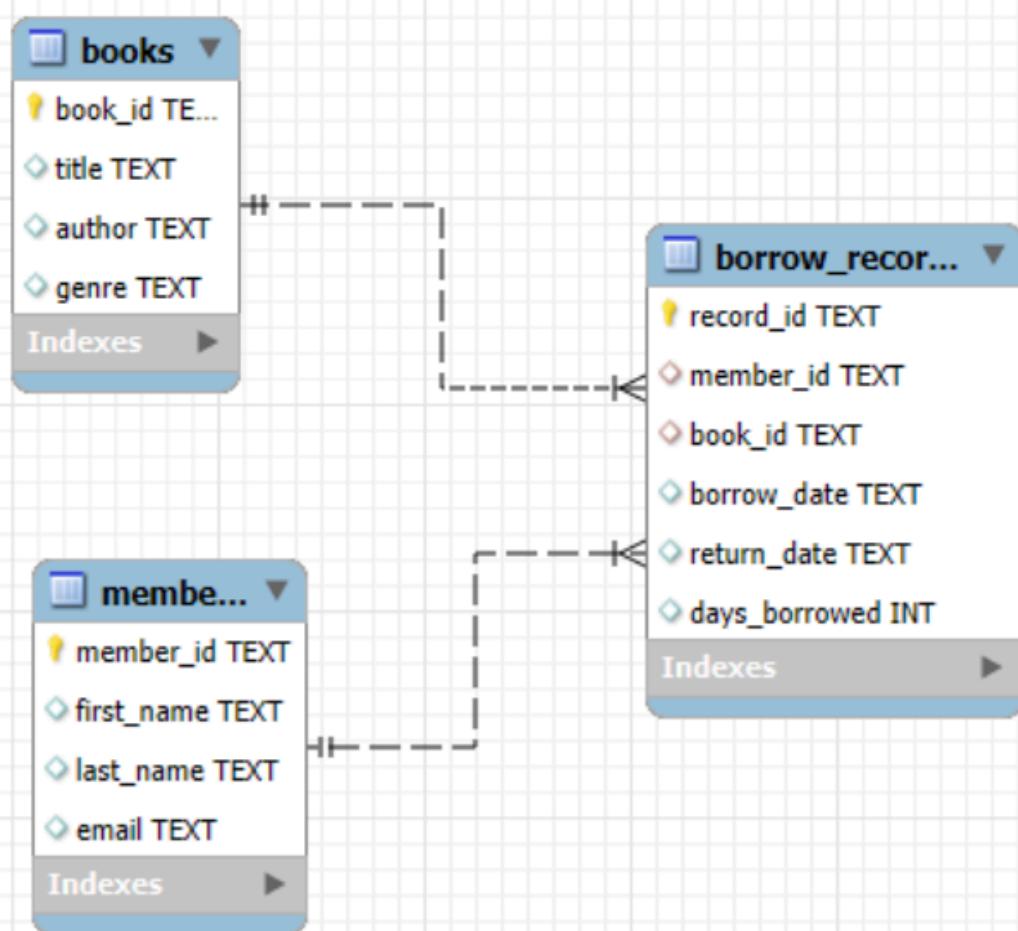
The asterisk is a wildcard that means “all columns.” So this part of the command is saying, “Retrieve every column from the table.”

FROM is a clause that specifies which table to get the data from.
`library_borrowing.members`: This is the full table name, which has two parts:

- `library_borrowing`: This is likely the database name or schema name
- `members`: This is the actual table name that contains the member data

We can therefore conclude that the entire statement
`SELECT * FROM library_borrowing.members;`

Shows all columns and all rows from the members table in the `library_borrowing` database.

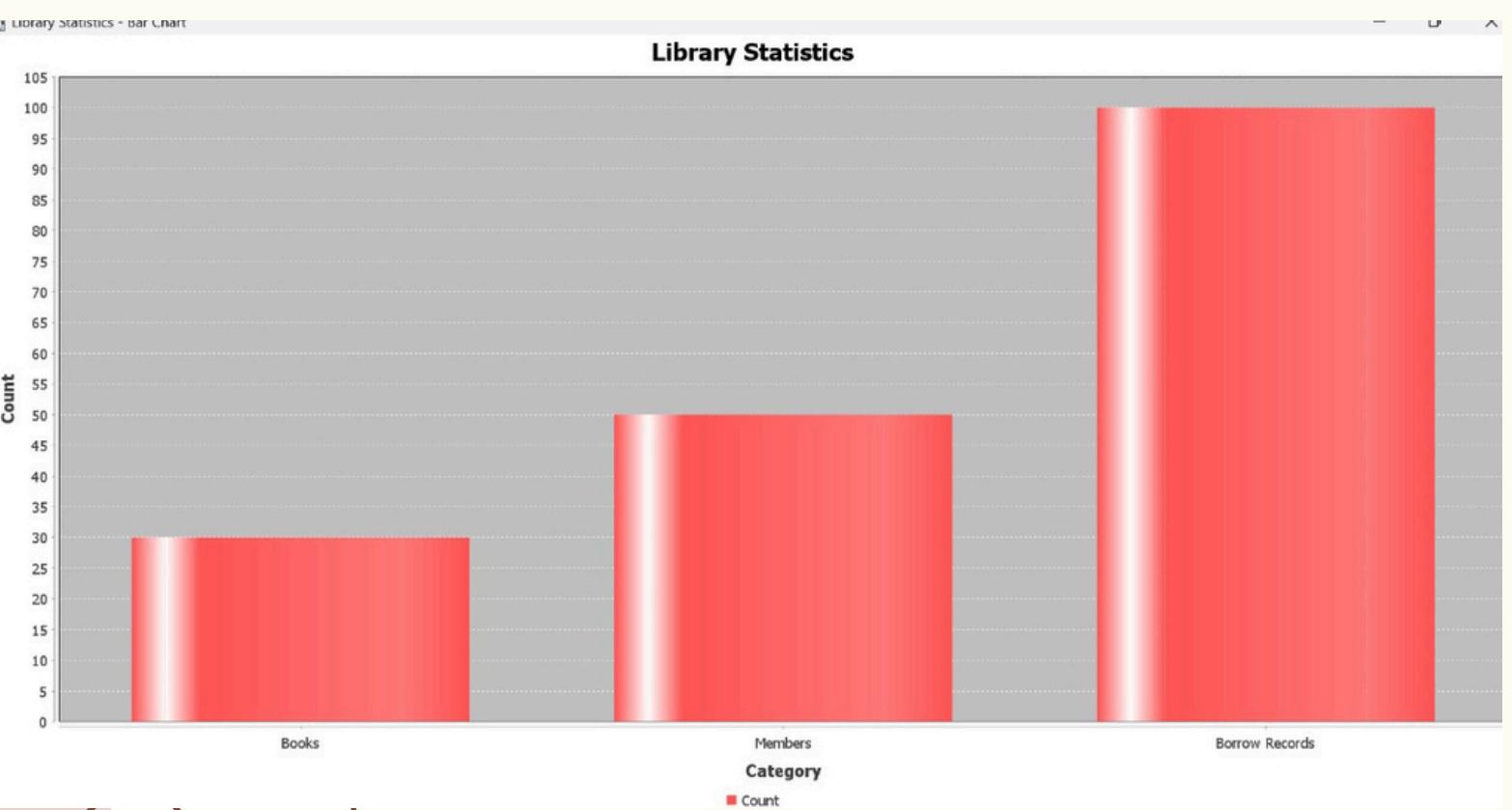


From the ER diagram we can conclude that it features three main tables:

- books: Stores information about the books in the library.
 Primary Key : book_id
 Attributes: title , author , genre
- members: Stores information about the library's members.
 Primary Key : member_id
 Attributes: first_name , last_name, email .
- borrow_records: A junction table that records each instance of a book being borrowed.
 Primary Key : record_id
 Foreign Keys : member_id (referencing members), book_id (referencing books).
 Attributes: borrow_date , return_date , days_borrowed .

Relationship	Entities	Type	Meaning
Member to Borrow Record	members and borrow_records	One-to-Many (1:N)	<p>A single member can have multiple borrow records, but each borrow record belongs to exactly one member.</p>
Book to Borrow Record	books and borrow_records	One-to-Many (1:N)	<p>A single book can appear in multiple borrow records (i.e., be borrowed multiple times), but each borrow record is for only one book.</p>
Implicit Member to Book	members and books	Many-to-Many (N:M)	<p>A member can borrow many books, and a book can be borrowed by many members. This relationship is resolved by the borrow_records junction table.</p>

JAVA APPLICATION DEVELOPMENT



The bar chart pulls three values from the database using the DataAnalyzer class:

`getTotalBooks()`

`getTotalMembers()`

`getTotalBorrowRecords()`

The values are inserted into a DefaultCategoryDataset. JFreeChart's `createBarChart()` method generates the visualization.

A Swing JFrame displays the chart.

Interpretation

Books: Indicates the size of the library's collection.

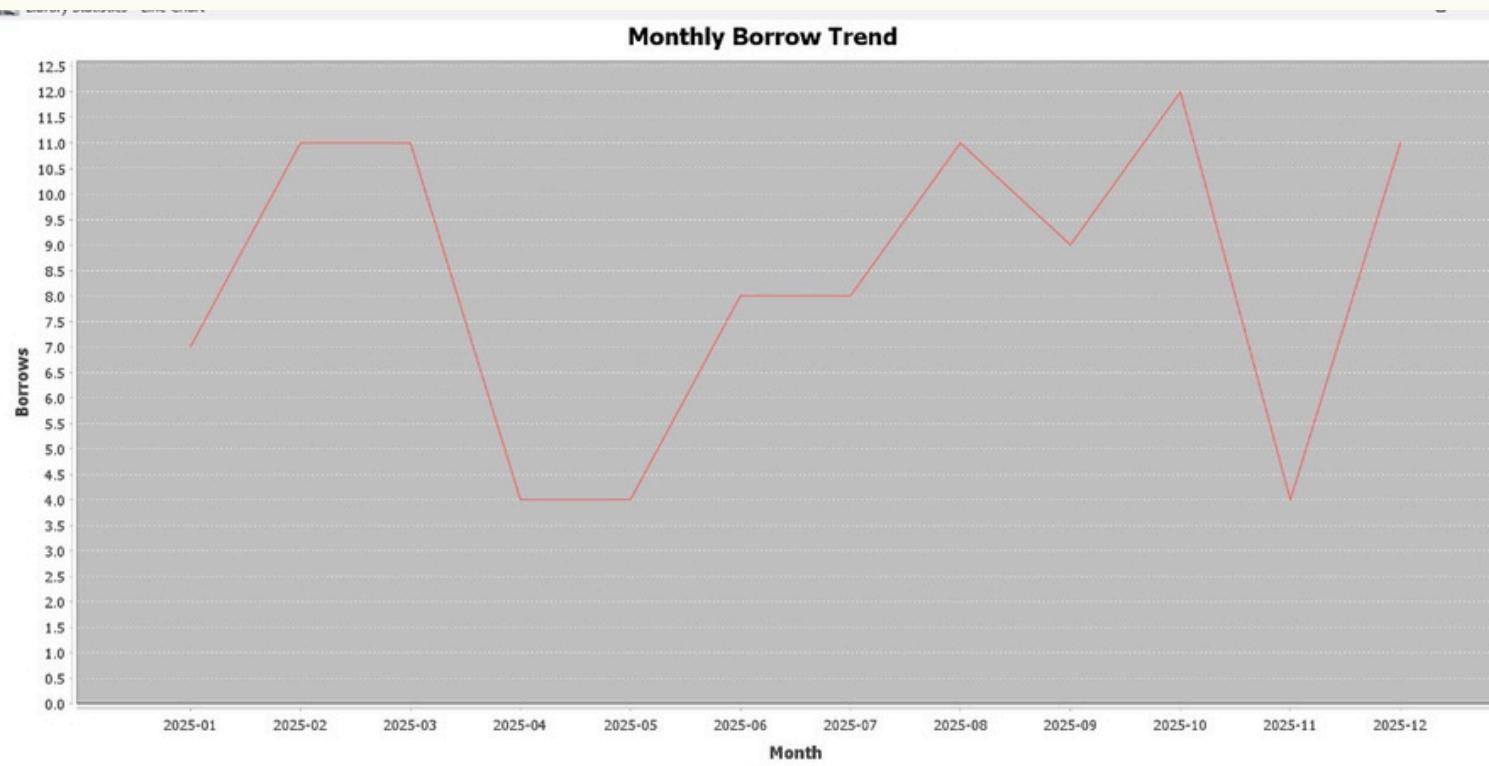
Members: Reflects the number of registered library users.

Borrow Record: Shows user engagement with the library.

Purpose

To analyze changes in borrowing activity over time, showing trends across months.

How It Was Generated



`DataAnalyzer.getMonthlyBorrowTrend()` runs a SQL query that groups borrow records by month.

Results are added into a `DefaultCategoryDataset`.
`JFreeChart's createLineChart()` function generates the trend visualization.

Interpretation

A rising line indicates increasing borrowing activity in certain months.

A falling line suggests seasonal declines or inactive periods.
This chart allows the library to see peak months and low-traffic periods.

The visualization component significantly enhances the interpretability of the raw data and transforms it into meaningful insights for strategic planning.

Purpose:

To highlight which book is the most borrowed and compare it against all other books.

How It Was Generated:

The DataAnalyzer.getMostBorrowedBook() method identifies the top book using an SQL GROUP BY + COUNT(*) query.

The borrow count of the top book is extracted from the returned string.

A pie chart dataset is created:

Slice 1: Borrow count of the most borrowed book

Slice 2: Total remaining borrow records

JFreeChart's createPieChart() generates the visualization.

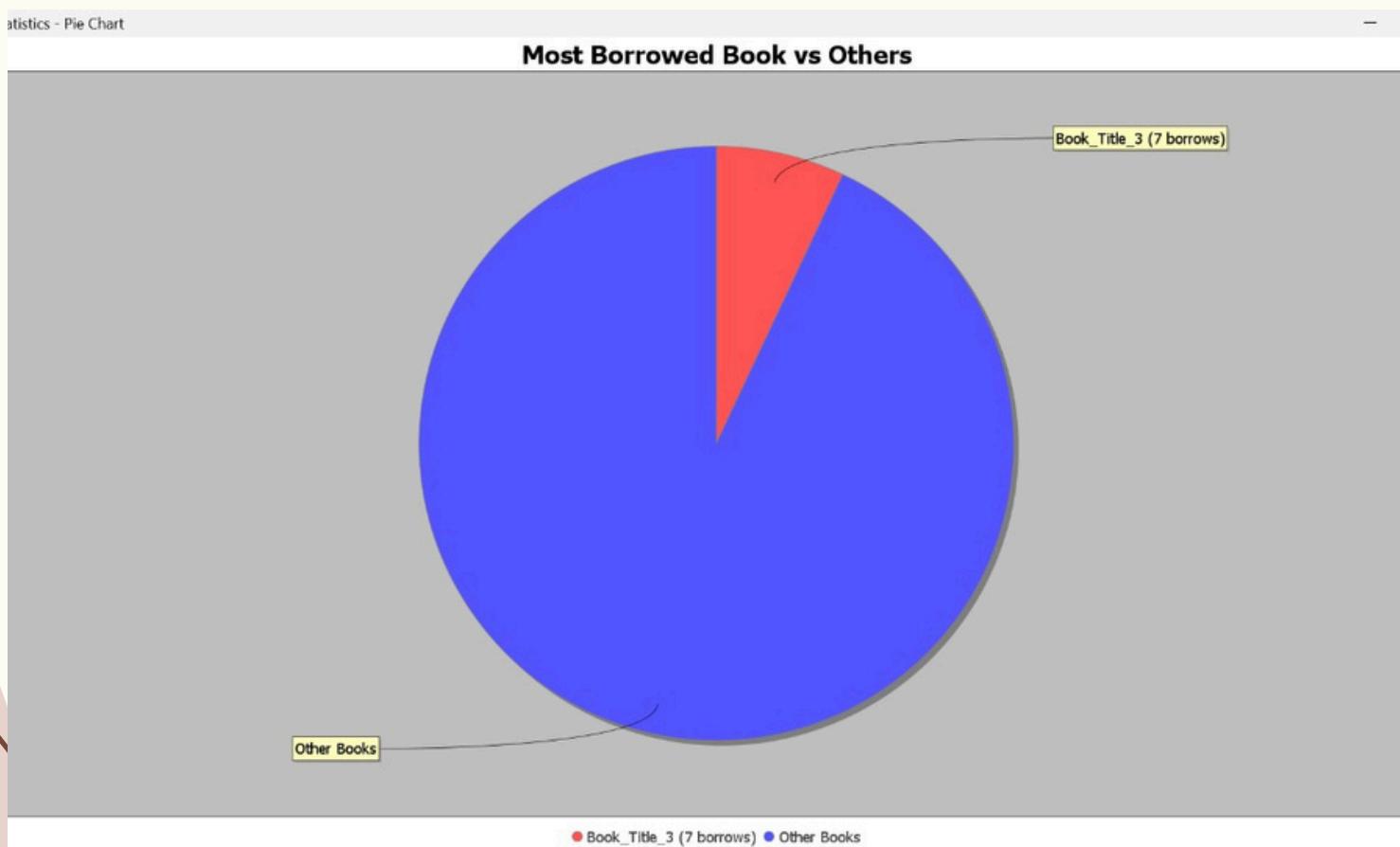
Interpretation

The chart illustrates how dominant the top-performing book is.

A large slice for "Other Books" suggests borrowing is spread across multiple titles.

A large slice for the most borrowed book means one book is significantly more popular.

This visualization is key for understanding user preferences and library resource demand.



Statistical Analysis

SQL Queries Used and interpretations

SELECT

```
b.title AS Book_Title,  
COUNT(br.book_id) AS Times_Borrowed  
FROM borrow_records AS br  
JOIN books AS b  
ON br.book_id = b.book_id  
GROUP BY b.title  
ORDER BY Times_Borrowed DESC  
LIMIT 10;
```

C	D
Book_Title	Times_Borrowed
Book_Title_3	7
Book_Title_7	6
Book_Title_15	6
Book_Title_11	5
Book_Title_29	5
Book_Title_28	5
Book_Title_23	5
Book_Title_25	5
Book_Title_10	5
Book_Title_17	4

a) Most borrowed books

The query counts how many times each book was borrowed and lists the top 10.

Book title 3 was the most borrowed book indicating stronger user interest in it.

	A	B	C
1	Member_Name	Total_Borrowed	
2	Lucy Khan	5	
3	Omar Ali	5	
4	Sam Wei	5	
5	Carlos Mworia	5	
6	Mary Ali	4	
7	Tom Mwangi	3	
8	Chen Mworia	3	
9	Grace Mugambi	3	
10	Brian Ali	3	
11	Aisha Murithi	3	

SELECT

```
CONCAT(m.first_name, ' ', m.last_name) AS Member_Name,
COUNT(*) AS Total_Borrowed
FROM borrow_records br
JOIN members m ON br.member_id = m.member_id
GROUP BY Member_Name
ORDER BY Total_Borrowed DESC
LIMIT 10;
```

b) Most active members

This query identifies members who borrowed the most books with the highest having borrowed five books.
It suggests that the members are frequently library users.

```
SELECT
```

```
    ROUND(AVG(days_borrowed), 2) AS Avg_Borrow_Duration
```

```
FROM borrow_records;
```

```
SELECT
```

```
    COUNT(*) AS Overdue_Count
```

```
FROM borrow_records
```

```
WHERE days_borrowed > 14;
```

A
Avg_Borrow_Duration
16.58

A	B
Overdue_Count	55

c) Average Borrowing Duration and Overdues

The above queries produced results that the average borrowing duration is approximately 16.58 days with 55 overdue records, whereby the borrowing duration was set at a maximum of 14 days.

```
SELECT  
    b.genre AS Genre,  
    COUNT(*) AS Borrow_Count  
FROM borrow_records br  
JOIN books b ON br.book_id = b.book_id  
GROUP BY b.genre  
ORDER BY Borrow_Count DESC;
```

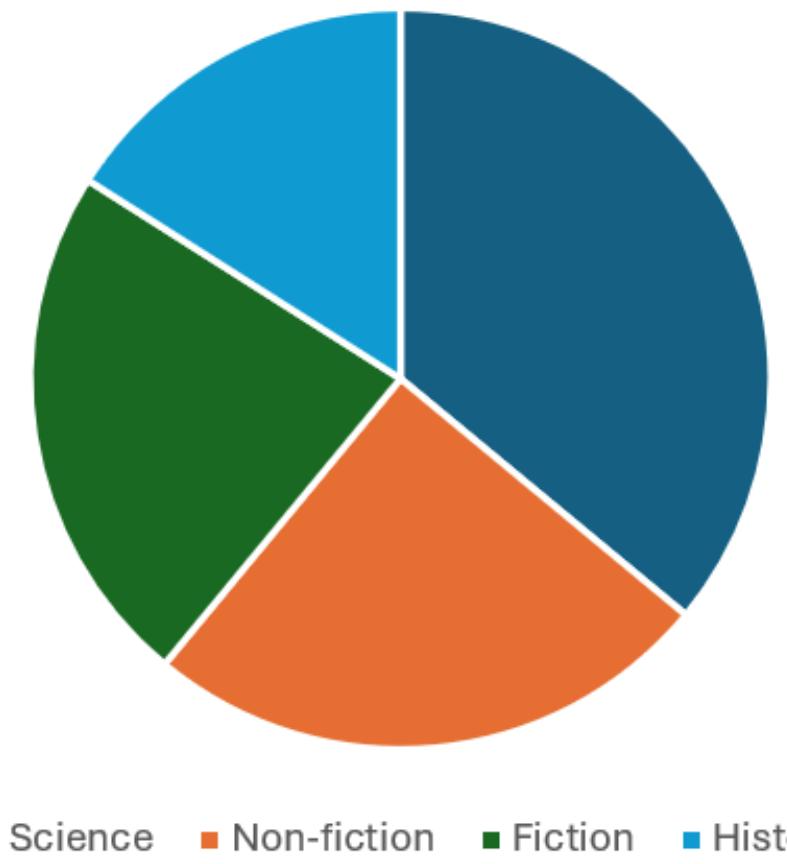
	A	B	C
1	Genre	Borrow_Count	
2	Science	36	
3	Non-fiction	25	
4	Fiction	23	
5	History	16	
6			

d. Genre Popularity

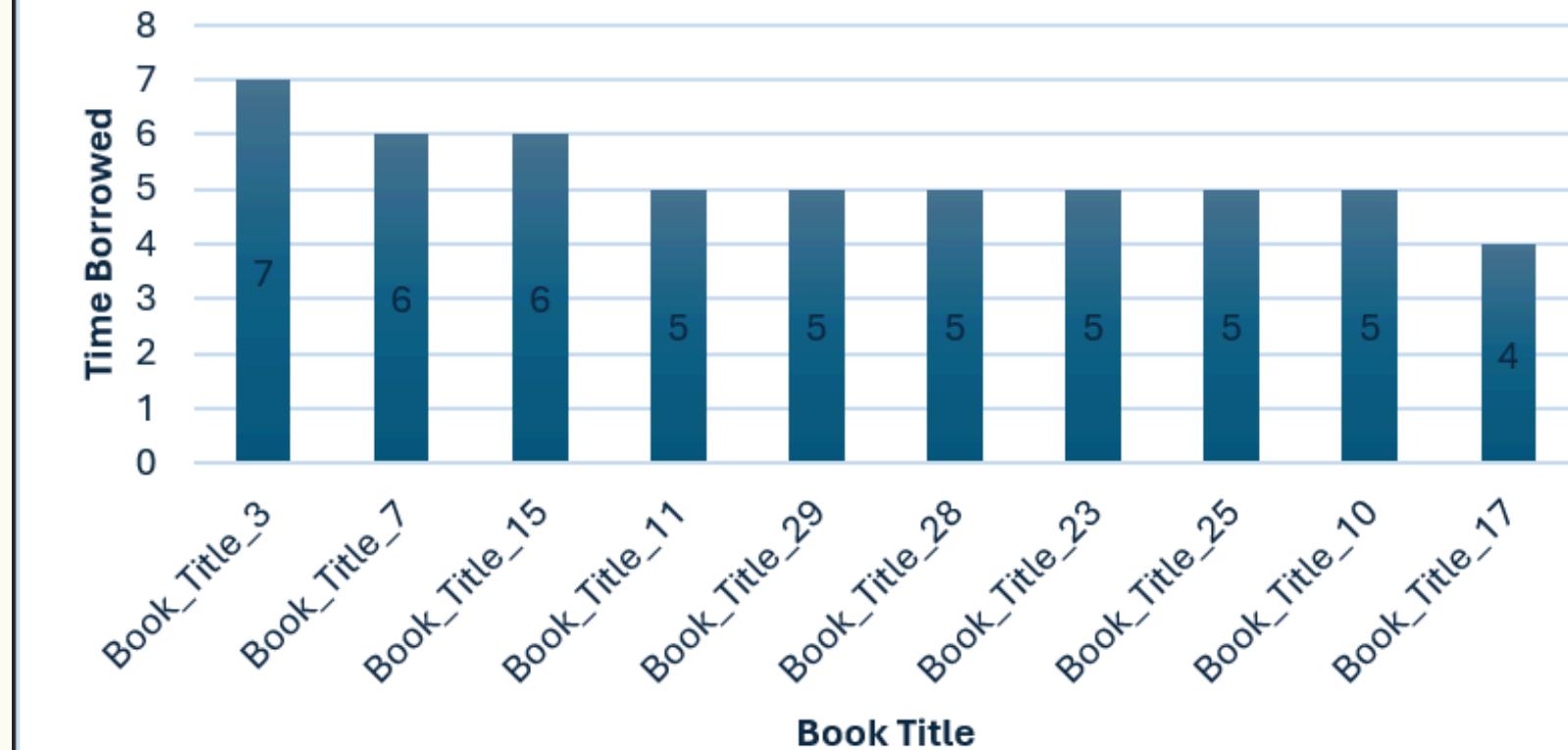
From the analysis, Science genre books were borrowed most.

e) Visualizations

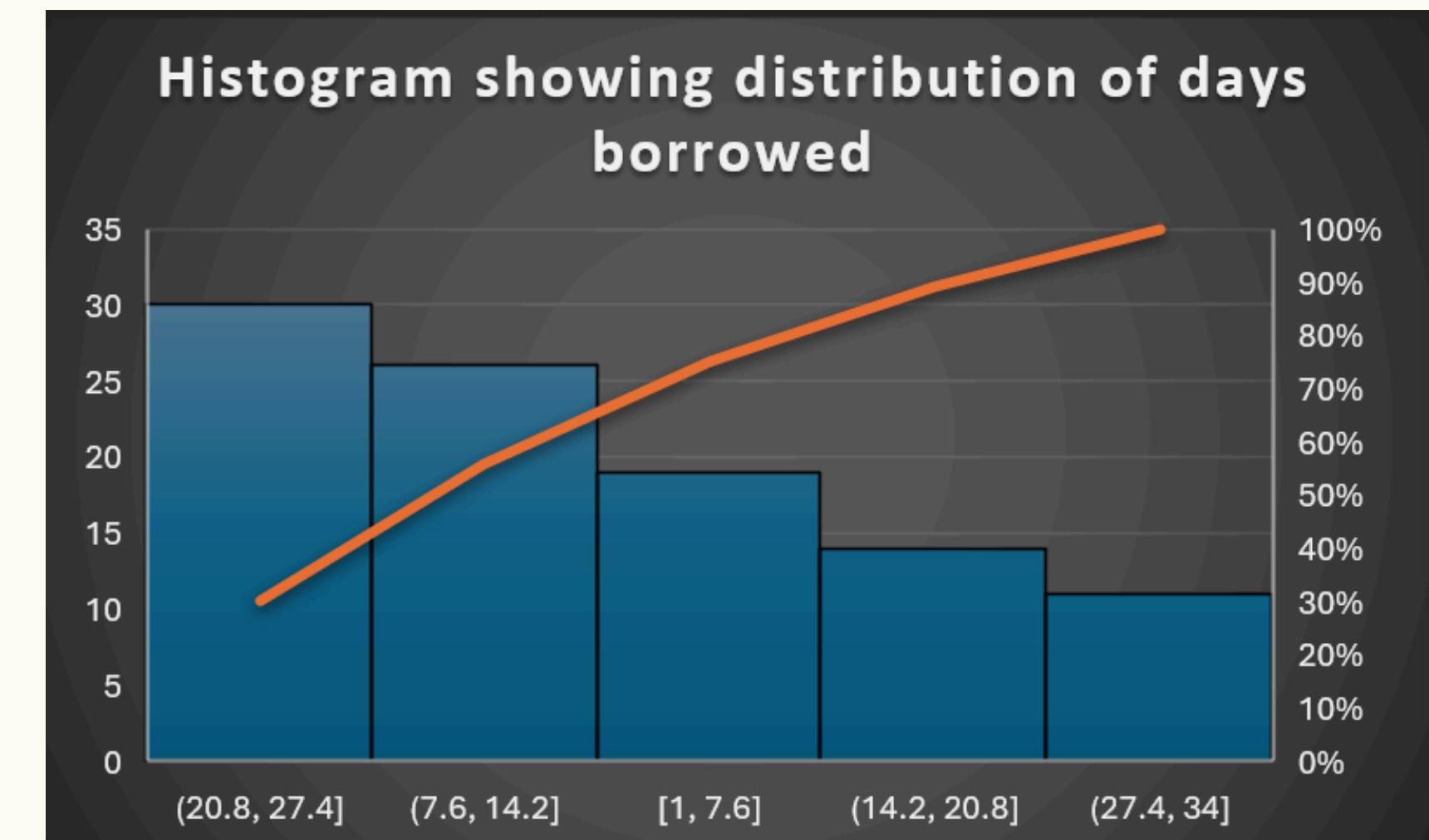
Pie chart showing Genre popularity



Bar graph of top borrowed books



Histogram showing distribution of days borrowed



Challenges

We experienced a challenge in collaborating using GitHub and learning how to use MySQL but after team work and research we were able to work our way out.

Github link : <https://github.com/mumtaz-data/DataAnalysisApp>

THANK
YOU