Library Management System - SQL Project Report

Project Title

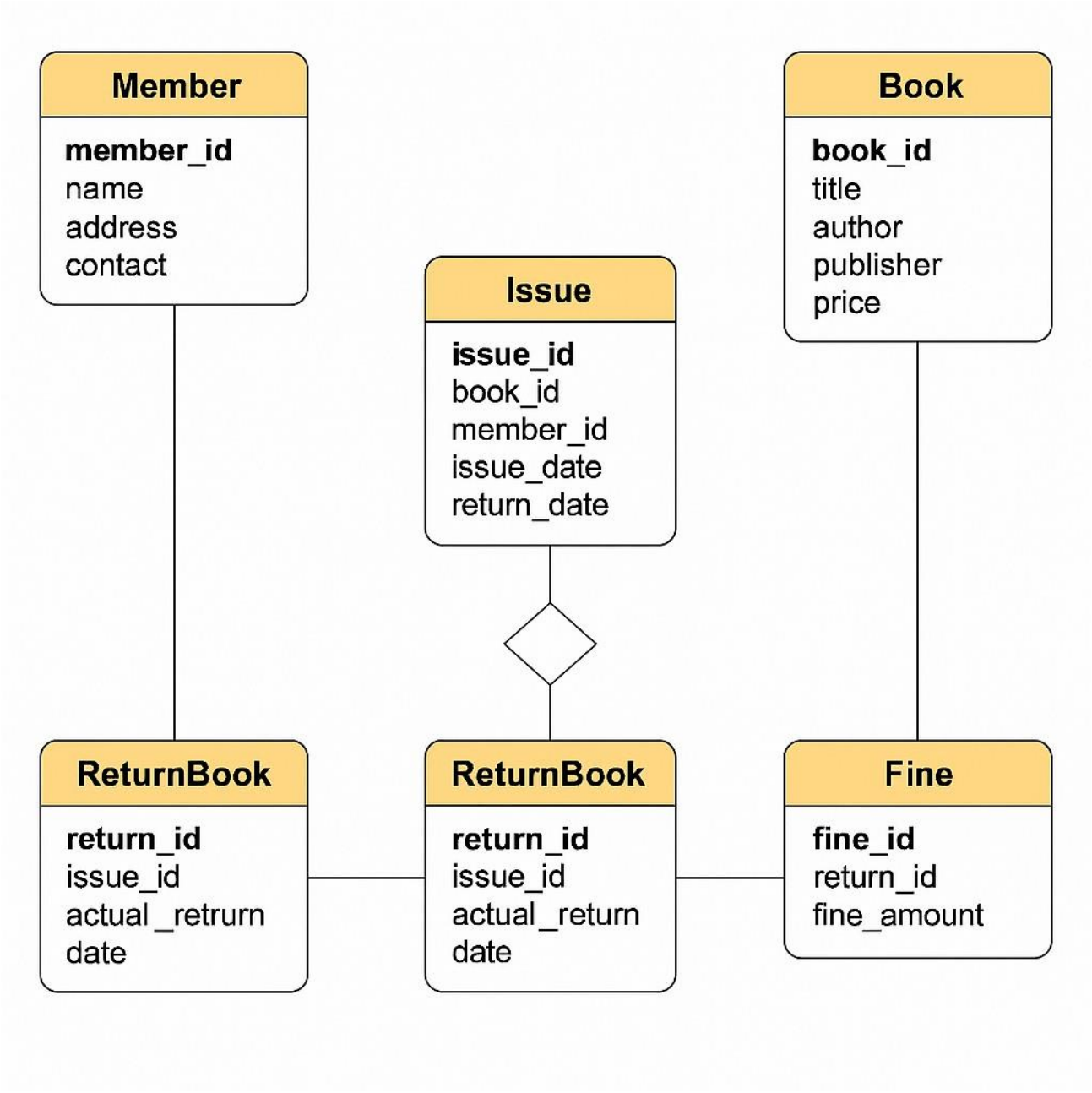
Library Management System using MySQL

Project Description

This project is a database-driven Library Management System developed using MySQL. It is designed to manage essential library operations such as managing members, books, issuing and returning books, staff details, and calculating fines on late returns. The system is entirely SQL-based and demonstrates real-time database design, CRUD operations, and query execution using joins and aggregate functions.

Entity Relationship (ER) Diagram

Library Management System - SQL Project Report



Technologies Used

- MySQL (SQL queries)

- MySQL Command Line Client / MySQL Workbench

Database Details

Database Name: LibraryDB

Library Management System - SQL Project Report

Tables Used:

- Member - Stores member details
- Book -Stores book details
- Staff -Stores staff details
- Issue - Stores issue transactions
- ReturnBook - Stores actual return details
- Fine - Stores fine amount for late returns

Features &Functionality

1. CRUD OPERATIONS

   - INSERT: Add new members, books, staff, issues, returns, and fines.
   - UPDATE: Modify book price or member address.
   - DELETE: Remove staff records and fine entries.

2. SELECT QUERIES

   - Retrieve data using SELECT * FROM table
   - Specific condition-based retrieval

3.JOINQUERIES

   SQL:

```
SELECT M.name AS Member_Name,B.title AS Book_Title, I.issue_date
FROM Member M
JOIN Issue I ON M.member_id  =I.member_id
JOIN Book B ON I.book_id = B.book_id;
```

4. AGGREGATE FUNCTIONS

   - COUNT(): Total number of books
   - MAX(): Highest book price
   - MIN(): Lowest book price

- AVG(): Average price of books
- SUM(): Total value of all books

5.FINECALCULATIONLOGIC

SQL:

```
SELECT R.return_id, I.return_date, R.actual_return_date,
DATEDIFF(R.actual_return_date,I.return_date)ASDays_Late, CASE
  WHENDATEDIFF(R.actual_return_date,I.return_date)>0THEN
     DATEDIFF(R.actual_return_date, I.return_date) * 10
  ELSE 0
ENDASFine_Amount FROM
Issue I
JOIN ReturnBook R ON I.issue_id = R.issue_id;
```

Normalization Used

- Tables are normalized up to 3rd Normal Form (3NF)
- Separated entities: Member, Book, Staff, Issue, Return Book, Fine
- Removed data redundancy using Primary and Foreign Keys
- Ensured referential integrity and modular structure

Learning Outcomes

- Real-time Database Design for Library Use Case
- SQL Queries: SELECT, JOIN, AGGREGATE, CASE, DATE DIFF
- Implementation of CRUD operations
- Data Modeling using ER Diagrams
- Understanding of Fine Calculation Logic using SQL
- Hands-on practice with MySQL database management

Library Management System - SQL Project Report

Conclusion

This SQL project helped in gaining practical knowledge of designing and managing relational databases. It simulated real-time operations in a library system and enhanced skills in query writing, ER modeling, data normalization, and logic implementation using SQL. The experience lays a strong foundation for working with enterprise-level database systems.