Library Management System – Database Project

Student: Pushpa

Role: Intern at Elevate Labs

Tools: PostgreSQL, pgAdmin, GitHub

Date:

Abstract

The Library Management System digitizes library operations by managing books, authors, members, and loans in a normalized relational schema. The system supports many-to-many relationships between books and authors, provides views for borrowed and overdue books, and implements triggers for automating due-date assignments and overdue notifications. Analytical SQL reports such as most borrowed books and borrow trends help generate valuable insights. This project demonstrates the end-to-end design and implementation of a real-world relational database system.

Objectives

- Design normalized schema for books, authors, members, and loans.
- Manage many-to-many relationships with a bridge table.
- Create views for borrowed and overdue books.
- Implement triggers for due-date automation and overdue notifications.
- Write analytical reports using joins and aggregate functions.

Tools Used

PostgreSQL (RDBMS), pgAdmin 4 (GUI), GitHub (version control)

Schema & Design

Entities include: members, books, authors, book_authors (bridge), loans, and notifications. Constraints and relationships enforce referential integrity. The loans table includes due_date checks. The ERD is documented in ERD.md (Mermaid format) for visualization.

Implementation Steps

1. Create database schema and tables.
2. Insert test data for books, authors, members, and loans.
3. Define many-to-many relationships with book_authors.
4. Create views: v_borrowed_book_details, v_current_loans, v_overdue_loans.
5. Implement triggers for due-date automation and overdue notifications.
6. Generate reports using joins, grouping, and aggregation.

Reports

The following reports are included in overdue_report_queries.sql:
- Borrow count per member.
- Most borrowed books.
- Overdue loans with delay days.
- Top authors by borrows.
- Monthly borrow counts.

Results & Observations

Views simplify recurring queries for dashboards and analytics. Triggers automate overdue checks and notifications, ensuring data accuracy and reducing manual intervention. The design is modular and extensible for additional features like penalties or reservations.

Conclusion

The Library Management System project demonstrates core database concepts: schema design, normalization, constraints, views, triggers, and reporting queries. It provides a strong foundation for managing real-world data-driven systems. The project also improved hands-on skills with PostgreSQL, pgAdmin, and SQL best practices, which are directly applicable in IT and data-centric roles.