Submitted By.,

Muhammad Abdullah Khan
Muhammad Usman Khattak
Muhammad Zayam Sabir

Registration Numbers.,

4747-FOC/BSSE/F23
4763-FOC/BSSE/F23
4765-FOC/BSSE/F23

Submitted To.,

Sir Shakir Rasheed Khan Khattak

Course.,

Data Base (CS 242)

Department of Software Engineering

Library Management System

Project Overview

This project successfully designed and implemented a **fully normalized PostgreSQL database** for a Library Management System using **pgAdmin 4**. The database follows **all five normal forms (1NF, 2NF, 3NF, BCNF, 4NF, 5NF)** and supports core library operations including book cataloging, member registration, book loans, returns, and fine management.

Database Schema Design

Core Entities Implemented (8 Tables)

- **Publisher** - Stores publishing companies
- **Author** - Stores book authors
- **Book** - Main book catalog with ISBN, title, price, and copy tracking
- **Book_Author** - Junction table for many-to-many book-author relationship
- **Reader** - Library members with personal information
- **Staff** - Library employees with login credentials
- **Book_Loan** - Tracks borrowing transactions and status
- **Fine** - Manages penalties for overdue books

Normalization Achievements

1NF (First Normal Form) ✓

- Eliminated composite attributes: Reader name split into first_name, middle_name, last_name
- All values are atomic (single, indivisible values)
- No repeating groups or arrays

2NF (Second Normal Form) ✓

- Single-column primary keys throughout (SERIAL auto-increment)
- No partial dependencies on composite keys
- All non-key attributes fully dependent on primary key

3NF (Third Normal Form) ✓

- **Key Improvement**: Extracted Author entity from Book table
- Eliminated transitive dependency: ISBN → Author_Name → Author_Details
- Book_Author junction table handles many-to-many relationship properly

BCNF, 4NF, 5NF Compliance ✓

- Single determinant per table (no overlapping candidate keys)
- Separate junction tables for each M:N relationship
- No multi-valued dependencies
- Lossless decomposition maintained

Sample Transactions Demonstrated

1. **Ali Khan** borrowed "*The Kite Runner*" (ISBN: 9780143034902)
    - Available copies decreased from 5 → 4
    - Loan record created with 14-day due date
2. **Sara Ahmed** borrowed "*Harry Potter*" (overdue)
    - Loan status marked 'Overdue'
    - Fine issued: PKR 120 for 6 days late
    - Available copies decreased from 10 → 9
3. **Ali returned his book**

- Loan status updated to 'Returned'
- Available copies increased back to 5

Project Objectives Achieved

✅ **Database Normalization**: Complete 1NF-5NF compliance
✅ **Referential Integrity**: All foreign key relationships enforced
✅ **Data Validation**: CHECK constraints prevent invalid data
✅ **Performance**: Indexes ensure fast queries
✅ **Usability**: Simple pgAdmin interface, ready-to-run scripts
✅ **Completeness**: Full CRUD operations demonstrated
✅ **Documentation**: Clear entity relationships and constraints

Conclusion

The Library Management System database represents a **production-ready, fully normalized relational database** that successfully demonstrates advanced database design principles. The implementation balances academic requirements (normalization proof) with practical usability for real-world library operations. All core functionalities—book cataloging, member management, loan tracking, and fine collection—are fully operational with proper data integrity and performance optimization.

GitHub Link

Library Management System