**Library Management System using SQL**

**Project Overview**

**Project Title**: Library Management System

**Level**: Intermediate

**Database**: library\_db

This project demonstrates the implementation of a Library Management System using SQL. It includes creating and managing tables, performing CRUD (Create, Read, Update, Delete) operations, and executing advanced SQL queries. The goal is to showcase skills in database design, manipulation, and querying.

**Objectives**

1. **Set up the Library Management System Database**: Create and populate the database with tables for branches, employees, members, books, issued status, and return status.
2. **CRUD Operations**: Perform Create, Read, Update, and Delete operations on the data.
3. **CTAS (Create Table As Select)**: Utilize CTAS to create new tables based on query results.
4. **Advanced SQL Queries**: Develop complex queries to analyze and retrieve specific data.

**Project Structure**

**1. Database Setup**

* **Database Creation**: A database named library\_db was created.
* **Table Creation**: Tables were created for branches, employees, members, books, issued status, and return status. Each table includes relevant columns and relationships to form a coherent database schema.

**2. Business Tasks & Questions Addressed**

The following tasks were performed and questions were answered using SQL:

**CRUD Operations**

* **Task 1**: Create a new book record for 'To Kill a Mockingbird'.
* **Task 2**: Update an existing member's address.
* **Task 3**: Delete a specific record from the issued status table.
* **Task 4**: Retrieve all books issued by a specific employee.
* **Task 5**: List members who have issued more than one book.

**CTAS (Create Table As Select)**

* **Task 6**: Create a summary table (book\_issued\_cnt) showing the total issue count for each book.

**Data Analysis**

* **Task 7**: Retrieve all books in a specific category.
* **Task 8**: Find the total rental income generated by each book category.
* **Task 9**: List members who registered in the last 180 days.
* **Task 10**: List employees along with their branch details and their manager's name.
* **Task 11**: Create a new table (expensive\_books) for books with a rental price above a certain threshold.
* **Task 12**: Retrieve the list of all books that have been issued but not yet returned.

**Advanced SQL Operations**

* **Task 13**: Identify members with overdue books (assuming a 30-day return period) and calculate the days overdue.
* **Task 14**: Develop a procedure to automatically update a book's availability status to "Yes" when it is returned.
* **Task 15**: Create a branch performance report showing the number of books issued, returned, and the total rental revenue for each branch.
* **Task 16**: Use CTAS to create a table of active\_members who have issued at least one book in the last two months.
* **Task 17**: Find the top 3 employees who have processed the most book issues and display their details.
* **Task 18**: Identify members who have issued books with a "damaged" status more than twice.
* **Task 19**: Create a stored procedure to manage the issuance of a book, checking its availability and updating its status accordingly.
* **Task 20**: Use CTAS to create a table that calculates the total fines for members with overdue books, based on a fine of $0.50 per day.

**Conclusion**

This project demonstrates the application of SQL skills in creating and managing a library management system. It includes database setup, data manipulation, and advanced querying, providing a solid foundation for data management and analysis.

**How to Use**

1. **Clone the Repository**: Clone this repository to your local machine.
2. **Set Up the Database**: Execute the SQL scripts in the database\_setup.sql file to create and populate the database.
3. **Run the Queries**: Use the SQL queries in the analysis\_queries.sql file to perform the analysis.
4. **Explore and Modify**: Customize the queries as needed to explore different aspects of the data or answer additional questions.

**Acknowledgments**

The dataset used for this analysis was provided by the YouTube channel [**Zero Analyst**](https://www.youtube.com/@zero_analyst) from one of their project videos. While the dataset originates from this source, all SQL queries for data cleaning, exploration, and analysis were written independently for this project.

**Author**

* **Parveen Jalwal**
* **LinkedIn**: <https://www.linkedin.com/in/parveen-jalwal-201a2a302>