PROJECT REPORT

ON

E-LIBRARY MANAGEMENT SYSTEM

SUBMITTED BY

VENKATA SIVA SAI MANOJ YATRASI (02080998)

HARSHA VARDHAN REDDY SAPPIDI (02081369)

KRISHNA KISHORE KUNAM (02083290)

submitted on December 15, 2023

University of Massachusetts Dartmouth

**Problem Statement :**

The current system in libraries often faces challenges in efficiently managing book records, user information, and loan transactions. Manual processes lead to errors, difficulty in tracking books, and delays in managing user requests. Retrieving specific book information or user details can be time-consuming, impacting the overall library experience.

This inefficiency hampers the smooth functioning of libraries, affecting both librarians and users. Implementing a digital Library Management System (LMS) utilizing databases will streamline operations.

It will enable librarians to easily manage author details, diverse book departments, individual book information, user records, and book loans. With this system, librarians can swiftly track book availability, manage user borrowing histories, and organize books by department for easier access.

Users can conveniently find and borrow books, reducing wait times and enhancing their overall library experience. This database-driven solution promises accuracy in record-keeping and timely retrieval of information, ensuring a seamless library experience for both librarians and patrons.

**Database Design:**

1. **author\_details Table:** Stores information about authors such as their name, nationality, birth year, relevant links, and contact details. Each author is assigned a unique ID for identification.

2. **departments Table:** Contains records of different book departments and their sub-departments. Each department has its unique ID for easy referencing.

3. **book\_details Table:** Holds data about books including the book title, publication year, the author's ID (linked to the author\_details table), and the department ID (linked to the department's table). Each book has a unique ID.

4. **library\_books Table:** Records each physical copy of a book in the library. It includes a unique ID for each book copy and the date it was added to the library. It references the book\_details table to link specific book copies.

5. **library\_users Table:** Stores user information like their names, email, phone numbers, and registration date. Each user has a unique ID for identification.

6. **book\_loans Table:** Tracks the borrowing history of books by users. It includes details like the user ID (linked to library\_users), book ID (linked to library\_books), loan date, return date, and any penalties incurred.

Relationships between tables are established using Foreign Keys, ensuring data integrity by linking information across tables accurately. For instance:

* The book details table references author details and departments tables via the author\_id and department\_id fields, respectively.
* The library\_books table references the book\_details table through the book\_id field.
* The library\_users and book\_loans tables reference each other through user\_id and book\_id, respectively.

This schema maintains data consistency and relationships between different aspects of the library, facilitating effective management of books, authors, users, and their interactions.

A screenshot of a computer

Description automatically generated

**Data Collection and SQL Query Development :**

For our library management system project, we embarked on a diligent data collection journey. We scoured the internet to gather pertinent information about books and authors, manually compiling a dataset in an Excel file. This dataset encompassed crucial details such as book titles, publication years, author names, nationalities, birth years, and relevant links. However, encountering challenges in finding a pre-existing dataset, we meticulously curated and structured the data to fit our database schema. This involved rigorous data cleaning, ensuring accuracy, and organizing the information to adhere to the defined structure.

* The subsequent step was the development of SQL queries and statements to seamlessly integrate our compiled data into the database.
* These queries were designed not just to insert the data but also to link authors to their respective books, categorize books into specific departments, and record the library's book inventory.
* Each table was carefully populated, considering the unique identifiers assigned to authors, departments, books, library books, and users.
* The SQL statements were crafted with precision to facilitate future data extraction, manipulation, and analysis, thereby addressing the core objectives of our project.

Our SQL queries encompassed a spectrum of operations, enabling us to extract comprehensive insights. These operations included retrieving information about authors, listing book titles and publication years within specific departments, tracking book borrowing history by users, and calculating penalties incurred for late returns. Through this meticulous data collection and SQL query development process, we ensured a robust foundation for our library management system, facilitating efficient data handling and analysis for enhanced library operations.

**Performance Tuning :**

In enhancing database performance, our team dedicated effort to optimize the system for swift data access. Focused on efficiency, we implemented indexing strategically across primary and foreign keys within the tables .By intelligently indexing fields like

* author ID
* department ID
* book ID
* user ID

These crucial columns used in joins and filtering conditions, we significantly expedited data retrieval processes. Additionally, query optimization became a cornerstone in refining our SQL commands. We carefully crafted and fine-tuned queries to minimize execution time and resource usage. Leveraging JOINs efficiently, we ensured seamless connections between tables, streamlining the retrieval of relevant information while reducing unnecessary data processing.

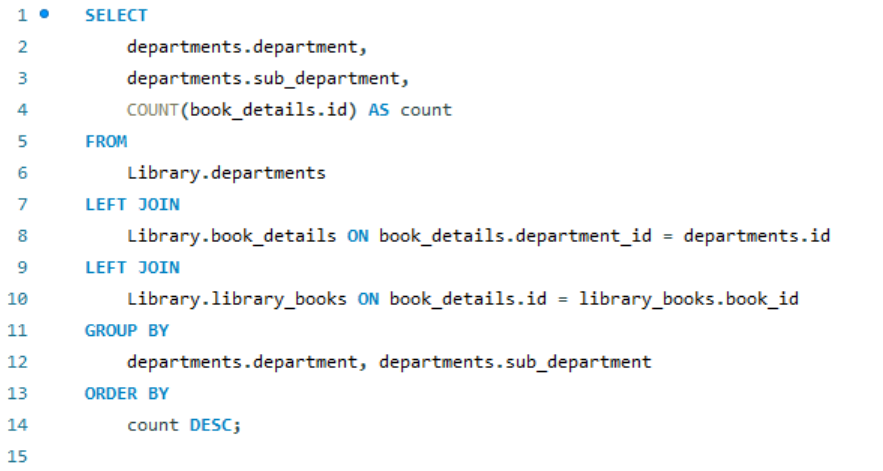
Furthermore, the optimization strategies extended to specific queries targeting essential functionalities.For instance, a query was designed to count books department-wise. By optimizing this query, we efficiently aggregated book details and department information, generating accurate counts of books associated with each department and sub-department swiftly.

Moreover, another tailored query was developed to compute the total fines accrued by users due to late returns. This optimization involved joining loan records with user details and summing penalties, delivering precise results promptly. In parallel, we optimized a specific query to fetch users with pending loans. This refined query efficiently identified users with outstanding book loans by joining loan records with user and book details, filtering specifically for non-returned books.

This optimization ensured the quick identification of users with pending obligations. Through meticulous indexing, query optimization, and tailored query enhancements, our team significantly improved the database's efficiency and performance. These optimizations empower the system to handle operations swiftly, providing users with faster access to critical information while ensuring smoother functionality across various database interactions.

**Questions and Solutions :**

**Question 1:** *How many books are there in each department and sub-department?*

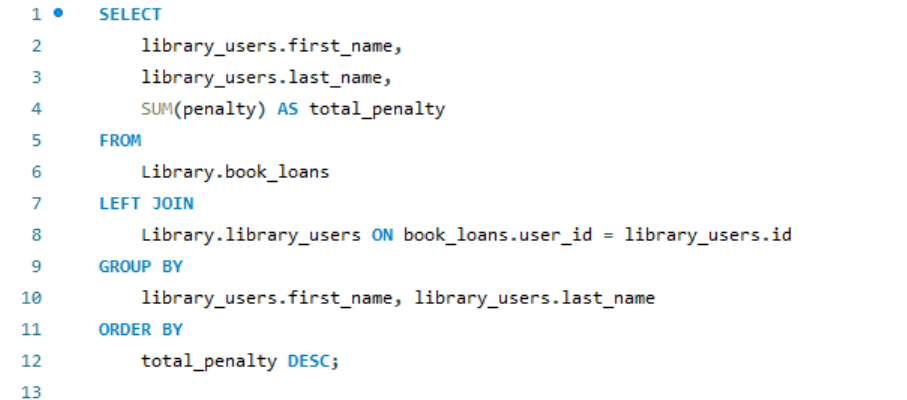


**Query Execution Plan**:

* Join departments with book details and library books.
* Group results by department and sub-department.
* Perform count and sort by count in descending order.

**Question 2:** *What is the total fine accrued by each student?*

***SQL QUERY***

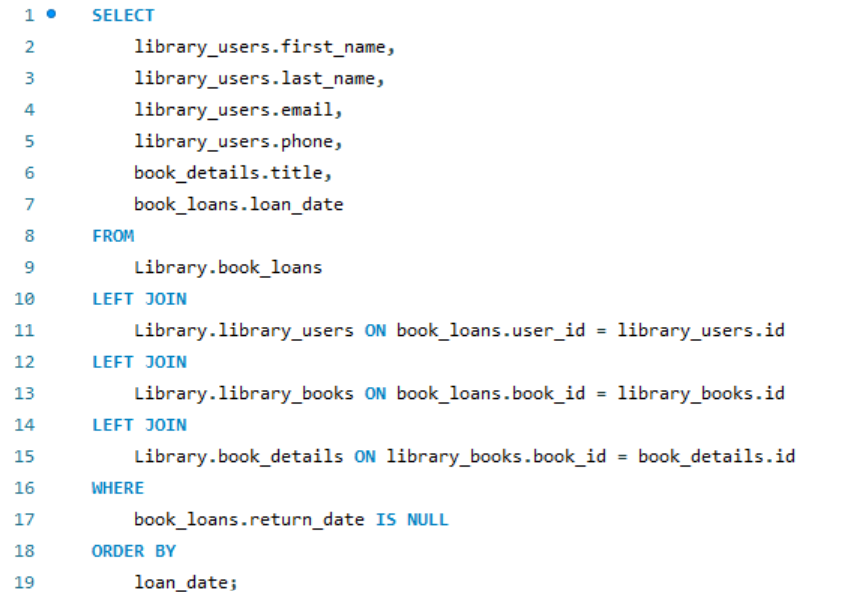
**

**Query Execution Plan**:

* Join book\_loans with library\_users.
* Group results by user names.
* Sum penalties and sort by total penalties in descending order.

**Question 3:**  *Who are the users with pending book loans along with their contact details and the titles of the books they've borrowed?*

***SQL QUERY***

**

**Query Execution Plan**:

* Join book loans with library users, library books, and book details.
* Filter records where the return date is NULL.
* Order results by loan dates.

**SQL QUERIES:**

Creating tables for

* author details
* departments,books
* details,library
* books,library
* users,book loans

**code:**

**A screenshot of a computer

Description automatically generated­­­­­**

**Adding user :**

**A screenshot of a computer

Description automatically generated**

**Adding data to sql :**

* Name
* Nationality
* birth year
* wikipedia link
* contact email

**SQL QUERY**

**A screen shot of a computer

Description automatically generated**

**Adding data related to departments ID , department subject,department**

**SQL QUERY**

**A computer screen shot of a computer screen

Description automatically generated**

**Adding author details in database sql query**

**SQL CODE:**

**A computer screen shot of a computer screen

Description automatically generated**

**Adding book id and book entry data**

**A screenshot of a computer

Description automatically generated**

**Adding id , user id, book id,loan date,return date, penalty**

**A screenshot of a computer

Description automatically generated**

**Indexing and switches library schema:**

**A screenshot of a computer

Description automatically generated**

**Deployment :**

I used MYSQL WorkBench for this project :

Deploying a Library Management System (LMS) using MySQL Workbench involves several steps, including database creation, schema design, and data population. Here's a general outline of the deployment process using MySQL Workbench.

* Open MySQL Workbench and connect to your MySQL Server.
* Design the database schema for your LMS. Identify tables for books, users, book loans etc.
* In MySQL Workbench, create tables based on your schema design
* You can manually insert sample data using SQL INSERT statements
* Set up your LMS application to establish a connection with the MySQL database. Furnish the essential connection information, encompassing the hostname, port, database name, username, and password.
* Execute sample queries.

**Limitations :**

In our library database project, certain limitations and constraints need consideration.

**Data Limitations:**

Firstly, data limitations could affect the completeness of our records. Incomplete or inaccurate author or book information may exist due to external sources or human error during data entry.

Integrity of data:

Validation and limitations: The data should be adhere to the rules i.e. in general we take multiple factors to represent a book such as key words constraints etc. these should have to maintain a data integrity to validate otherwise these may cause data loss.

**Time Limitations:**

Additionally, time constraints played a role; the project timeline might have affected the depth of data verification and entry. Moreover, the database's effectiveness relies on the regular update and maintenance of records, which might pose a challenge in a dynamic library environment.

Lastly, the scope of the project might not encompass all potential future needs of the library, potentially requiring adjustments or expansions in the database structure. These limitations could impact the accuracy and comprehensiveness of the database, influencing the outcomes and decisions based on its data.