Library management system

Anirudh Sowmyanarayan (50415686)
Shri Vignesh Senthil Kumar (50414786)

ABSTRACT

The project focuses on designing a library management system which helps in maintaining a database by the librarian, that is useful to enter new books and record books borrowed by the members with respective return dates.

I. INTRODUCTION

A library management system allows the librarian to efficiently maintain library resources in a more operative manner that will help save time. Processes such as maintaining the information about the books present in the library, their authors, the members of library to whom books are issued are made much more convenient for the librarian. Maintenance of all this information manually is a very complex task. Owing to the advancement of technology, organization of library management system becomes much simpler. This has been designed to computerize and automate the operations performed over the information about the members, book issues and returns and all other operations. It reduces the workload of management as most of the manual work done is reduced.

II. PROBLEM STATEMENT

Library is has many books categorized by genre, authors and editions which can become difficult track in a lending library scenario. Members of the library can be a high and tracking the number of books lent, edition and details can be tedious in Excel or other manual methods.
When members look for a book which is often lent to other members, library assistants and technicians will have to look for the book in the shelves or other records to identify the book.
In some cases the book might have been moved to collectibles which is not lent anymore due to the condition of the book.

This database provides an efficient solution to this issue by helping multiple users in parallel. Apart from the assistants, the books table can be made available to the members where they can check the stock availability of a book before walking in to borrow the same.
This database serves as a reserve for any information required by the library employees as well as the members.

III. TARGET USER

This dataset contains the book information about all book, authors, publishers and the stock of the book in hand which is used by Technicians. They can categorize the books by genre, authors or publishers and identify the most preferred books among the members.

When limited data from this database is made accessible to the members, they can pre-check the availability of the books before walking into the library for borrowing the same.

This database is useful for Library Assistants who are with the job of capturing newly borrowed books to easily book up the members and books by their IDs and record the same.

IV. DATASET DESCRIPTION

It contains all the necessary details regarding the books, authors, publishers, members, and the borrowing records at the library. This is in the form of an excel sheet which is very difficult to interpret for common people, therefore it's made into a database so that people can use it.

V. LIST OF RELATION AND ATTRIBUTES

- Books (book_id: varchar, book_name : varchar,genre : varchar,author_id : varchar, book_issue_datev : timestamp,publisher_id : varchar, book_total_count : integer)

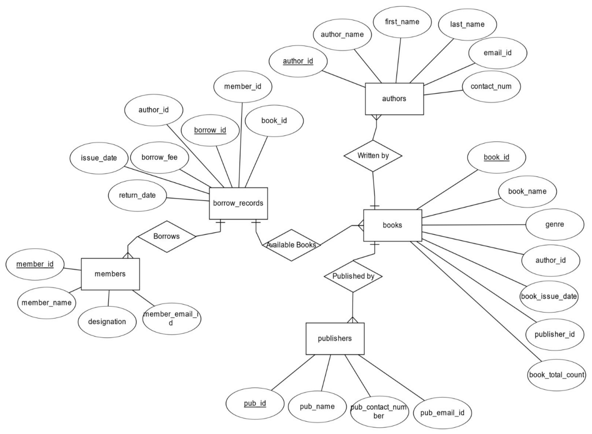
- Members (member_id: varchar,member_name: varchar,member_email_id: varchar,designation:varchar)

- Publishers(pub_id: varchar, pub_name: varchar, pub_email_id: varchar, pub_contact_number:varchar)

- Authors (author_id: varchar first_name: varchar, last_name:varchar, email_id: varchar, contact_number : varchar)

- Borrow_records(borrow_id:varchar, book_id : varchar, member_id : varchar, author_id : varchar,issue_date : timestamp, return_date : timestamp, borrow_fee : float

VI. ER DIAGRAM



VII. CREATING TABLES

1) Authors

```
2)drop table if exists authors;
3)create table authors    (author_id
  varchar(20) NOT NULL primary key,
4)                         author_name
  varchar(200) NOT NULL,
5)                         first_name
  varchar(200) NOT NULL,
6)                         last_name
  varchar(200) NOT NULL,
7)                         email_id
  varchar(200)  NOT NULL,
8)                         contact_num
  varchar(10)    NOT NULL);
```

2) Publishers

```
drop table if exists publishers;
create table publishers (pub_id varchar(20)
NOT NULL primary key,
                         pub_name
varchar(200) NOT NULL,
                         pub_email_id
varchar(200) NOT NULL,
                         pub_contact_number
varchar(10) NOT NULL);
```

3) Members

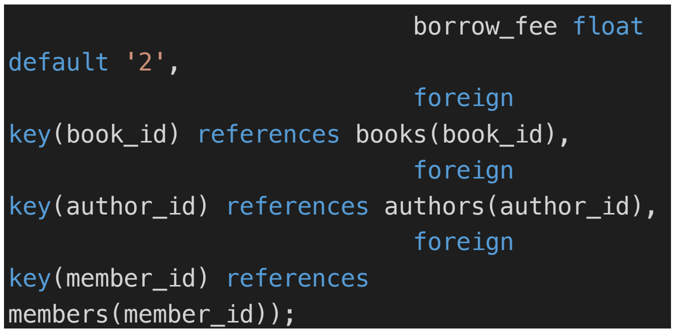
```
drop table if exists members;
create table members    (member_id varchar(20)
NOT NULL primary key,
                         member_name
varchar(200) NOT NULL,
                         member_email_id
varchar(30)    NOT NULL,
                         designation
varchar(20) NOT NULL);
```

4) Books

```
drop table if exists books;
create table books(book_id varchar(30)NOT NULL
primary key,
                    book_name varchar(200) NOT
NULL,
                    genre varchar(200) NOT
NULL,
                    author_id varchar(20) NOT
NULL,
                    book_issue_date timestamp
NOT NULL,
                    publisher_id varchar(20)
NOT NULL,
                    book_total_count int,
                    foreign key(publisher_id)
references publishers(pub_id),
                    foreign key(author_id)
references authors(author_id));
```

5) Borrow_records

```
drop table if exists borrow_records;
create table borrow_records(borrow_id
varchar(20) NOT NULL primary key,
                            book_id
varchar(30)NOT NULL,
                            member_id
varchar(20) NOT NULL,
                            author_id
varchar(20) NOT NULL,
                            issue_date
timestamp NOT NULL check
(issue_date<=return_date),
                            return_date
timestamp,
```



```
                            borrow_fee float
default '2',
                            foreign
key(book_id) references books(book_id),
                            foreign
key(author_id) references authors(author_id),
                            foreign
key(member_id) references
members(member_id));
```

VIII. BOYCE-CODD NORMAL FORM

Boyce–Codd Normal Form (BCNF) is based on functional dependencies that take into account all candidate keys in a relation; however, BCNF also has additional constraints compared with the general definition of 3NF.

List of Functional Dependencies **:**
Books
Book_id_id --> {Book_name}

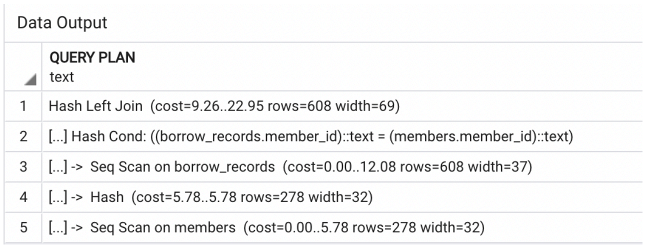
Borrow_records
Borrow_id --> {record_id, cost}

Authors:
author_id -->
{author_name,first_name,last_name}

members:
member_id--> {member_name,member_email_id
}

IX. QUERY ANALYSIS EXECUTION

EXPLAIN select * from members right outer join borrow_records on members.member_id = borrow_records.member_id


Data Output

```
QUERY PLAN
text
1  Hash Left Join  (cost=9.26..22.95 rows=608 width=69)
2  [...] Hash Cond: ((borrow_records.member_id)::text = (members.member_id)::text)
3  [...] ->  Seq Scan on borrow_records  (cost=0.00..12.08 rows=608 width=37)
4  [...] ->  Hash  (cost=5.78..5.78 rows=278 width=32)
5  [...] ->  Seq Scan on members  (cost=0.00..5.78 rows=278 width=32)
```

X. CONTRIBUTION OF TEAM MEMBERS

Anirudh Sowmyanarayanan
- Creating tables for the dataset
- Inserting data in tables created
- Putting into BCNF form.
- Report writing
- Testing and debugging queries

Shri Vignesh Senthil Kumar
- Writing queries
- Report writing
- Query execution analysis