Ahsanullah University of Science and Technology (AUST)
Department of Computer Science and Engineering

Project Proposal: [Library Management System]

Course No.: CSE4126

Course Title: Distributed Database Systems Lab

Semester: Fall 2022

Date of Submission - [22.08.2023]

Submitted To-

[Sanzana Karim Lora & Zarin Tasnim Shejuti]

Submitted By-

[190204076] : [Sanjida Akter]

Lab Group: [B1]
Year: 4th
Semester: 1st
Department: CSE

Library Management System

Description:

It gives us great pleasure to offer our proposal for developing a comprehensive library management system. This system's objective is to give librarians and other library staff a productive a user-friendly platform by simplifying and automating the various administrative tasks required in running a library. A number of features will be part of our solution, all of which are intended to enhance the overall library experience for both staff and users. Librarians will find it easy to manage the library's resource collection, which includes books, magazines, journals, multimedia, and other items, thanks to our system. We'll create a catalogue so people can look for resources, see details, and make reservations. With the ability to remotely view the library's collection, this function will improve users' accessibility and convenience.

Database Schema:

Global Schema:

Books(book_id, title, author_name, publication_date, available_book, genre, age_group, total_book)

Members(member_id, name, semester, year, contact_details, membership_type, no_of_books_taken, total_fines, paid_fees)

Borrowings(borrowing_id, book_id(fk), member_id(fk), fee, borrowing_date, fine, return_date, actual_return_date, staff_id(fk))

Staff(staff_id, name, position, contact_details)

Fragmentation Schema:

Books_fic = SL genre = 'Fiction' Books

Books_nov = SL genre ='Novel' Books

Books_mys= SL genre = 'Mystry' Books)

Books_en= SL genre ='Engineering' Books

Members_stan = SL membership_type = 'Standard' Members

Members_prem = SL membership_type = 'Premium' Members

Members_active = PJ $_{member\_id, membership\_type, no\_of\_books\_taken, name, paid\_fees}$(SL $_{no\_of\_books\_taken>0}$ Members)

Members_inactive = PJ $_{member\_id, membership\_type, no\_of\_books\_taken, name, paid\_fees}$ (SL $_{membership\_status = 'Inactive'}$ Members)

Borrowings_returned= SL $_{return\_date\ IS\ NULL}$ Borrowings

Borrowings_NotReturned= SL $_{return\_date\ IS\ NULL}$ Borrowings

Borrowings_fee = PJ $_{member\_id,fee,borrowing\_date}$ (SL $_{fee<60}$Members)

Borrowings_fee_more = PJ $_{member\_id,fee,borrowing\_date}$ (SL $_{fee>60}$Members)

Borrowings_fine = PJ$_{member_id,fine,borrowing_date,return_date,actual_return_date}$ (SL $_{fine<=50}$Members)

Borrowings_fine_more = PJ $_{member\_id,fine,borrowing\_date,return\_date,actual\_return\_date}$ (SL $_{fine>50}$Members)

$Staff_1$ = PJ $_{staff\_id,staff\_name}$ (SL$_{is_Admin=”admin”}$Staff)
$Staff_2$ = PJ $_{staff\_id,staff\_name}$ (SL $_{is\_Admin=”not\ admin”}$ Staff)

Allocation Schema:

$Books^1_{books\_fic,books\_nov}$,$Book^2_{books\_mys,books\_en}$,
$Members^1_{members\_stand,members\_active}$,$Members^2_{members\_prem,members\_inactive}$,
$Borrowings^1_{borrowings\_fee,borrowings\_fine,borrowings\_not\_returned}$,
$Borrowings^2_{borrowings\_fee,borrowings\_fine,borrowings\_returned}$,
$Staff^1_{staff\_clerk}$,$Staff^1_{staff\_admin}$.

Project Functionalities:

Borrow a Book:

- If a student want to borrow a book, then we have to enter book_id, member_id and staff_id. Then It will check if the book is available or not as well as if the member is registered or not. If any of the information is missing, then we won't issue the book.

```
Package body created.

Enter value for book_id: 5
Enter value for member_id: 2
Enter value for staff_id: 1
Not ISSUING BOOK TO MEMBER 2.
Book not found.

PL/SQL procedure successfully completed.


Trigger created.
```

Here we can see the book is not available. That's why the book is not issued.

```
Package created.


Package body created.

Enter value for book_id: 2
Enter value for member_id: 9
Enter value for staff_id: 2
Not ISSUING BOOK
Member not found.
```

Here we can see the member is not registered. That's why the book is not issued.

- If the member and book is available the we can issue book. The member has to return the book after 14days. Otherwise he will be fined. If the member borrows fiction type of book, he will have to pay 50 tk. Otherwise he will have to pay 100 tk to borrow a book.

```
Package created.


Package body created.

Enter value for book_id: 1
Enter value for member_id: 2
Enter value for staff_id: 2
ISSUING BOOK TO MEMBER 2.
Book borrowed successfully.
Borrowing ID: 9
Return Date: 05-SEP-23
Fee: 50
Fine: 0
```

```
Package created.


Package body created.

Enter value for book_id: 4
Enter value for member_id: 2
Enter value for staff_id: 1
ISSUING BOOK TO MEMBER 2.
Book borrowed successfully.
Borrowing ID: 10
Return Date: 05-SEP-23
Fee: 100
Fine: 0
```

- A member can not borrow more than 5 books at a time. If he wants to borrow more, he will have to return the book.

```
Package created.


Package body created.

Enter value for book_id: 4
Enter value for member_id: 1
Enter value for staff_id: 1
Member has already borrowed the maximum allowed number of books.
```

- If the member borrows more than 3 book, then he will become a premium member. After that he will get 10% discount on his next purchase.

```
Package body created.

Enter value for book_id: 3
Enter value for member_id: 2
Enter value for staff_id: 1
ISSUING BOOK TO MEMBER 2.
Book borrowed successfully.
Congratulations!!You have become a premium member
Borrowing ID: 17
Return Date: 05-SEP-23
Fee: 90
Fine: 0
```

- The premium members will get additional discounts on special occasions like 15% discount on 15August, 16% discount on 16December, 21% discount on 21February, 26% discount on 26March. To show the example, we have modified the function like we have a discount of 22% on 22August. Then the user will get 22% discount.

```
Package created.


Package body created.

Enter value for book_id: 3
Enter value for member_id: 2
Enter value for staff_id: 2
ISSUING BOOK TO MEMBER 2.
Book borrowed successfully.
Borrowing ID: 18
Return Date: 05-SEP-23
Fee: 70.2
Fine: 0
```

Return A Book:

- To return the book we have to give the borrowing id and the book will be returned. But to return the book we will check if the book has already been returned or not.

```
Procedure created.

Enter value for borrowing_id: 16
Book returned successfully.
Borrowing ID: 16
Actual Return Date: 22-AUG-23
Fine: 0
Fee: 90
Staff ID: 1
The book is returned successfully
```

```
Enter value for borrowing_id: 16
Borrowing ID not found or the book has already been returned.
```

- We can get most active member by no of books taken in descending order.

```
The Active members are:
Member ID: 2, Name: abas, Books Taken: 5
The Active members are:
Member ID: 1, Name: 1, Books Taken: 5
```

- We have calculated the total revenue and displayed the amount that has been earned by lending books and fines.

```
-------------------------------------------------
Total Revenue: 1343
```

- We have calculated the staff performance based on accepting borrowing order and displayed the amount of orders the staff has received.

```
-------------------------------------------------
The Staff that has accepted the maximum number of orders:
Staff ID: 1, Name: hasib, Books Issued: 8
```

- We have retrieved the popular books that has been borrowed maximum times based on the book counts.

```
-------------------------------------------------
Most popular book is
Book ID: 1, Title: ajdf, Author: sdbhjds, Genre: Fiction, Borrowing Count: 3
-------------------------------------------------
Most popular book is
Book ID: 2, Title: acb, Author: asjhs, Genre: Novel, Borrowing Count: 3
-------------------------------------------------
Most popular book is
Book ID: 4, Title: dfssf, Author: dfssf, Genre: Mystry, Borrowing Count: 1
-------------------------------------------------
Most popular book is
Book ID: 3, Title: safd, Author: adsfd, Genre: Novel, Borrowing Count: 1
```

- We can renew the books if it is closed to deadline to avoid fines.

Here we can see ID 7's return date is 4th September .Now we will renew the date.

```
BORROWING_ID    BOOK_ID  MEMBER_ID BORROWING        FEE RETURN_DA ACTUAL_RE
------------ ---------- ---------- --------- ---------- --------- ---------
      FINE   STAFF_ID
---------- ----------
         7          3          1 21-AUG-23         90 04-SEP-23
         0          1
```

```
Enter value for choice: 6
Enter value for borrowing_id: 7
Select the function you want to run:
1. Most Active Members
2. Books Due Soon
3. Total Revenue
4. Staff Performance Report
5. Popular Books
6. Renew Books
Book with Borrowing ID 7 has been renewed.

PL/SQL procedure successfully completed.
```

```
BORROWING_ID    BOOK_ID  MEMBER_ID BORROWING        FEE RETURN_DA ACTUAL
------------ ---------- ---------- --------- ---------- --------- ------
      FINE   STAFF_ID
---------- ----------
         7          3          1 21-AUG-23         90 11-SEP-23
         0          1
```

Here the date has been renewed for 7 days more.

Invoice:

We have generated an invoice for the members. When the members pays for the book there will be a invoice to be generated.

```
Enter value for enter_user_name: abas
-------------INVOICE--------------
User Name: abas
Book Title: ajdf
Author: sdbhjds
Borrowing Date: 22-aug-2023
Return Date: 05-sep-2023
Staff Name: jasdh
============================
```

```
============================
-------------INVOICE--------------
User Name: 1
Book Title: ajdf
Author: sdbhjds
Borrowing Date: 21-aug-2023
Return Date: 04-sep-2023
Staff Name: hasib
============================
```