J –COMPONENT

REVIEW-2

DATABASE MANAGEMENT SYSTEMS

Online Library Management System

GROUP MEMBERS

Vinit Shahdeo

Shreya Deep Anand

Introduction

Manual process of keeping student records, book records, account details, managing employee is very difficult. There are various problems also faced by the student in library such as finding any particular book, information whether book is available or not, for what time this book will be available, searching of books using ISBN number etc. To eliminate this manual system, library management system has been developed. Library Management System will handle all the current issues faced by the students and by its admin personnel.

To store all the information in the database from where user will place their query and get the results on the basis of their query. Only valid users will be able to access this Library Management System. Through this Library Management System, it will be easy to manage accounts and various details of particular student and employees working under library along with the records of book.

The current Library Management System does not eliminate the process of searching books within the library campus. Students have to find books manually. They have to wait until they are not provided with their library card and token. For receiving book, they have to show their library card and wait in line for their turns. The admin personnel also have to look manually on which day which person will take the charge within library to manage the overall work.

Features

- Searching of books
- Issuing and returning books
- Paying fine(if any) online

- Librarian can read information about any member
- Librarian can track the books issued by a particular student

Functional Requirements

- The system must only allow user with valid id and password to enter the system.
- The user must be able to logout after they finished using system.
- System must be able to not allow two books having same book id.
- System must be able to search if book is available or not before issuing books.
- Admin can be able to see availability of the particular book, they can also be able to see the each user data (ie. Which book is issued to which user and the fine amount of the user)

Data requirements

- There will be the data of each books in the database.
- There will be the user name and password of each member and faculty in the database
- The record of the issued books will also be in the database.
- The record of all the members working in the library will be there.

DETAILED DATA REQUIREMENT:

Library will be managed by the admin. Each admin will have it's unique login id and password. The library will be managing the books, staffs and members(both students and faculties).

Each staff will be having it's unique id, a name, designation, salary, date_of_joining, address and dob.

Each of the books will have its unique book_code, a name, a subject_code, no_of_books, rack_no, cost, date_of_purchase and name of the author.

The members are the ones who would be accessing the library system for issuing and returning books and paying fine when not returned in time. The library consists of two members Faculties and Students.

Faculty (member) will be having it's unique faculty id, a name, an address, a contact number and a department. When the faculty issues books, it will contain attributes faculty id and book_id which in turn will contain issue and return date.

Student (member) will be having it's unique id, a name, an address, a contact number and a branch. When the student issues books, it will contain attributes id and book_id which in turn will contain issue and return date.

Software and Languages required

FRONT END:

1. HTML
2. CSS
3. JAVASCRIPT

BACK END:

4. PHP
5. APACHE(server)
6. MYSQL

Entity types

1. **BOOK**

Strong entity set- book_code is used to identify each entity uniquely

Book_code,book_name, Author, date_of_purchase, price, subject_code, rack_no, no_of_books

2. **STAFF**

Strong entity set- staff_id is used to identify each entity uniquely

Staff_id, staff_name, dob, address, designation, salary, date_of_joining,

3. **STUDENT**

Strong entity type- student_id is used to identify each entity uniquely

Student_id, name, branch, fine, address, phone_no, issue_date, expiry_date

4. **FACULTY**

Strong entity type- f_id is used to identify each entity uniquely

F_id, name, address, phone_no, department

5. **LIBRARIAN**

Strong entity type- admin_login is used to identify each entity uniquely

Admin_login,admin_password

6. **AUTHOR**

Strong entity type- author_id is used to identify each entity uniquely

Author_id,author_name, dob, address, experience

7. **PERIODICALS**

Strong entity type- pr_id is used to identify each entity uniquely

pr_id, pr_name, month_of_release, publisher_name

Relationships

1. **BOOK – STUDENT relationship**

It's a one to many relationship as one book can be issued by only one student.

This participation is partial from both the side because all the book cannot be issued and it is also not necessary that all the students of the library is issuing the book.

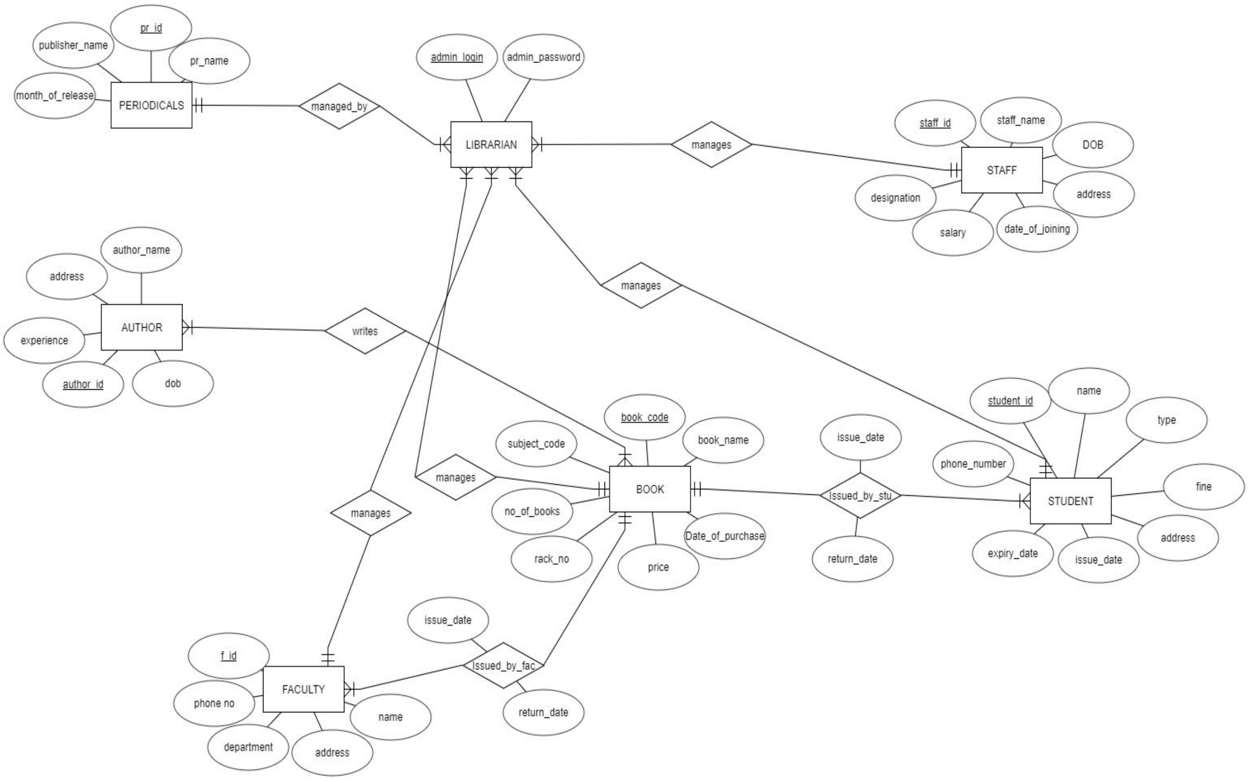
2. **BOOK – FACULTY relationship**

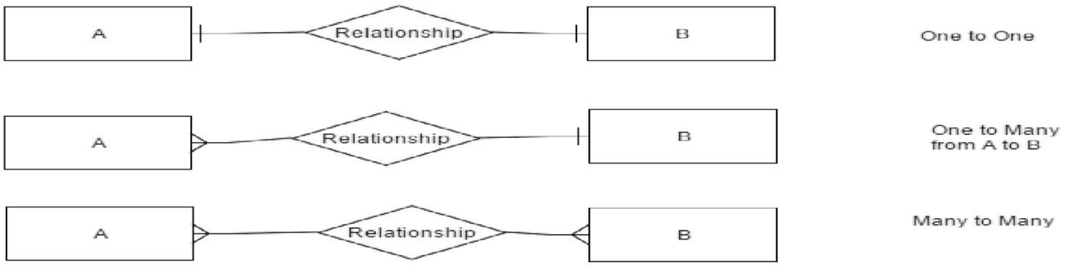
It's a one to many relationship as one book can be issued by only one faculty.

This participation is partial from both the side because all the book cannot be issued and it is also not necessary that all the faculty is issuing the book.

3. All other relationship is managed by the librarian ie. managing the other entity types of the library

Entity Relationship (ER) Diagram



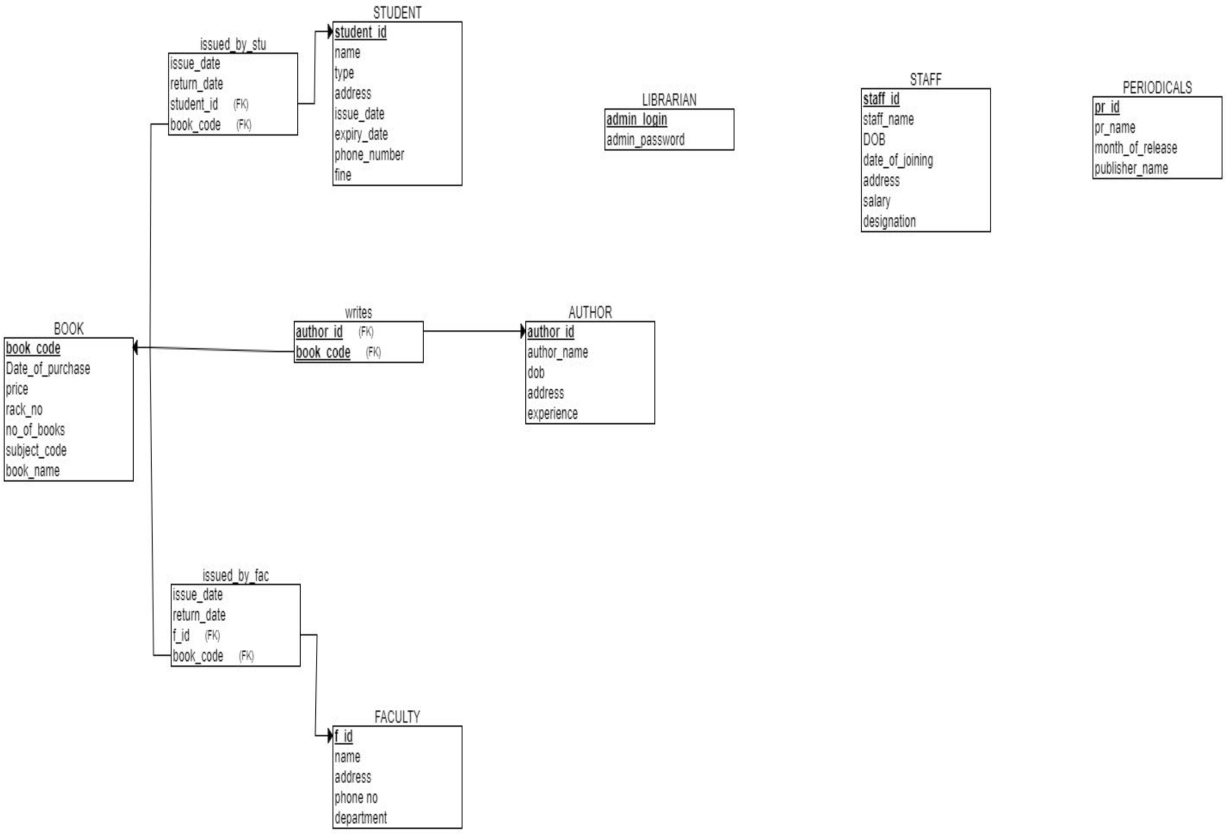


Schema

All the entity set used in this entity relationship diagram is strong so it can be directly reduced into relation schema as shown in the schema below.

For binary 1: N relationship, relation representing the participating entity type at N-side of relationship type is identified. In this relation we include as foreign key the primary key of relation that represents the other entity type.

Relation schema



Functional Dependencies: -

{book_code}→{date_of_purchase,book_name,price,rack_no,no_of_boo ks, subject_code}

{author_id}→{author_name, dob, address, experience}

{staff_id}→{staff_name, DOB, date_of_joining, address, salary, designation}

{Member_id}→{name, type, address, issue_date, expiry_date, phone_no, fine}

{faculty}→{name, address, phone_no, department}

{author_id}→{author_name, dob, address, experience}

{book_code,member_id}→{issue_date, return_date}

{book_code,f_id}→{issue_date, return_date}

Normalization: -

ent_id	e		ess		e_no	_date	y_date
	y		hi nagar, Chennai		325458	-2016	-2020
			adi		632588	-2016	-2016
	ash		Delhi	)	523456	-2017	-2021
	hant		alore		965876	3-2015	3-2019

We have reduced the student table from ER diagram, the given table is already in the First Normal Form since all the attributes is single valued. All the attributes of the student table are atomic.

Since the above table is already in the first normal form and there is no chance of partial dependency on the key attribute because it has only one key attribute so above table is in second normal form

Closure of the attribute set

F={student_id→ name, student_id→ type, student_id→ address, student_id→ fine, student_id→ phone_number, student_id→ issue_date, student_id→ expiry_date}

Student_id^{+} ={name, type, address, fine, phone_number, issue_date, expiry_date}

Transitive Dependency

A transitive dependency can occur only in a relation that has three
or more attributes. Let A, B, and C designate three distinct attributes
(or distinct collections of attributes) in the relation.

- No non-prime attribute is transitively dependent on prime key

attribute.

- For any non-trivial functional dependency, X → A, then either –
  - X is a superkey or,
  - A is prime attribute.

Since there is no transitive dependency, the table is already in Third
Normal Form

Boyde - Codd Normal Form

Since there is only one key attribute and key attribute is not dependent on the non key attribute thus we can say that the above table is in BCNF

2. Faculty (f_id,name, phone no, department, address)

The table is in BCNF. Due to the reason that f_id is the only candidate

key and all others are non-key attributes thus table is in 2NF. No non-key dependencies, hence in 3NF. One candidate key, so in BCNF

3. BOOK (book_code, book_name, subject_code, no_of_books, rack_no, price, Date_of_purchase)

The table is in BCNF. Due to the reason that book_code is the only candidate key and all others are non-key attributes thus table is in 2NF. No non-key dependencies, hence in 3NF. One candidate key, so in BCNF

4. STAFF (staff_id, staff_name, staff_name, address, date_of_joining, salary, designation)

The table is in BCNF. Due to the reason that staff_id is the only candidate key and all others are non-key attributes thus table is in 2NF. No non-key dependencies, hence in 3NF. One candidate key, so in BCNF

5. PERIODICALS (pr_id, pr_name, month_of_release, publisher_name)

The table is in BCNF. Due to the reason that pr_id is the only candidate key and all others are non-key attributes thus table is in 2NF. No non-key dependencies, hence in 3NF. One candidate key, so in BCNF

6. LIBRARIAN (admin_login, admin_password)

The table is in BCNF. Due to the reason that pr_id is the only candidate key and all others are non-key attributes thus table is in 2NF. No non-key dependencies, hence in 3NF. One candidate key, so in BCNF

7. AUTHOR (author_id, author_name, dob, address, experience)

The table is in BCNF. Due to the reason that author_id is the only candidate key and all others are non-key attributes thus table is in 2NF. No non-key dependencies, hence in 3NF. One candidate key, so in BCNF

8. issued_by_stu (student_id, book_code, issue_date, return_date)

The table is in BCNF. Due to the reason each attribute is atomic. Issue_date and return_date is dependent on the student_id and book_code and no partial dependency exists so the table is in second normal form. No transitive dependency exists in this table so it is in third normal form. And none of the non- key attribute is dependent on the key attribute so the above table is in BCNF.

9. issued_by_fac (f_id, book_code, issue_date, return_date)

The table is in BCNF. Due to the reason each attribute is atomic. Issue_date and return_date is dependent on the f_id and book_code and no partial dependency exists so the table is in second normal form. No transitive dependency exists in this table so it is in third normal form. And none of the non-key attribute is dependent on the key attribute so the above table is in BCNF.