Group 2 - Report Library management system (LMS)

1. Group members

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Email | Work distribution |
| 1 | Đỗ Khánh Nam (leader) | nam.dk225988@sis.hust.edu.vn | ERD, queries 21-30, indexes |
| 2 | Trần Anh Khôi | khoi.ta225981@sis.hust.edu.vn | Relational Schema, queries 1- 10 , indexes |
| 3 | Ngô Minh Đức | duc.nm226028@sis.hust.edu.vn | Report, queries 11 – 20, indexes |

1. Description
2. Objectives

* The goal is to build an application to manage information related to the operations of a library.

1. Model: It will work similarly to a real physical library (non-profit) with the help of an application.
2. Library - Staff side:

* Books: As books are the core element in the library system, our database needs to contain information about them, including bookid, title, author, language, year\_published,… Each book is assigned to a category, relating to their position in the library
* Our library may have multiple copies of the same book 🡪 each copy is uniquely defined by a book\_copyid. We also stored the status of a copy – available, unavailable, or removed
  + Each librarian has an account storing their id, full name, date of birth, gender, address, email, phone number, and role (manager, employee).
  + The librarian updates books:
    - Add new books.
    - Remove books.
  + The librarian updates book copies:
    - Add new book copies.
    - Remove book copies.
  + The library also wants to keep track of the borrowers’ preferences, since they want more and more readers. So, they make statistics of most loaned books, and most loaned book categories, so that they will buy more books of these types.

1. User side
   * If a person wants to borrow a book, he/she, first will register for a free reader card in the app. Users must sign up using their real information, including full name, date of birth, gender, address, phone numbers, email (1 phone number can only create 1 account). Then, they will have barcodes to scan whenever they go to the library.
   * Using the app, users can search for books. The search engine will have filters like titles, authors, categories, language to help users search for the books they want more easily. So, when they click on a book they search in the app, apart from the books’ information, they will also see the status of the book – available to borrow or not, and the number of these books left in the library. They can also see the book's position in the library and the library layout.
   * In case the status of the book they want to borrow is unavailable, users have an option to sign for notifications, which means that they are placed on a waiting list and when the book is available (the number of book copies > 0), they will get notified.
   * Users must borrow books in person by bringing the books they want to borrow to the front desk to process borrow procedure. The librarian will scan the user card's code and the book copies’ code to input loan information in the system. The date\_out is stored, and the return\_date is NULL.
   * The borrowed time of a book is 90 days. Borrowers must return books before that time. The app will send them notifications to remind them to return books.
   * To return borrowed books, borrowers need to bring the books to the front desk. The librarian will check the condition of the books. If the books are damaged, the borrowers need to pay compensation fee equal to the price of the books plus a processing fee for each book.
   * The app also sends notification to remind borrowers to return books.
2. Functionalities

* For user:

+ Create account and login.

+ Allow users to browse book inventory, search for book by category, title, author, publisher, year of publication, language. Find the position of a book in the library.

+ check if the wanted books are available for borrowing, if not, you can sign up for a waitlist. The system will notify you when your wanted books are available.

* For librarian:

+ Login.

+ Manage inventory: add or remove books titles, copies. Remove and restock damaged books.

+ Make statistics of most loaned book titles, waitlist record, borrow frequency of each book to understand our readers’ preferences, therefore we can adjust our inventory accordingly in term of genre, book titles, and number.

+ verify book return book.

+ summarize your working shift.

1. Requirements
2. Entities & Attributes:
   * Book
     + **BookID**
     + Title
     + Language
     + PublisherID
     + CategoryID
     + Number of copies (available)
     + Year published
     + Location
   * Author
     + **AuthorID**
     + Name
   * Publisher
     + **PublishID**
     + Name
   * Book copy
     + **Book\_copyID**
     + Price
     + Status (available/unavailable/removed)
   * Employee
     + **EmployeeID**
     + Name
     + DOB
     + Gender
     + Address
     + Phone
     + Email
     + Role
   * Borrower
     + **BorrowerID**
     + Name
     + DOB
     + Gender
     + Address
     + Phone
     + Email
     + Number of books allowed (<=5)
   * Loan
     + **LoanID**
     + Date\_out (the date the borrower borrows a bookcopy)
     + Return\_date (the date the borrower returns a bookcopy)
     + Damaged (= TRUE if the employee confirms that the bookcopy is damaged when the borrower returns, = FALSE otherwise)
   * Notification
     + **NotiID**
     + Message
     + Sent\_At
   * Visit
     + **VisitID**
     + Time
3. Relationship
   1. Book – book copy: 1 – m (A book can have numerous book copies, but a book copy only belongs to 1 book).
   2. Book – author: m – n (A book can be written by some authors, and an author can write many books).
   3. Category – book: 1 – m (A book can only belong to 1 category, but a category can contain many books).
   4. Publisher – book: 1 – m (A book can only be published by 1 publisher, but a publisher can publish many books).
   5. Employee – book: m –n (An employee can update many books, and a book can be updated by many employees).
   6. Employee – book copy: m – n (An employee can update many book copies, and a book copy can be updated by some employees.)
   7. Borrower – Loan: 1 – m (a borrower can borrow many times, but a loan can only have 1 borrower)
   8. Employee – Loan: 1 – m (an employee can verify many loans; a loan can only be verified by one employee).
   9. Book copy – Loan: 1 – m (a loan only has one book copy; a book copy can be loaned many times).
   10. Borrower – book copy: m – n (a copy can be damaged many times; a borrower can damage many copies).
   11. Borrower – book: m – n (a borrower can wait for many books; a book can be waited for by many borrowers).
   12. Borrower – notification: 1 – m (a borrower can have many notifications; a notification only has one borrower).
   13. Borrower – visit: 1 – m (a borrower can make many visits; a visit only has one borrower).
4. Queries

Searching books

* 1. Search by title.
  2. Search by language.
  3. Search by publisher.
  4. Search by author.
  5. Search by year published.
  6. Search by category.
  7. Search for location of a book.

Searching borrowers

* 1. Search by name.
  2. Search by year of birth.
  3. Search by gender.
  4. List borrowers who have visited in a period. (function)
  5. List borrowers that make the most visits.
  6. List how many books a borrower borrows in a year (2023).

Searching employees

* 1. Search by name.
  2. Search by role.

Managing books

* 1. List all the books in the library.
  2. Count the number of copies of a book. (function)
  3. Trigger to update number of copies of a book when add or remove copy.
  4. Count number of copies loaned in a period (function).

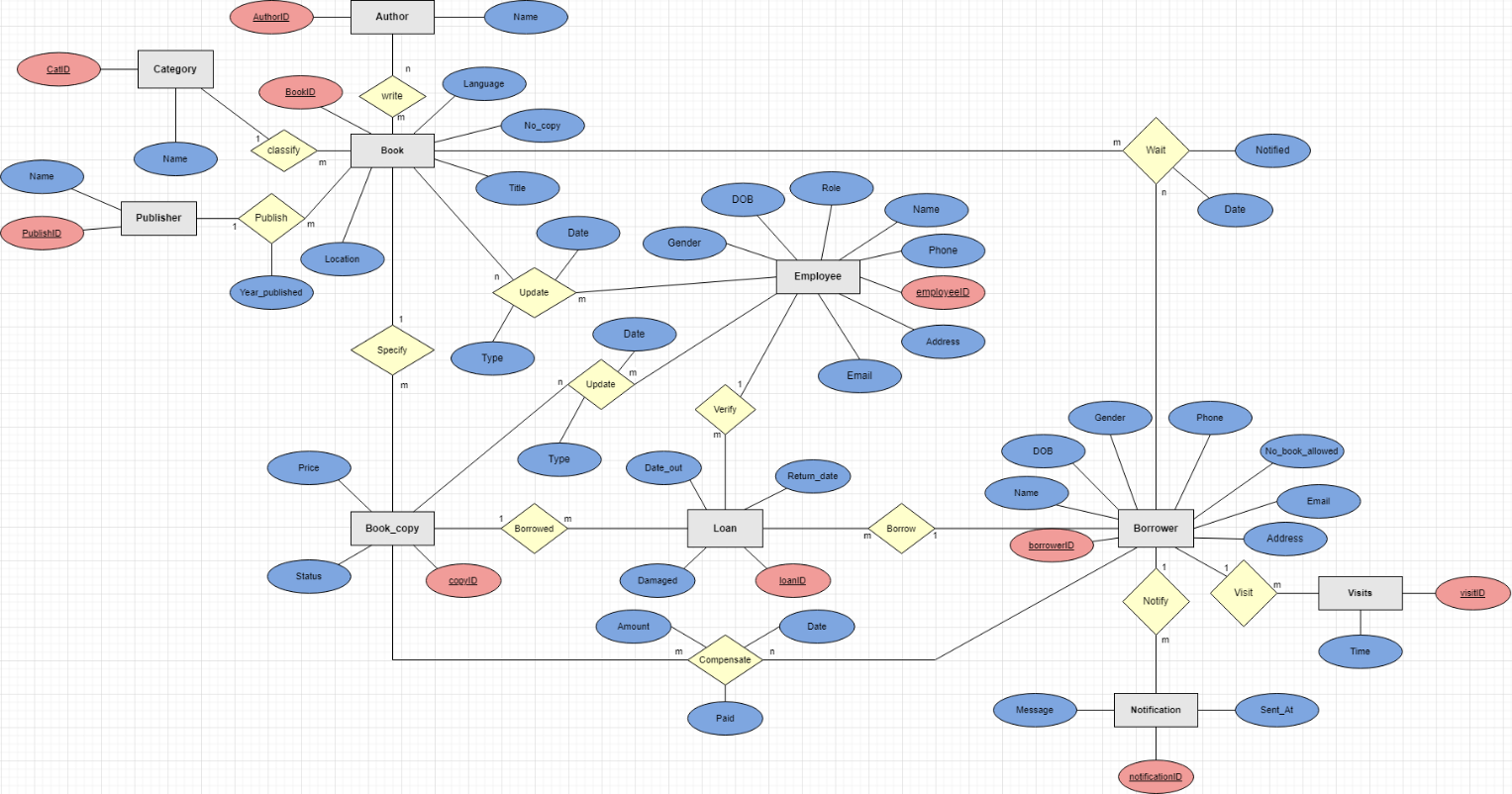
Statistics (to have a better understanding of readers)

* 1. List the most borrowed categories in 2023.
  2. List number of borrowers by gender.
  3. List number of people that have borrowed book by age group: < 18, 18 – 25, 25 – 40, 40 – 55,> 55.
  4. Return how many times a book is loaned in a given year.
  5. Count number of books returned damaged.
  6. Find the employee verify a given loan.
  7. List all overdue books.
  8. Trigger to update table book, book\_copy, borrower when new loan record is inserted in loan table.
  9. Trigger to update table book, book\_copy, borrower when a book is returned.
  10. Trigger to insert into table compensate when the returned book is damaged.
  11. Trigger to notify borrower when the books they have been waiting for are available.

1. Indexes: We create single-column indexes on
   1. In table “book”: columns: “title”, “language”, “location”, “catID”, “publisherID”, “year\_published”
   2. In table “borrower”: columns: “name”, “gender”, “dob”,
   3. In table “publisher”: columns: “name”,
   4. In table “author”: columns: “name”,
   5. In table “loan”: columns: “book\_copyID”, “borrowerID”,
   6. In table “compensate”: columns: “borrowerID”,
   7. In table “wait”: columns: “bookID”,

* because those tables have many records, those columns are frequently searched for and rarely changed.

1. ERD



1. Relational Schema

