**DB System**

**Assumptions:**

Primary Goals of the System:

1. Track and manage library resources, including physical books, eBooks, and electronic devices.
2. Facilitate borrowing and returning resources by students and staff, ensuring that borrowing limits and fines are enforced.
3. Manage reservations and loan offers for unavailable resources.
4. Record overdue loans and manage fine payments and member suspensions.
5. Keep track of popular resources based on borrowing history.

**Core Assumptions Derived from the Specification**:

1. The system tracks physical books, eBooks, and electronic devices such as laptops and tablets. Each resource has a unique ID and attributes relevant to its type (e.g., class number for physical and electronic resources, digital copy count for eBooks).
2. Each resources has its own borrowing rules, the usual loan period of a resource is 3 weeks, but some resources are available for short loan(3 days) only. Some resources may only be available for in-library use.
3. The library spans 3 floors, and physical items are stored on shelves identified by both floor number and shelf number. Items on the same topic share a class number.
4. Electronic devices are treated as physical resources for borrowing purposes but do not have a class number.
5. eBooks do not have a physical location but can have a limited number of digital copies available for borrowing, which restricts simultaneous access.
6. Library members include students and staff. Students can borrow up to 5 resources at a time, while staff can borrow up to 10.
7. Each library member holds a unique library card, which is used to track loans, fines, and borrowing history.
8. When a resource is unavailable, members can reserve it. Multiple reservations can exist for the same item, and when it becomes available, the earliest reservation is processed first.
9. A member must respond to a reservation offer within 3 days. If they fail to do so three times, the reservation is canceled.
10. Overdue items incur a fine of £1 per day. If a member’s total fines exceed £10, they are suspended from borrowing additional resources until the fines are paid, and all overdue resources are returned.
11. The system will track popular resources based on borrowing history, which includes details of past loans.
12. The system must be scalable to handle an indefinite number of resources, members, reservations, and loan records.

**Normalization:**

To enhance functionality and ensure efficient data handling, normalization is applied to the library database system. Below are the required attributes and entities in their normalized form, ensuring the database avoids redundancy and maintains integrity.

**Attributes for the Library Database System**

Based on the requirements, the system involves the following entities and their attributes:

* **Member**: memberId PK, Name, Email, DOB, memberType, totalFine, totalLoan, reservationFailed, resourceLimit
* **Resource**: resourceId PK, resourceType, locationId FK, borrowRule, digitalCopy, classNumber FK, resourceTitle, availability
* **Location**: locationId PK, floorNumber, shelfNumber, sectionName, classNumber FK
* **Loan**: loanId PK, memberId FK, resourceId FK, loanDate, dueDate, returnDate
* **Reservation**: reservationId PK, memberId FK, resourceId FK, reservationDate, expirationDate

**Functional Dependencies and Entity Groupings**

1. **memberId ⇒ Name, Email, DOB, memberType, totalFine, totalLoan, reservationFailed, resourceLimit**
2. **resourceId ⇒ resourceType, locationId, borrowRule, digitalCopy, classNumber, resourceTitle, availability**
3. **locationId ⇒ floorNumber, shelfNumber, sectionName, classNumber**
4. **loanId ⇒ memberId, resourceId, loanDate, dueDate, returnDate**
5. **reservationId ⇒ memberId, resourceId, reservationDate, expirationDate**

**Entities and Primary Keys**

1. **Member** (*memberId*, Name, Email, DOB, memberType, totalFine, totalLoan, reservationFailed, resourceLimit, resourceLimit)
   * Primary Key: memberId
2. **Resource** (*resourceId*, resourceType, locationId FK, borrowRule, digitalCopy, classNumber FK, resourceTitle, availability)
   * Primary Key: resourceId
   * Foreign Key: locationId (links to Location), classNumber(links to Location)
3. **Location** (*locationId*, floorNumber, shelfNumber, sectionName, classNumber)
   * Primary Key: locationId
   * Foreign Key: classNumber(links to Resource)
4. **Loan** (*loanId*, memberId FK, resourceId FK, loanDate, dueDate, returnDate)
   * Primary Key: loanId
   * Foreign Keys: memberId (links to Member), resourceId (links to Resource)
5. **Reservation** (*reservationId*, memberId FK, resourceId FK, reservationDate, expirationDate)
   * Primary Key: reservationId
   * Foreign Keys: memberId (links to Member), resourceId (links to Resource)

**Normalization Stages**

**First Normal Form (1NF)**

1NF requires each attribute to be atomic and contain no repeating groups.

* **Member**: All attributes are atomic, satisfying 1NF.
* **Resource**: All attributes are atomic, with each resource uniquely identified, satisfying 1NF.
* **Location**: Attributes are atomic, with each location uniquely identified, satisfying 1NF.
* **Loan**: Each loan transaction has a unique loanId, satisfying 1NF.
* **Reservation**: Each reservation has a unique reservationId, satisfying 1NF.

**Second Normal Form (2NF)**

2NF requires that each non-primary-key attribute is fully functionally dependent on the primary key.

* **Member**: All attributes (Name, Email, DOB, memberType, totalFine, totalLoan, reservationFailed) depend fully on memberId, satisfying 2NF.
* **Resource**: All attributes (resourceType, locationId, borrowRule, digitalCopy, classNumber, resourceTitle) depend fully on resourceId, satisfying 2NF.
* **Location**: All attributes (floorNumber, shelfNumber, sectionName, classNumber) depend fully on locationId, satisfying 2NF.
* **Loan**: Attributes (memberId, resourceId, loanDate, dueDate, returnDate) depend fully on loanId, satisfying 2NF.
* **Reservation**: Attributes (memberId, resourceId, reservationDate, expirationDate) depend fully on reservationId, satisfying 2NF.

**Third Normal Form (3NF)**

3NF requires that there are no transitive dependencies on the primary key.

1. **Member**: Noticed that resourceLimit is depend on memberType which is transitive, so we decide to separate it to another table called ResourceLimit.
2. **Resource**: Attributes depend directly on resourceId. Although locationId and classNumber are foreign keys, they do not create transitive dependencies within this entity, satisfying 3NF.
3. **Location**: Attributes are directly dependent on locationId, with no transitive dependencies, satisfying 3NF.
4. **Loan**: Attributes are non-transitively dependent on loanId, satisfying 3NF.
5. **Reservation**: Attributes are non-transitively dependent on reservationId, satisfying 3NF.

**Entities and Primary Keys with normalization:**

1. **Member** (*memberId*, Name, Email, DOB, memberType, totalFine, totalLoan, reservationFailed, resourceLimit)
   * Primary Key: memberId
   * Foreign Key: memberType
2. **ResourceLimit**(*memberType,* resourceLimit)
   * Primary Key: memberType
3. **Resource** (*resourceId*, resourceType, locationId FK, borrowRule, digitalCopy, classNumber FK, resourceTitle, availability)
   * Primary Key: resourceId
   * Foreign Key: locationId (links to Location), classNumber(links to Location)
4. **Location** (*locationId*, floorNumber, shelfNumber, sectionName, classNumber)
   * Primary Key: locationId
   * Foreign Key: classNumber(links to Resource)
5. **Loan** (*loanId*, memberId FK, resourceId FK, loanDate, dueDate, returnDate)
   * Primary Key: loanId
   * Foreign Keys: memberId (links to Member), resourceId (links to Resource)
6. **Reservation** (*reservationId*, memberId FK, resourceId FK, reservationDate, expirationDate)
   * Primary Key: reservationId
   * Foreign Keys: memberId (links to Member), resourceId (links to Resource)