

EER REPORT

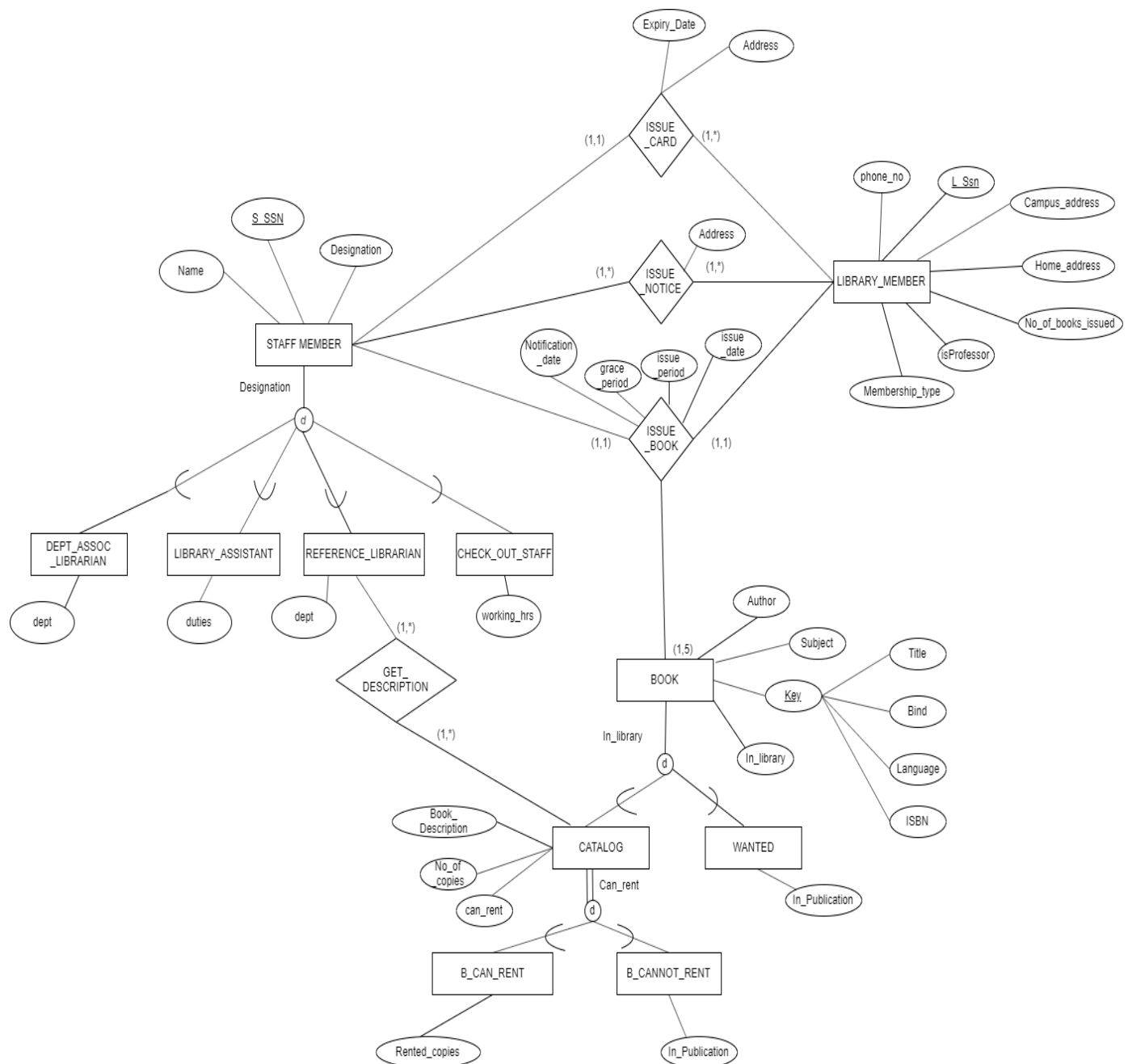
Prepared by:

Arpitha Patil – 1001777613

Megha Vijendra - 1001736938

The **enhanced entity–relationship (EER)** model (or extended entity–relationship model) in computer science is a high-level conceptual data model incorporating extensions to the original entity–relationship (ER) model, used in the design of databases. It was developed to reflect more precisely the properties and constraints that are found in more complex databases.

The EER diagram for the Library Database is as follows:



The EER diagram pertaining to the LIBRARY database has been made with the following assumptions:

1) Main entity types and their primary key:

Each entity has a unique attribute as shown below:

- LIBRARY_MEMBER: L_Ssn
- BOOK: Key (Title, Bind, Language, ISBN)

Here we consider the primary key called 'Key' which is a composite key containing Title, Bind, Language, ISBN because some books may have the same title and every book is identified by its International Standard Book Number (ISBN), a unique international code assigned to all books also two books with the same title can have different ISBNs if they are in different languages or have different bindings (hardcover or softcover). Also we get to know that editions of the same book have different ISBNs. Hence, we have a primary key as a composite key for this entity.

- STAFF_MEMBER: S_Ssn

Every entity and relationship types have names to identify them.

2) Specialization:

- Based on the attribute Designation, STAFF_MEMBER is specialized into dept_assoc_librarian, library_assistant, reference_librarian and check_out_staff. It is partial disjoint specialization.
- CATALOG is specialized into can_b_can_rent and b_cannot_rent based on the attribute can_rent. It is a total disjoint specialization. Meaning that the book in the library catalog can either be rented or can be just taken for reference.
- BOOK is specialized into catalog and wanted based on the attribute IN_LIBRARY. It is a partial disjoint specialization. If the book is in_library then we store the description of the book in the catalog. If not, we will check if the book is in publication then put it under wanted type.

3) Aggregation:

- If a relationship type has member attributes, then it has aggregates.
- All entity types are aggregation of inherent aggregation as seen from EER diagram as shown above.

4) Constraints:

- Books can be checked out after 21 days. Members are allowed to have only five books at a time.
- Professors are allowed to check out books for three-month intervals and have a two-week grace period
- Members have one week of grace before a notice is sent to them.
- A library card is valid for four years.
- The library does not lend some books, such as reference books, rare books, and maps.
- Fine is charged for lost cards and damaged books.

5) Relationships and Relationship types:

We can see many types of relationships in the schema and they are as follows:

- **ISSUE_CARD**: is a relationship of type 1: N where one staff member can issue card to a library member after getting all their details. Notice that the library member has an attribute called `isProfessor` which tells us that if that member is a professor then we need to fetch all his details from the employee records into staff members entity. If he is not a professor then all the details need to be collected before issuing the card.
- **ISSUE_NOTICE**: is a relationship of type M: N where staff members can issue notice regarding the renewal of their library cards one month before the library members card expires. It also has attribute called `address` so that we could send the notices via mail as well.
- **ISSUE_BOOK**: is a ternary relationship where in a staff member can issue one or max five books to the library members for rent at any given point of time. Here book entity type is used to check where the book requested by the library member is there in the library or not. And if it is there in the library to see if the book can be rented by the library member or not and also the number of copies of book is rented at the current time. Hence, we have ternary relationship in the schema.

Also, when the book is being given on rent, we would also store the issue date and period and also a grace period within which the books can be returned back to the library.

- **GET_DESCRIPTION**: is a relationship of type M: N where in the reference librarians should be able to access this description when members request information about a book from the catalog so that it would be helpful for the library member to know if the book could be rented or not.