# 4.3. Conceptual database design

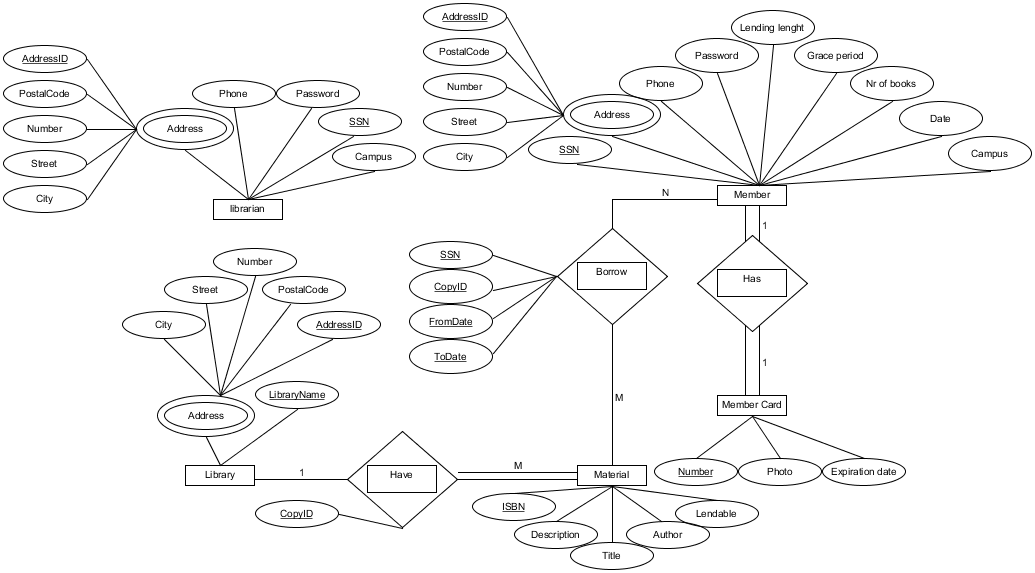
## 4.3.1. Entity Relationship (ER) Diagrams

### a. Overview:

An Entity Relationship diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

### b. Implementation:

The Entity Relationship Diagram was made by reviewing the Georgia Tech Libraries request for solution. Based on the information provided an early implementation of diagram (Figure 1) was created, this would be further improved in future iterations. To make our ER diagram we found the initial entity types (Member, member card, librarian, library, book) and their attributes, following that the relations between the entities were established and gradually perfected until the development group was satisfied with the results.



Figure

Focus when creating this diagram was set on borrowing books, as it is the main use case of a library. It was determined, that to identify a unique lending of material, SSN and BookID would not be enough, as book then could not be repeatedly borrowed by the same member. Furthermore, it was decided to have each book reference the library from which it has been borrowed, as GTL has agreements with other libraries allowing them to share books between them. To give more detailed view of what was required it was decided to further improve this diagram by using enhanced ER diagram.

## 4.3.2. Enhanced Entity Relationship (EER) Diagrams

### a. Overview:

Enhanced entity-relationship (EER) diagrams are basically an expanded upon version of ER diagrams. EER models are helpful tools for designing databases with high-level models. With their enhanced features, you can plan databases more thoroughly by delving into the properties and constraints with more precision.

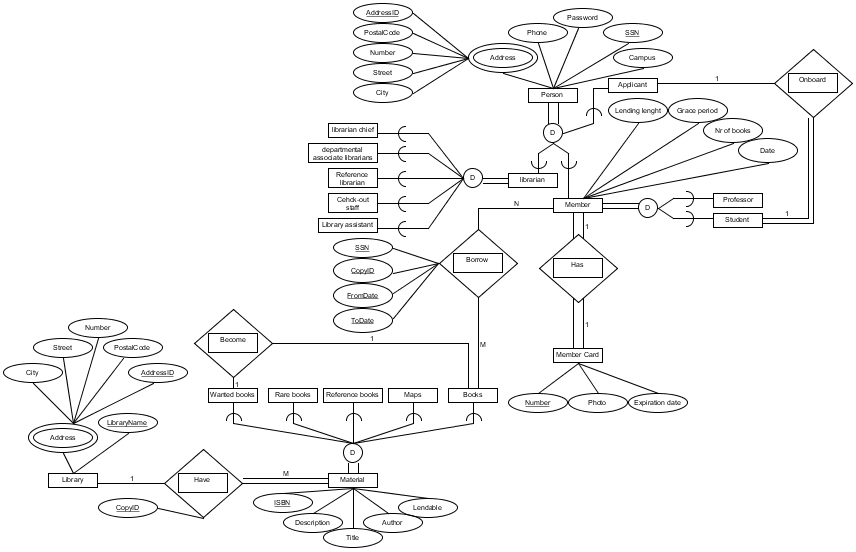
**An EER diagram provides you with all the elements of an ER diagram while adding:**

* Attribute or relationship inheritances
* Category or union types
* Specialization and generalization
* Subclasses and super classes

Both ER and EER diagrams provide the ability to design your database, while ER diagram gives you the visual outlook of your database (It details the relationships and attributes of its entities, paving the way for a smooth database development in the steps ahead), the EER diagrams, on the other hand, are perfect for taking a more detailed look at your information, so we opted to create one.

### b. Implementation:

Since we wanted our diagram to have higher amount of detail than provided by ER, we created Enhanced Entity Relationship Diagram, by using the previously made ER diagram. The results of this can be seen in Figure 2.



Figure

The connection between Figure 1 and 2 can be easily observed, with main difference here being the implementation of subclasses and super classes. The reason to add them is that each of the specializations have their own rules and when making the solution we did not want to forget about them, causing us to solve the issue later at a higher cost.

## 4.3.3. Database diagram

### a. Overview:

Database diagram shows how the information in database is organized and thus gives the reader a better understanding of how it works and what can be done with it. In a relational database, the schema can define information like the tables, fields, relationships, views, indexes, packages, procedures, functions, queues, triggers, types, sequences, materialized views, synonyms, database links, directories, XML schemas, and other elements.

### b. Mapping EER Model Constructs to Relations:

When we started work on creating the database schema it was decided to use the following steps, to be sure information saved is well representative of what was requested by the project, shown in EER diagram. (First 7 steps are shared with ER diagram but last two are unique only to enhanced version of ER)

1. Mapping of Regular Entity Types
2. Mapping of Weak Entity Types
3. Mapping of Binary 1:1 Relation Types
4. Mapping of Binary 1:N Relationship Types
5. Mapping of Binary M:N Relationship Types
6. Mapping of Multivalued attributes
7. Mapping of N-ary Relationship Types
8. Mapping Specialization or Generalization
   1. Multiple relations-Superclass and subclasses
   2. Multiple relations-Subclass relations only
   3. Single relation with one type attribute
   4. Single relation with multiple type attributes
9. Mapping of Union Types (Categories)

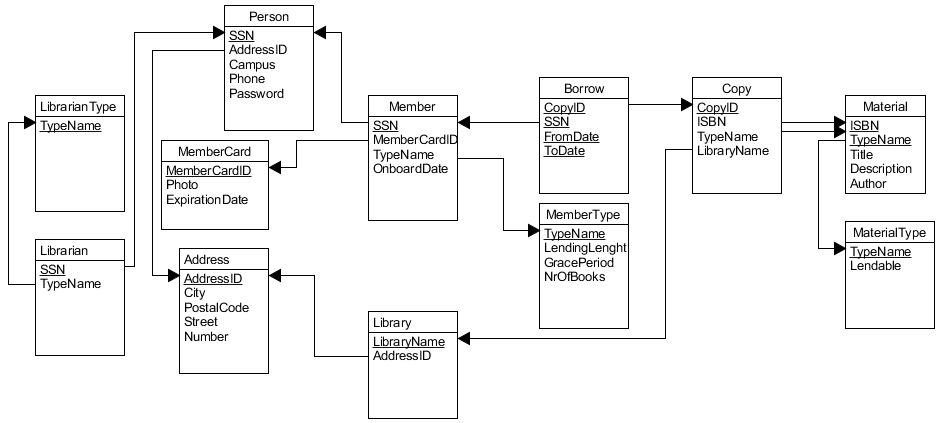
### c. Normalization:

Normalization is performed, to facilitate reduced data redundancy and improved integrity of the information. Normal forms are very important for relational database and if not followed could result in database being impossible to use or/and inefficient. This could affect the project in mayor way in the long term, possibly requiring complete rework of the system. There are many different normalization forms, with each being a step towards improved data management. The following is list of normal forms satisfied when creating the database:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. BCNF (Boyce-Codd Normal Form)

### d. Implementation:

Our database diagram, just like the EER early on changed quite a lot, but we managed to get the solution satisfying all the needs quite fast. As seen in Figure 3, we translated EER into database diagram which we later implemented and used throughout the project for data management. Mapping of specialization was done using 8a for super class: person and subclasses: member, librarian. This was done as we wanted all closely related information (Persons) to be saved in one easily accessible table. The rest of inherences (different types of: materials, librarians, members) were mapped using 8c, but none of them had any unique attributes to ones provided by superclass, only having different business logic associated.



Figure

Used:

<https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model>

<http://jcsites.juniata.edu/faculty/rhodes/dbms/eermodel.htm>

Ch8\_FundamentalsOfDatabaseSystems.pdf (I forgot where I got it XD)