COMP40725 Lab Book 3
Entity Relationship Modelling

- Create a folder called `Lab3` in your `COMP40725` folder. You may wish to have this inside a folder that is being synced by Google Drive
- Download the editable version of the lab document and save it as Lab3_studentNumber_firstName_LastName.doc

Note: To complete this lab book, multiple iterations of your ER diagram will need to be included at different stages of the design process. A single completed ERD is not sufficient.

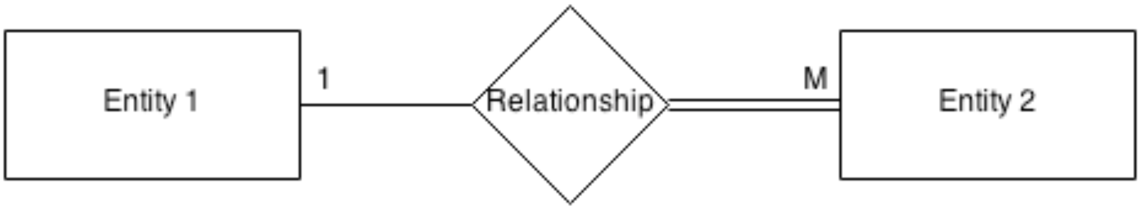
Any diagrams must be created using www.draw.io and the Chen notation must be used. The diagrams must be included in this word document for submission. When you have your diagram drawn using draw.io, this can be downloaded as a png image to include in this Word document via:

```
File > Download As > Portable Network Graphic (.png)
```

You should also save it as a .xml file for your own reference.

It is expected that the diagrams will include Entities, Relationships, Cardinality and Optionality as in the sample below. To draw these diagrams, the "Rectangle", "Rhombus", "Line", "Link" and "Simple Text" elements in the General section of www.draw.io are all that is needed. E

Try drawing this sample to get you started:



Description

A publishing company produces scientific books on various subjects. Authors, who can specialise in numerous subject areas, write the books. The company employs editors who, not necessarily being specialists in a particular area, each take sole responsibility for editing one or more publications. A publication covers essentially one of the specialist subjects and is normally written by a single author. When writing a particular book, each author works with on editor, but may submit another work for publication to be supervised by other editors. To improve their competitiveness, the company tries to employ a variety of authors, more than one author being a specialist in a particular subject.

How to approach this practical

1) Try to draw the entity example above with Draw.io
2) Read the Bus Company example carefully in lecture 5
3) Perform each question in sequence do not jump ahead

[Question 1 to 7 are explained in further detail in the slides for Lecture 4/5]

Q1. Identify all the **potential** entities required for designing a database for the above problem.

A1. Briefly, an entity correlates roughly to any real-world object that we are trying to model and store data about. Further, they generally correspond to nouns when trying to initially pinpoint them in a client's specification for a database (DB). In this case study, a list of potential entities would include the following:

- Company
- Book
- Subject
- Author
- Subject Area
- Editor
- Specialist
- Publication
- Work
- Competitiveness

Q2. Remove duplicate entities. For each entity you remove from Q1, include an explanation of why you decided to remove it

A2.

- ~~Company~~ -- Removed because it is not really an entity; rather, it is the system on which entities will be based
- Book – This will be a central entity in our DB
- ~~Subject~~ - Removed because it is thought to best be captured as an attribute of

‘Book’, and not an entity in and of itself due to its lack of attributes

- Author – This is deemed a necessary and focal entity for our DB
- ~~Subject Area~~ – Removed as it is a duplicate of ‘Subject’, which was previously considered ineligible as a candidate entity
- Editor – This should be retained as a entity as it is an entirely separate object on which data needs to be stored
- ~~Publication~~ – Duplicate of ‘Book’
- ~~Work~~ – Duplicate of ‘Book’ or ‘Publication’
- ~~Specialist~~ – Duplicate of ‘Author’
- ~~Competitiveness~~ - Although a noun, it is certainly not relevant to the design of our DB

Q3. List the attributes for each entity

A3.
It is now timely to list the attributes of each entity, where an attribute is simply a property of an entity that is relevant to the application in question.

1. Book (ISBN_10, ISBN_13[1], Title, Subject, Language, in_Stock, Number of pages, Format (Paperback/Hardcover/Audio/E-book), Dimensions, Retail Price, Bestsellers_rank, Author_ID, Editor_ID)
2. Author (Author_ID, Name_Author, Address_A, City_A, State_A, SSN, Phone, e-mail, A_Biography)
3. Editor (Editor_ID, Name_Editor, Address_E, City_E, State_E, SSN, Phone, e-mail, E_Biography)

[1] ISBN 13 is a newer version 13-digit version of the book identifier convention, and should be included for books published after its inception (January 1, 2007)

Q4. For each entity outline its primary key

A4.

A primary key is the combination of one or more column values in a table that make a row of data unique within the table. Aside from allowing us to join tables, primary keys prevent entry of duplicate records (Stephens & Plew, 2001). The primary keys of each chosen entity are underlined below, as per ERD conventions:

1. Book (ISBN_13, ISBN_10, Title, Subject, Language, in_Stock, Number of pages, Format (Paperback/Hardcover/Audio/E-book), Dimensions, Retail Price, Bestsellers_rank, *Author_ID, Editor_ID*)
2. Author (Author_ID, Name_Author, Address_A, City_A, State_A, SSN, Phone, e-mail, A_Biography)
3. Editor (Editor_ID, Name_Editor, Address_E, City_E, State_E, SSN, Phone, e-mail, E_Biography)

Q5. Define the Relationships. A section of an ER diagram drawn using www.draw.io will be required for each – see example above.

A5. (See next page)

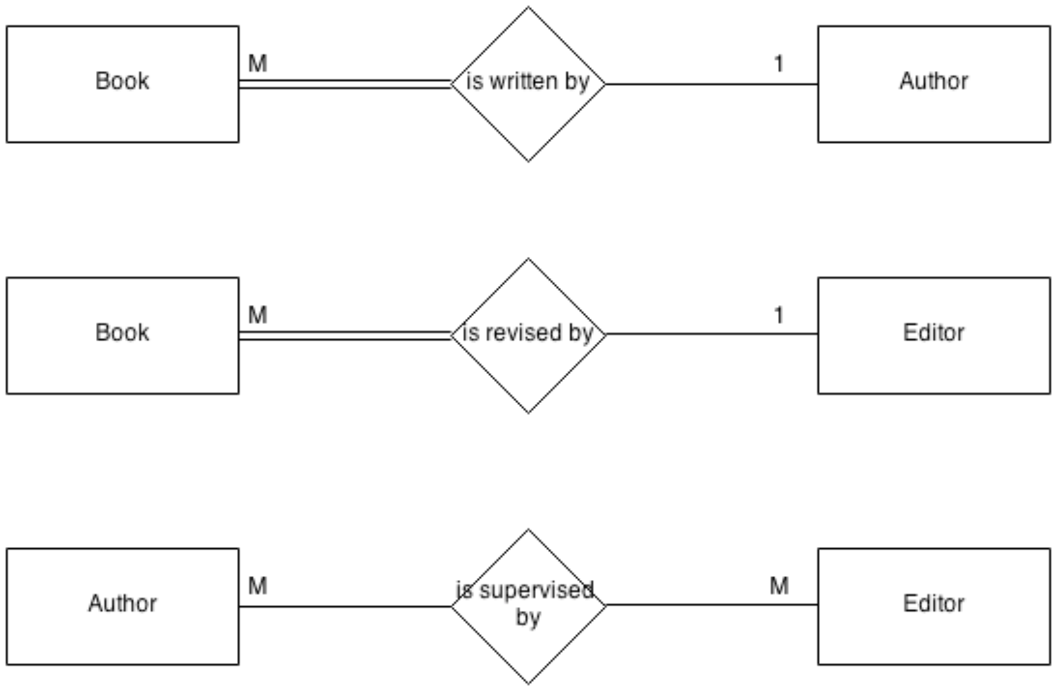


Figure 1. Overview of relationships between entities

The first relationship observed is between Book and Author. Specifically, the ‘Book’’s relationship with ‘Author’, is that they are written by same. Equally, the ‘Author’ is responsible for writing the ‘Book’.

Next, the ‘Book’ entity is related to ‘Editor’ as books must necessarily undergo the iterative and recursive process of editing before becoming widely available/commercialized. Likewise, the ‘Editor’ is the agent responsible for this revision of the ‘Book’.

Finally, the ‘Author’ entity is related to the ‘Editor’ entity because the former is supervised throughout the writing/editing process by the latter.

Q6. Describe (IN WORDS) the cardinality and optionality of the relationships

A6.

To parse the relationship between ‘Book’ and ‘Author’, the student maintains that a many-to-one relationship is appropriate. First, we must begin by explicitly stating our assumption that a book can have a sole contributor/author. Also, an author can most definitely write more than one book. Thus, we get a many-to-one relationship. Furthermore, a mandatory-optional relationship is derived. To wit, a book must have an author, but an author does not necessarily need to have a book. The latter statement is an inference predicated on the assumption that the publishing house recruits authors regularly (to author its books) and when they recruit new authors (by 'new', we mean that the author has not written a book before for that particular publishing house), then logically that author will have not titles (for that publisher) under its belt.

The conceptualization of the relationship between Book and Editor, is that of a many-to-one relationship. A book can be assigned only to one editor, as is usually the case in book editing due it being quite a tedious and lengthy process, and cognizant of the fact that many hours and a high level of personal knowledge or familiarity of the book is necessary for the tweaking procedures. Thus, this is an assumption utilized for the purpose of the task at hand. Regarding the editor->book relation, one editor, over its lifetime (or career as an editor) will edit many books, and this is rather intuitive. Apropos optionality, the relationship should be denoted as mandatory-optional, based on the notion that a book, in order to commercially viable, necessitates an editor to make corrections, and make it appealing for public consumption. Likewise, the HR department of the company will most likely recruit editors on an occasional basis, and thus we may have an editor who has yet to edit a book for the company, thus making entailment in this direction, optional.

The Author-Editor relationship can be aptly described as a many-to-many relationship. Briefly, an author is supervised by many editors, because they will be invariably have to work with different editors for various books written by them, and also an editor will supervise many different authors the course of its career, and also while considering possibly different authors as stated in the specification. Further to this, the relation will be optional-optional following our previous assumptions that new editors and authors can be introduced into the company without yet finding their feet, or a respective author and editor, for that matter.

Q7. Remove redundant relationships. For each relationship you remove, explain why you think that relationship is redundant.

A7.

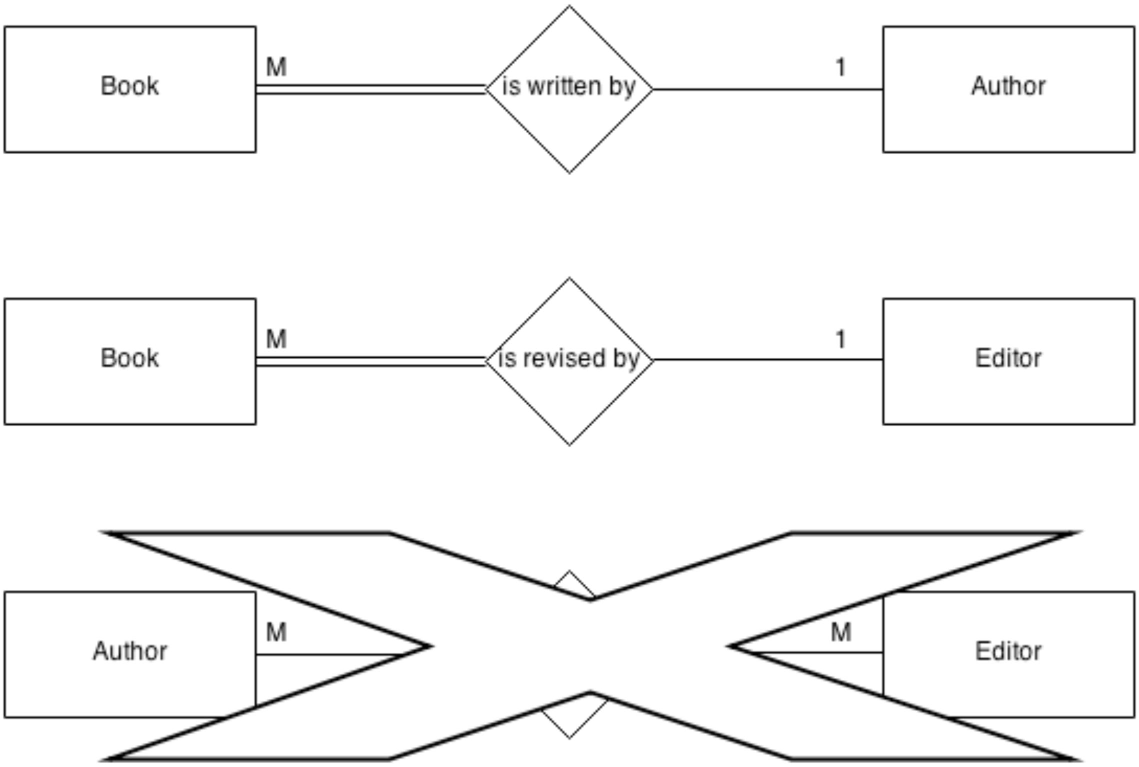


Figure 2. Removing redundant relationships

The Author-Editor relationship is deemed redundant following my considered opinion below:

Since Author and Editor both necessarily come into contact due to working on a book together, it is reasonable to assume that they can be tied together by utilizing one of their IDs as an attribute in Book. After all, we are dealing in relational DBs. This would work well given that both Book-Author and Book-Editor have a many-to-one relationship and so data can easily be retrieved by means of a foreign key in that entity (Book).

Finally, Author-Editor shows a many-to-many relationship in our ER model, and this can usually conceal an intermediate entity. In this case, the Author-Editor relationship can be replaced by an Employee entity, invoking some enhanced ER modelling concepts, which will be discussed in Q9.

Q8. Draw a complete ER diagram combining all of the work you have done above.

A8.

So far, our ER diagram looks like this:

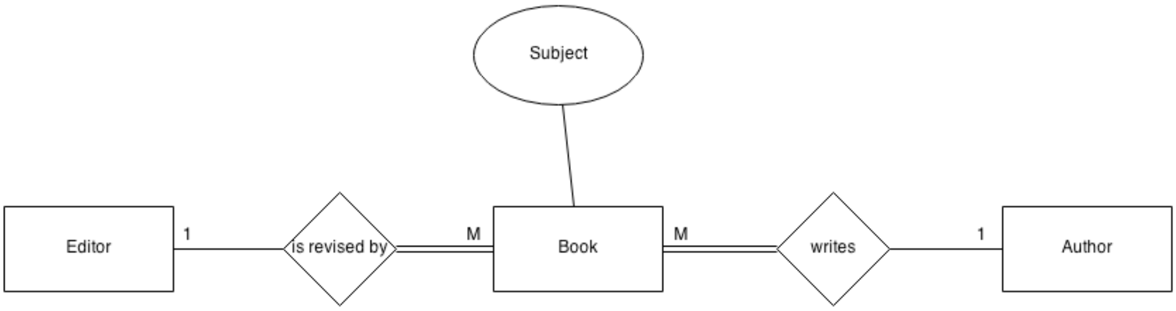


Figure 3. Preliminary Entity-Relationship model

Yet, it is not complete…(See final question)

The apparent chasm trap (m:n) that occurs in this diagram – where the model suggests the existence of a relationship between entity types, but this pathway is obscured (between Authors and Editors) – is not fatal for our DB. Since, author/editor each work with one book (based on assumptions) and thus one author/editor at any one time, it will be easy to identify author-editor collaborations by utilizing *Author_ID* and *Editor_ID* as foreign keys in our Book entity (see Q2). Not only that, we in the previous step, we removed the Author-Editor relationship, demoting it as redundant, since we can take other measures to gain information about which author-editor pair worked on a particular book together. Regardless, it is prudent to elucidate and address chasm traps, which we will elaborate on in the final question.

Q9. Check any M-M relationships to see if they are hiding a new entity. If so, fix your ER diagram to reflect the change and include a brief explanation of why you changed your diagram.

A9.

Our next-to-last ER diagram in Question 8 was revised to reflect the changes we purported to make to the Author-Editor relationship. Essentially, we aim to introduce a superclass, Employee, as a new entity construct, to treat authors and editors as distinct subclasses that need to be represented, but yet are also member of a superclass. This is helpful for our current purposes, as we can maximize differences between authors and editors by identifying their distinct characteristics (Dr. Scanlon, personal communication, 2014) and simultaneously contain all the general/common attributes pertinent to both authors and editors (e.g. name, address, SSN) in the superclass. An explicit assumption is made that these subclasses of Author and Editor will be mutually exclusive, that is, an author may not be an editor and an editor may not be an author, and this is denoted in the diagram by using the ‘d’ character, standing for ‘disjoint’. Further, we will assume that this is a medium-to-small size or even start-up publishing firm, and so its employees are only either authors OR editors. This fact is represented in the mandatory relationship between the superclass entity, Employee, and its subclasses.

Now that we have created a superclass to hold all the common attributes of Author and

Editor, we need to re-evaluate these entities' original (and now, distinctive) properties. The Author entity may now hold 'Author_ID' as its primary key, along with other information. Similarly, the Editor entity will contain 'Editor_ID' as its primary key, and along with other attributes. Finally, we should include Employee_ID as a foreign key in both the Author and Editor entities to ensure a link between these tables from subclasses to the superclass (Employee).

The final ER diagram is depicted below. Please note that because draw.io did not feature the requisite symbol for representing subclasses (inverted U shape), I substituted this for a U-shaped arrow instead.

Note, that there also exists a relationship between Book and Employer, and cardinality nor optionality representations were not included here as they are made explicitly for its subclasses' relations to Book (and as based on Lecture Notes). Relevant attributes are also included for clarification of entities.

Our revised entity list may now resemble the following:

1. Book (ISBN_10, ISBN_13, Title, Subject, Language, in_Stock, Number of pages, Format (Paperback/Hardcover/Audio/E-book), Dimensions, Retail Price, Bestsellers_rank, *Author_ID, Editor_ID*)
2. Employee (Employee_ID, Name, Address, City, State, SSN, Biography, DOB,..)
3. Author (Author_ID, Salary, Speciality, *Employee_ID*)
4. Editor (Editor_ID, Salary, Bonus, *Employee_ID*)

See next page for final ER diagram.

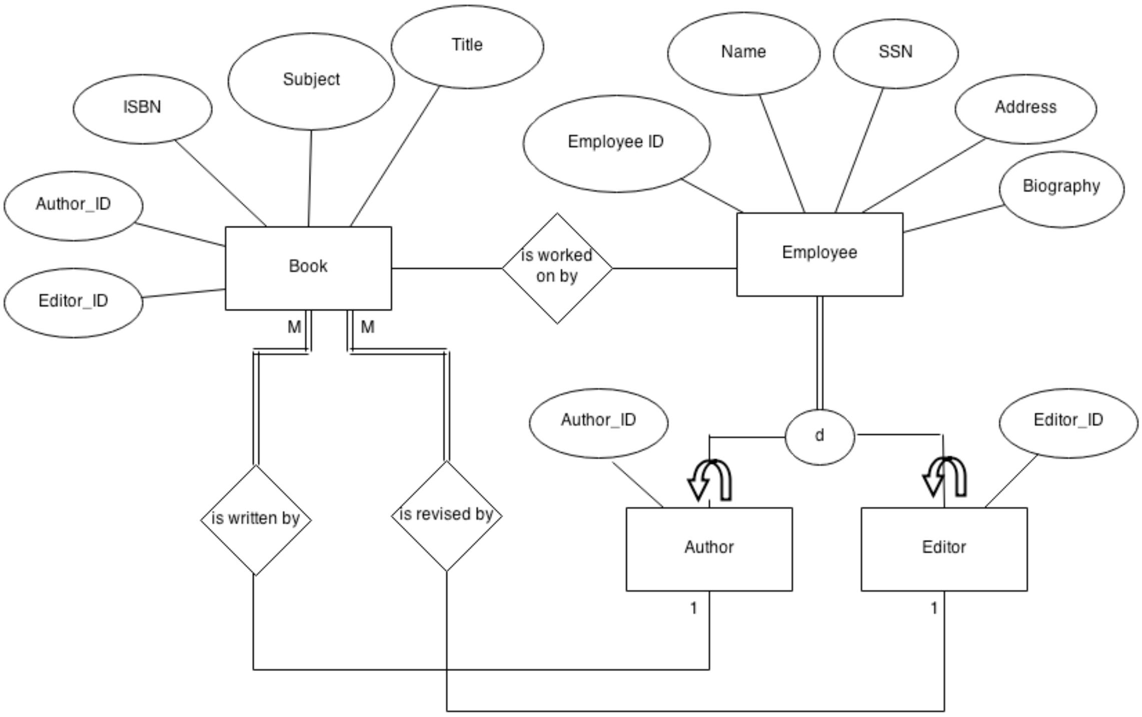


Figure 4. Final Entity-Relationship model for case study