

Student ID-

University of Essex Online

[Introduction to Data Modelling]

[Unit 4]

960 Words.

Contents

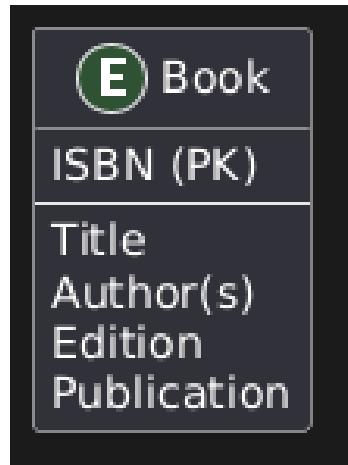
Task 1 – Entity-Relationship Modelling	3
Task 2 – Normalisation to Third Normal Form (3NF)	8
References.....	12

Task 1 – Entity-Relationship Modelling

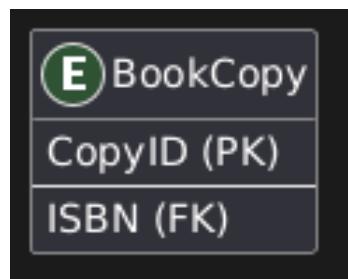
Using the scenario provided, it is possible to identify its entities and attributes and also determine the relationships between them. The scenario is analysed in the order presented, starting with bullet point number one:

In bullet point one, the scenario states that the library has multiple books, each with a unique **ISBN**, **Title**, **Author(s)**, **Edition**, and **Publication Year**.

From this, the **Book** entity is identified, containing the highlighted information above and ISBN as the primary key. Although the library has a multitude of books, each row in the entity represents one book.



Bullet point two implies that a single book can have multiple copies. Thus, a new entity **BookCopy** is created. The attributes will be **CopyID** (Primary key) and **ISBN** (Foreign key), establishing a one-to-many relationship between **Book** and **BookCopy**. This design allows different copies of the same book to be tracked separately.



By separating **BookCopy** from **Book**, the chance of redundant data being entered into the table repeatedly is reduced. This improves data integrity by reducing chances of human error.

Bullet point three defines library members. And therefore, a **Member** entity is created with the attributes **Name**, **Address**, **PhoneNumber**, and **Email** with **MemberID** being the primary key so that each member can be identified individually.

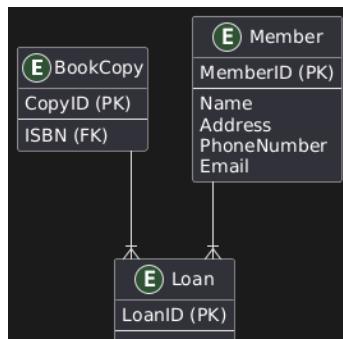
E	Member
	MemberID (PK)
	Name
	Address
	PhoneNumber
	Email

Similar to member, bullet point four creates the **Staff** entity, with the attributes **StaffID** (Primary key), **Name**, **Role**, and contact details, referring to their **PhoneNumber** and **Email**.

E	Staff
	StaffID (PK)
	Name
	Role
	Email
	PhoneNumber

Bullet point five outlines the borrowing rules of the library: a member may borrow multiple books, though a book copy may only be on loan to one member at a time. To model this, a **Loan** entity is created with the primary key **LoanID**.

This results in a one-to-many relationship between **Member** and **Loan** as a member can have multiple loans, and a one-to-many relationship between **BookCopy** and **Loan** as a book copy can be loaned multiple times, but never at the same time.



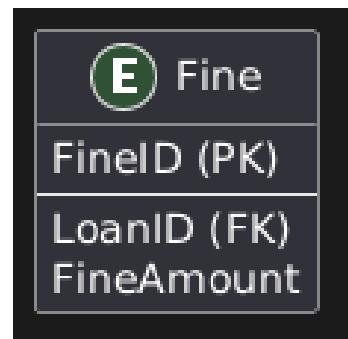
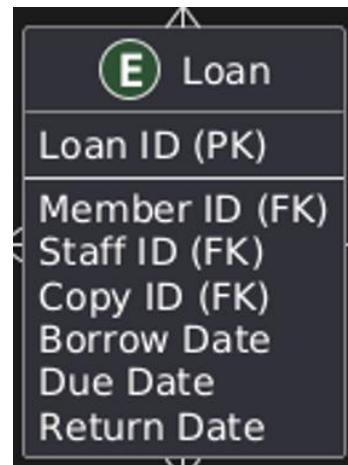
Bullet point six provides further information on the new **Loan** entity, appending it with the attributes **BorrowDate**, **DueDate**, and **ReturnDate**, and the foreign keys of **MemberID**, **StaffID**, and **CopyID**.

The choice of treating **CopyID** as a singular entity within **Loan**, is to handle the possibility that books may have policies restricting their maximum loan period (due to demand, for instance).

Finally, bullet point seven outlines how overdue books are tracked and that a fine will be imposed if a book is returned late. Tracking this requires a new entity, **Fine**.

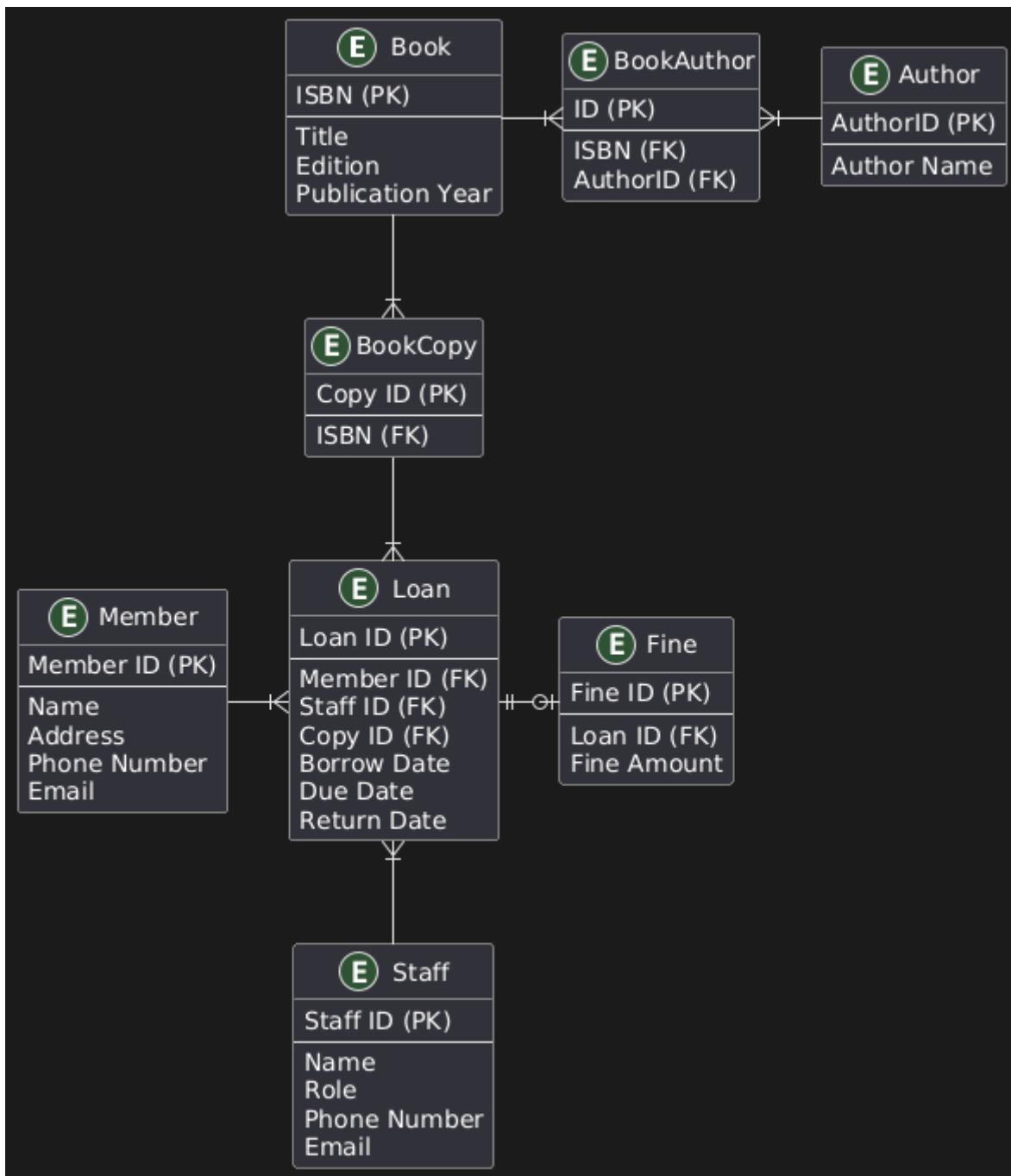
The **Fine** entity will be connected to **Loan** via a one-to-zero-or-none relationship, as a loan will not always result in a fine.

The keys will be a primary key of **FineID** with a foreign key **LoanID**, allowing the database to track what loan the fine is linked to. Finally, **FineAmount** will keep a record of how much the fine was.



Authors are potentially sharing a many-to-many relationship (multi-valued) with **Book**, since there can possibly be multiple authors for a single book. Thus, resolving it into a separate entity, **BookAuthor** and further into **Author** prevents potential conflicts.

Finally, the **Staff** attribute will have a one-to-many relationship with **Loan** as there can be many loans but only one staff member who signed it off. Here is the completed entity relationship diagram:



Task 2 – Normalisation to Third Normal Form

(3NF)

(Not included in word count)

UNF	1NF	2NF	3NF
<u>LoanID</u> (PK) CopyID BookISBN BookTitle BookAuthors BookEdition BookPublicationYear MemberName MemberAddress MemberPhoneNumber MemberEmail StaffName StaffRole StaffEmail StaffPhoneNumber BorrowDate DueDate ReturnDate FineAmount	<u>LoanID</u> (PK) CopyID MemberName MemberAddress MemberPhoneNumber MemberEmail StaffName StaffRole StaffEmail StaffPhoneNumber BorrowDate DueDate ReturnDate FineAmount <u>Book</u> BookISBN (PK) BookTitle BookEdition BookPublicationYear <u>BookCopy</u> CopyID (PK) BookISBN (FK) <u>Author</u> AuthorID (PK) AuthorName <u>BookAuthor</u> BookISBN (FK) AuthorID (FK) <u>Member</u> MemberID (PK) MemberName MemberAddress MemberPhoneNumber MemberEmail <u>Staff</u> StaffID (PK) StaffName StaffRole StaffEmail StaffPhoneNumber <u>Loan</u> LoanID (PK) CopyID (FK) MemberID (FK) StaffID (FK) BorrowDate DueDate ReturnDate FineAmount	<u>Book</u> ISBN (PK) Title Edition PublicationYear <u>BookCopy</u> CopyID (PK) BookISBN (FK) <u>Author</u> AuthorID (PK) AuthorName <u>BookAuthor</u> BookISBN (FK) AuthorID (FK) <u>Member</u> MemberID (PK) Name Address PhoneNumber Email <u>Staff</u> StaffID (PK) Name Role Email PhoneNumber <u>Loan</u> LoanID (PK) CopyID (FK) MemberID (FK) StaffID (FK) BorrowDate DueDate ReturnDate	<u>Book</u> ISBN (PK) Title Edition PublicationYear <u>BookCopy</u> CopyID (PK) BookISBN (FK) <u>Author</u> AuthorID (PK) AuthorName <u>BookAuthor</u> BookISBN (FK) AuthorID (FK) <u>Member</u> MemberID (PK) Name Address PhoneNumber Email <u>Staff</u> StaffID (PK) Name Role Email PhoneNumber <u>Loan</u> LoanID (PK) CopyID (FK) MemberID (FK) StaffID (FK) BorrowDate DueDate ReturnDate <u>Fine</u> FinelD (PK) LoanID (FK) FineAmount

By analysing the scenario, an unnormalized form (UNF) is established. At this stage, all data relating to a book loan is placed into the **Loan** entity. This structure contains multi-valued (non-atomic) values and redundancy, which is not suitable for database management. Therefore, normalization is necessary, starting with first normal form.

To Achieve 1NF, all entities in the diagram must be Atomic, and non-repeating. In layman's terms, a field should contain only one fact (Bagui, 2022, ch. 3.2).

Since Author(s) has a possibility to be multiple values for a book, an entity of Authors needs to be created to resolve this issue. This entity will simply consist of the **AuthorID** and **AuthorName** attributes.

However, to complicate the matter, while it is not stated within the scenario that Authors can create multiple books, common knowledge will lead us to infer this fact and as such, **Book** and **Author** are in fact a many-to-many relationship.

To resolve this, an intermediary (pivot) table can be introduced to model the link between **Book** and **Author**. This table will consist of the foreign keys of **AuthorID** referencing the **Author** table and **BookISBN** referencing the Books Table.

Next, second normal form (2NF) requires the table to be in 1NF and that all non-key attributes are dependent on the primary key (Bagui, 2022, ch. 3.6). In the 1NF structure, many attributes do not rely on their primary key.

For instance, within the **Loan** table, the details about staff and member do not rely on **LoanID**, as such they are extracted from the **Loan** table into a table called **Staff** and a table called **Member**. In these entities, the **StaffID** will act as the primary key in **Staff**, and **MemberID** in **Member**. These will be added as foreign keys into the **Loan** table.

Furthermore, **BookCopy** is separated from **Book** with the primary key of **CopyID**, since all details relating to the book rely on **ISBN** and not **LoanID**. In later iterations of the database, more attributes may be added to **BookCopy**, such as purchase date or condition.

Finally, to achieve third normal form (3NF), all transitive dependencies must be removed by ensuring all non-key attributes depend only on the primary key, not on other non-key attributes (Connolly and Begg, 2020, ch. 14.8).

In this stage, data related to fines is separated from the **Loan** entity, because **FineAmount** depends on when a loan becomes overdue rather than on the loan itself. Therefore, a new entity, **Fine**, is created with the primary key **FineID** and foreign key **LoanID**. The database is now normalized to 3NF.

References

- Bagui, S.S. (2022) *Database design using entity-relationship diagrams*. 3rd edn. Boca Raton, FL: Auerbach Publications.
- Connolly, T. and Begg, C. (2014) *Database systems: a practical approach to design, implementation, and management, Global Edition*. 6th edn. Harlow: Pearson Education.
- Fong, J. and Yan, K.W.T. (2021) *Information systems reengineering, integration and normalization: heterogeneous database connectivity*. 4th edn. Cham: Springer International Publishing.
- OpenAI (2026) *ChatGPT* (version 5) [Online]. Available at: <https://chat.openai.com/> (Accessed: 9 February 2026). Used for guidance on referencing and citation formatting.
- Watt, A. (2014) *Database design*. 2nd edn. [Place of publication not identified]: BCcampus.