Contents

1. Introduction

The College Library System report outlines the design of a database for information retrieval with the aim of developing a computerised system to maintain the library's daily work. The system supports the management of the resources such as their representation and circulation. The report will include the conceptual design, comprising of mapping the entities in the conceptual model to the relational model, as well as transformation to normalised relations. As such the report will include the following:

- Assumptions made during the design phase
- Entity-Relationship Model
- Relational Schema
- Normalisation of the design up to third normal form

2. Assumptions

1. The system will be designed for a particular college library
2. There are two types of members, students and staff
3. Each member has a library card containing their ID number
4. Library cards expire after a certain amount of time
5. Members can borrow and return resources
6. The resources are books, videos, DVDs and CDs
7. The number of copies per resource can vary
8. The library contains 3 floors
9. Each floor has shelves where the resources are stored
10. Each resource belongs to a class
11. Identical copies of each resource are stored on the same floor and shelf
12. Loan periods for resources are either 2 days or 2 weeks
13. Some resources can only be used in the library (0 days)
14. Students can borrow 5 resources at a time
15. Staff can borrow 10 resources at a time
16. Members can only borrow 1 copy of the same resource
17. Loans become overdue if they are not returned by the due date
18. Loans that have been returned become previous loans
19. The popularity of a resource is based on the number of times it has previously been loaned
20. Members are fined $1 per day for delays
21. Fines are calculated from the due date of the resource to the date the member makes the payment
22. Fines can be paid either partially or in full
23. Fine payments will be deducted starting from the member's oldest fine
24. Members are suspended if their fine exceeds $10
25. Suspensions can be reversed if fines are paid and overdue loans are returned
26. Students can enrol in a class only once without resitting

3. Entity-Relationship Model

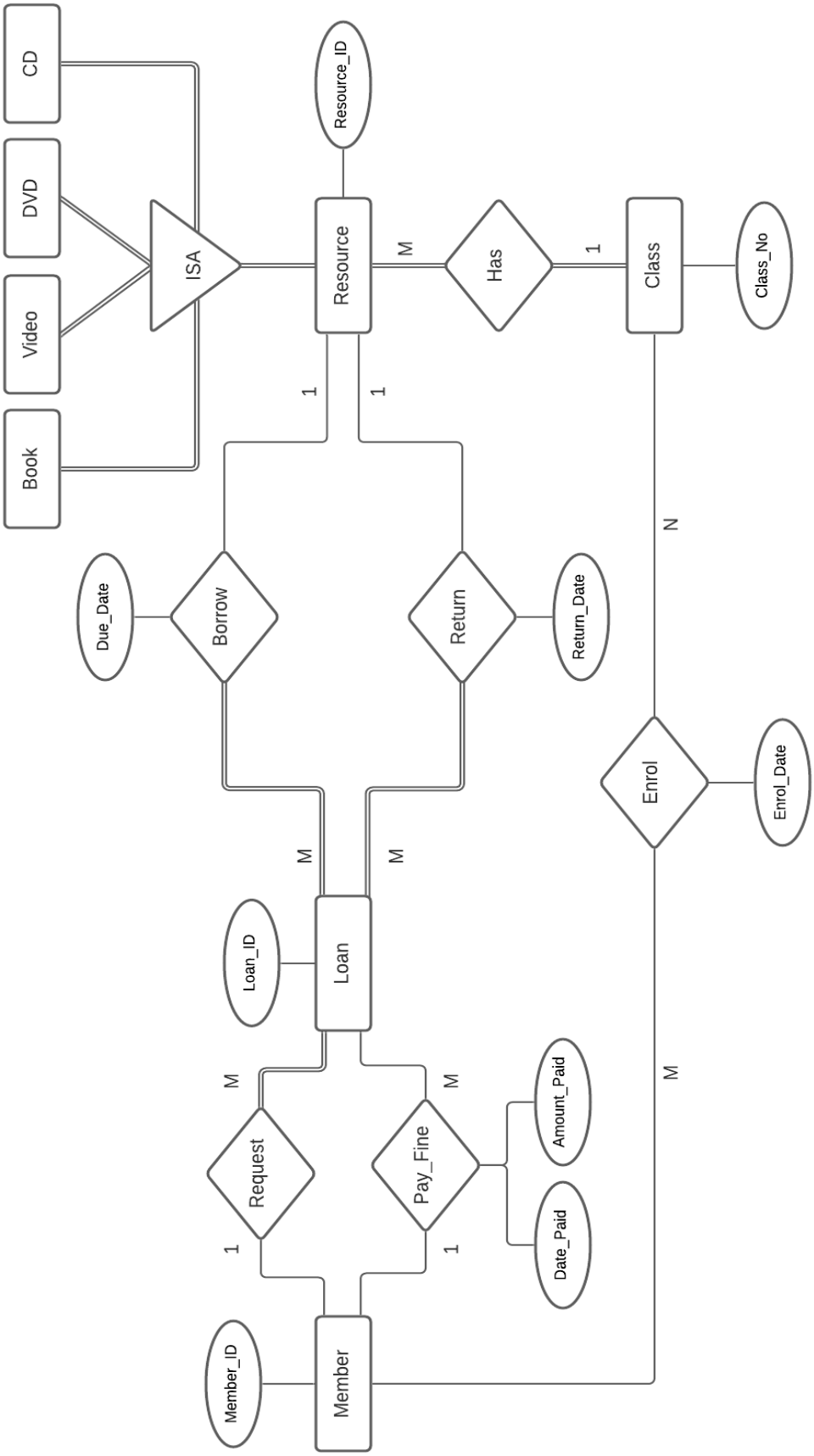

Figure 1 Entity-Relationship Diagram

4.1 Entities and Attributes

Member (Member_ID, Member_Type, Member_Name, Member_Email, Copies_Allowed, Expiry_Date)

Member is a strong entity with Member_ID as its primary key. The Member_Name and Member_Email attributes include each member's name and email address. To improve search functionality within the database, Member_Name was not used as a composite attribute consisting of first name and surname. Copies_Allowed is an attribute containing the maximum number of copies a member can have. This is dependent on Member_Type (student or staff). Expiry_Date is the date when a member's library membership expires.

Resource (Resource_ID, Resource_Type, Resource_Title, Total_Copies, Loan_Period, Location (Floor_No, Shelf_No))

Resource is a strong entity with Resource_ID as its primary key. Resource is a superclass of Book, Video, DVD and CD. Originally, Book_Title, Video_Title, DVD_Title and CD_Title were used as attributes but after generalisation, the Resource_Title attribute was used instead. Resource_Type is either Book, Video, DVD or CD. Although Resource_Type is not required for the loaning and returning of items, it is being included to make searching easier for members. Total_Copies is the number of copies each resource has. Loan_Period is the maximum amount of time a resource may be borrowed. Location is a composite attribute consisting of Floor_No and Shelf_No of the resource.

Book (Author, Publisher, Publish_Date, Edition)

Book is a subclass of the Resource entity. The attributes Author, Publisher, Publish_Date and Edition include information relating to each specific book.

Video (Director, Release_Date)

Video is a subclass of Resource. The attributes Director and Release_Date include information relating to each specific video.

DVD (Director, Release_Date)

DVD is a subclass of Resource. The attributes Director and Release_Date include information relating to each specific DVD.

CD (Artist, Release_Date)

CD is a subclass of Resource. The attributes Artist and Release_Date include information relating to each specific CD.

Loan (Loan_ID, *Member_ID, Resource_ID*)

Loan_ID is the primary key of the Loan entity with Resource_ID and Member_ID as its foreign keys.

Class (Class_No, Class_Name)

Class is a strong entity with Class_No as its primary key. Class_Name is the attribute containing the subject name.

4.2 Relationships

Request

Request is a relationship between the Member and Loan entities. The cardinality between the two entities is one to many. Therefore, a member can request many loans but each loan only has one member.

Borrow (Due_Date)

Borrow is a relationship between the Loan and Resource entities. The cardinality between the two entities is one to many. Therefore, a resource can have many loans but each loan only belongs to one resource. Due_Date is the date by which the resource must be returned. Issue_Date was not used as the modelling of the database will be more convenient with the Due_Date attribute.

Return (Return_Date)

Borrow is a relationship between the Loan and Resource entities. The cardinality between the two entities is one to many. Therefore, a resource can be returned many times but for each loan, there is only one return.

Enrol (Enrol_Date)

Enrol is a relationship between the Class and Member entities. The Member entity has partial participation as not every member is enrolled. The cardinality between the two entities is many to many. Enrol_Date is the date on which a member enrols in a class.

Pay_Fine (Date_Paid, Amount_Paid)

Pay_Fine is a relationship between the Member and Loan entities. The Member entity has partial participation as not every member will pay fines. The cardinality between the two entities is one to many. Date_Paid is the date on which the fine was paid. Amount_Paid is how much was paid.

4. Relational Schema

5.1 Mapping

Member (Member_ID, Member_Type, Member_Name, Member_Email, Copies_Allowed, Expiry_Date)

The relation member remains the same.

Resource (Resource_ID, Resource_Type, Resource_Title, Total_Copies, Loan_Period, *Class_No,* Floor_No, Shelf_No)

The cardinality between Resource and Class is many to one. Class_No is a foreign key from the relation Class.

Book (Resource_ID, Author, Publisher, Publish_Date, Edition)

The attribute Resource_ID is the primary key and is mapped from the relation Resource.

Video (Resource_ID, Director, Release_Date)

The attribute Resource_ID is the primary key and is mapped from the relation Resource.

DVD (Resource_ID, Director, Release_Date)

The attribute Resource_ID is the primary key and is mapped from the relation Resource.

CD (Resource_ID, Artist, Release_Date)

The attribute Resource_ID is the primary key and is mapped from the relation Resource.

Loan (Loan_ID, *Member_ID, Resource_ID, Due_Date, Return_Date, Date_Paid, Amount_Paid*)

The cardinalities between Loan and Resource for the Borrow and Return relationships are many to one. The attributes Due_Date and Return_Date will be used as foreign keys within the relation Loan. The cardinality between Loan and Member for the Pay_Fine relationship is many to one. Therefore, the attribute Date_Paid and Amount_Paid can be used as a foreign key within the relation Loan. Member_ID and Resource_ID are also foreign keys from the Member and Resource entities respectively.

Class (Class_No, Class_Name)

The relation Class remains the same.

Enrol (*Member_ID*, *Class_No*, Enrol_Date)

The cardinality between Member and Class is many to many. Therefore, a separate relation (Enrol) was created. Member_ID are Class_No are both primary and foreign keys.

5.2 Finalised Relations

The full relational schema can be seen as follows:
Member (Member_ID, Member_Type, Member_Name, Member_Email, Copies_Allowed, Expiry_Date)
Resource (Resource_ID, Resource_Type, Resource_Title, Total_Copies, Loan_Period, *Class_No,* Floor_No, Shelf_No)
Book (Resource_ID, Author, Publisher, Publish_Date, Edition)
Video (Resource_ID, Director, Release_Date)
DVD (Resource_ID, Director, Release_Date)
CD (Resource_ID, Artist, Release_Date)
Loan (Loan_ID, *Member_ID, Resource_ID*, *Due_Date, Return_Date, Date_Paid, Amount_Paid*)
Class (Class_No, Class_Name)
Enrol (*Member_ID*, *Class_No*, Enrol_Date)

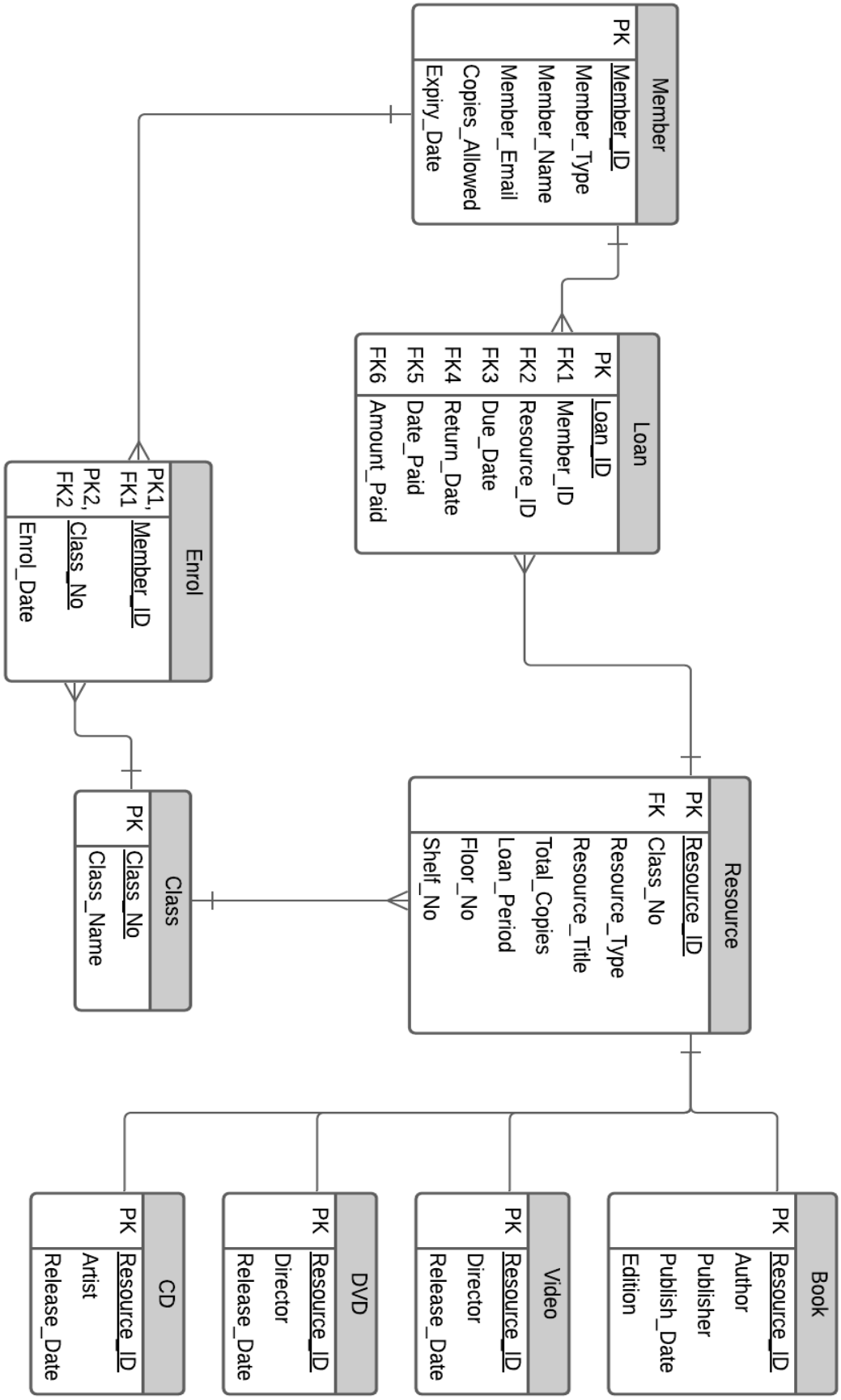

Figure 2 Relational Schema Diagram

5. Normalisation

Normalisation is used in order to remove redundancy and optimise the database design.

6.1 Universal Relations

U (Member_ID, Member_Type, Member_Name, Member_Email, Copies_Allowed, Expiry_Date, Resource_ID, Resource_Type, Resource_Title, Total_Copies, Loan_Period, Class_No, Floor_No, Shelf_No, Author, Publisher, Publish_Date, Edition, Director, Release_Date, Artist, Loan_ID, Due_Date, Return_Date, Date_Paid, Amount_Paid, Class_Name, Enrol_Date)

6.2 Functional Dependencies

Member_ID → Member_Type, Member_Name, Member_Email, Copies_Allowed, Expiry_Date

Resource_ID → Resource_Type, Resource_Title, Total_Copies, Loan_Period, Floor_No, Shelf_No, Author, Publisher, Publish_Date, Edition, Director, Release_Date, Artist

Member_ID, Resource_ID, Due_Date → Loan_ID

Loan_ID → Return_Date, Date_Paid, Amount_Paid

Class_No → Class_Name

Member_ID, Class_No → Enrol_Date

The relations are now:
Member (Member_ID, Member_Type, Member_Name, Member_Email, Copies_Allowed, Expiry_Date)

Resource (Resource_ID, Resource_Type, Resource_Title, Total_Copies, Loan_Period, Floor_No, Shelf_No, Author, Publisher, Publish_Date, Edition, Director, Release_Date, Artist)

Borrow (Member_ID, Resource_ID, Due_Date, Loan_ID)

Loan (Loan_ID, Return_Date, Date_Paid, Amount_Paid)

Class (Class_No, Class_Name)

Enrol (Member_ID, Class_No, Enrol_Date)

6.3 First Form (1NF)

Member (Member_ID, Member_Type, Member_Name, Member_Email, Copies_Allowed, Expiry_Date)

Resource (Resource_ID, Resource_Type, Resource_Title, Total_Copies, Loan_Period, Floor_No, Shelf_No, Author, Publisher, Publish_Date, Edition, Director, Release_Date, Artist)

Borrow (Member_ID, Resource_ID, Due_Date, Loan_ID)

Loan (Loan_ID, Return_Date, Date_Paid, Amount_Paid)

Class (Class_No, Class_Name)

Enrol (Member_ID, Class_No, Enrol_Date)

The above relations are already in first normal form as there are no composite or multivalued attributes.

6.4 Second Form (2NF)

Member (Member_ID, Member_Type, Member_Name, Member_Email, Copies_Allowed, Expiry_Date)

Resource (Resource_ID, Resource_Type, Resource_Title, Total_Copies, Loan_Period, Floor_No, Shelf_No, Author, Publisher, Publish_Date, Edition, Director, Release_Date, Artist)

Loan (Loan_ID, Return_Date, Date_Paid, Amount_Paid)

Class (Class_No, Class_Name)

As the above relations only have one primary key, they are already in second normal form.

Borrow (Member_ID, Resource_ID, Due_Date, Loan_ID)

As Loan_ID depends on Member_ID, Resource_ID and Due_Date, it is already in second normal form.

Enrol (Member_ID, Class_No, Enrol_Date)

As Enrol_Date depends on both Member_ID and Class_No, it is already in second normal form.

6.5 Third Form (3NF)

Resource (Resource_ID, Resource_Type, Resource_Title, Total_Copies, Loan_Period, Floor_No, Shelf_No, Author, Publisher, Publish_Date, Edition, Director, Release_Date, Artist)

Borrow (Member_ID, Resource_ID, Due_Date, Loan_ID)

Loan (Loan_ID, Return_Date, Date_Paid, Amount_Paid)

Class (Class_No, Class_Name)

Enrol (Member_ID, Class_No, Enrol_Date)

The above relations are in third normal form as the attributes in all the relations are non-transitively dependent on their primary keys.

Copies_Allowed is transitively dependent on Member_ID because it directly depends on Member_Type. Therefore, it is necessary to decompose into two 3NF relations. The second relation can have Member_Type and the corresponding Copies_Allowed.

Member (Member_ID, Member_Type, Member_Name, Member_Email, Expiry_Date)

Copies (Member_Type, Copies_Allowed)

6. Conclusion

In conclusion, there are now seven relations that are normalised up to third normal form. The model is in third normal form as none of the attributes is transitively dependent on their primary keys.

Final relations after normalisation:
Member (Member_ID, Member_Type, Member_Name, Member_Email, Expiry_Date)
Resource (Resource_ID, Resource_Type, Resource_Title, Total_Copies, Loan_Period, Floor_No, Shelf_No, Author, Publisher, Publish_Date, Edition, Director, Release_Date, Artist)
Loan (Loan_ID, Return_Date, Date_Paid, Amount_Paid)
Class (Class_No, Class_Name)
Borrow (Member_ID, Resource_ID, Due_Date, Loan_ID)
Enrol (Member_ID, Class_No, Enrol_Date)
Copies (Member_Type, Copies_Allowed)

The model covers the specification for the College Library System. Some concepts could not be directly included at this stage of modelling as they are derived. These are:

- Fine
- Overdue Loan
- Current Loan
- Previous Loan
- Suspension
- Popularity
- Copies Available

These concepts will be introduced during the next stage where they will be calculated. This normalised model can now be implemented using the SQL database system.

7. Future Work

In the future, when using the design to implement the database in SQL, calculations can be used for the derived concepts.

If the Return_Date value is null, it is a current loan.

If Return_Date value is not null, it is a previous loan.

When the Due_Date of a current loan is surpassed, that loan becomes overdue.

Fine is $1 for each day delayed. It is calculated by subtracting Due_Date from Date_Paid for the same Loan_ID. Fine = (Date_Paid - Due_Date) x $1.

A member's total fine is calculated by adding all their fines together. If the fines exceed $10, they are suspended. Suspensions can be reversed if a member pays all their fines in full (bringing their total value to zero) and returns all overdue loans.

Copies Available = (Total_Copies - number of Resource_ID's in current loan).

Popularity is given by the number of times a Resource_ID appears in previous loans.

8. Bibliography

Lemahieu, W. (2018). *Principles of Database Management: The Practical Guide to Storing, Managing and Analyzing Big and Small Data*. Cambridge: Cambridge University Press.

Stockman, A. (2020). "Specification for coursework 1 + marking scheme". ECS740P: *Database Systems*. Available at: https://qmplus.qmul.ac.uk/course/view.php?id=15472 (Accessed: 11/10/2020).