Homework 3
DD1368 HT21
Rafael Bechara & Paula Hanna

Our database model

1. Published_Book (Title: String, Edition: String, publisher: String, DateOfPublish: Date)

2- Book(PhysicalBook_ID: Integer, Title: String, Edition: String, ISBN: String, language: String)

3- Damaged_Book(PhysicalBook_ID: integer, Damage: string)

4- Prequel_And_Sequel(PhysicalBook_ID: Integer,Title : String , prequel : String , sequel: String)

5- Book_Author(PhysicalBook_ID: integer, Author: string)

6- Book_Genre(PhysicalBook_ID: integer, Genre: string)

7- Student(StudentID : Integer,FirstName: string, LastName: string , Email: string, Program: string, Address: string)

8- Admin(AdminID: integer, FirstName: string, LastName: string, Email: string, Adress: string, Department: string, PhoneNumber: string)

9- Borrowed_Book(Borrowing_ID: Integer, DateOfBorrowing: Date, ExpiringDate: Date, ReturnDate: Date, PysicalBook_ID: integer, Student_ID: integer)

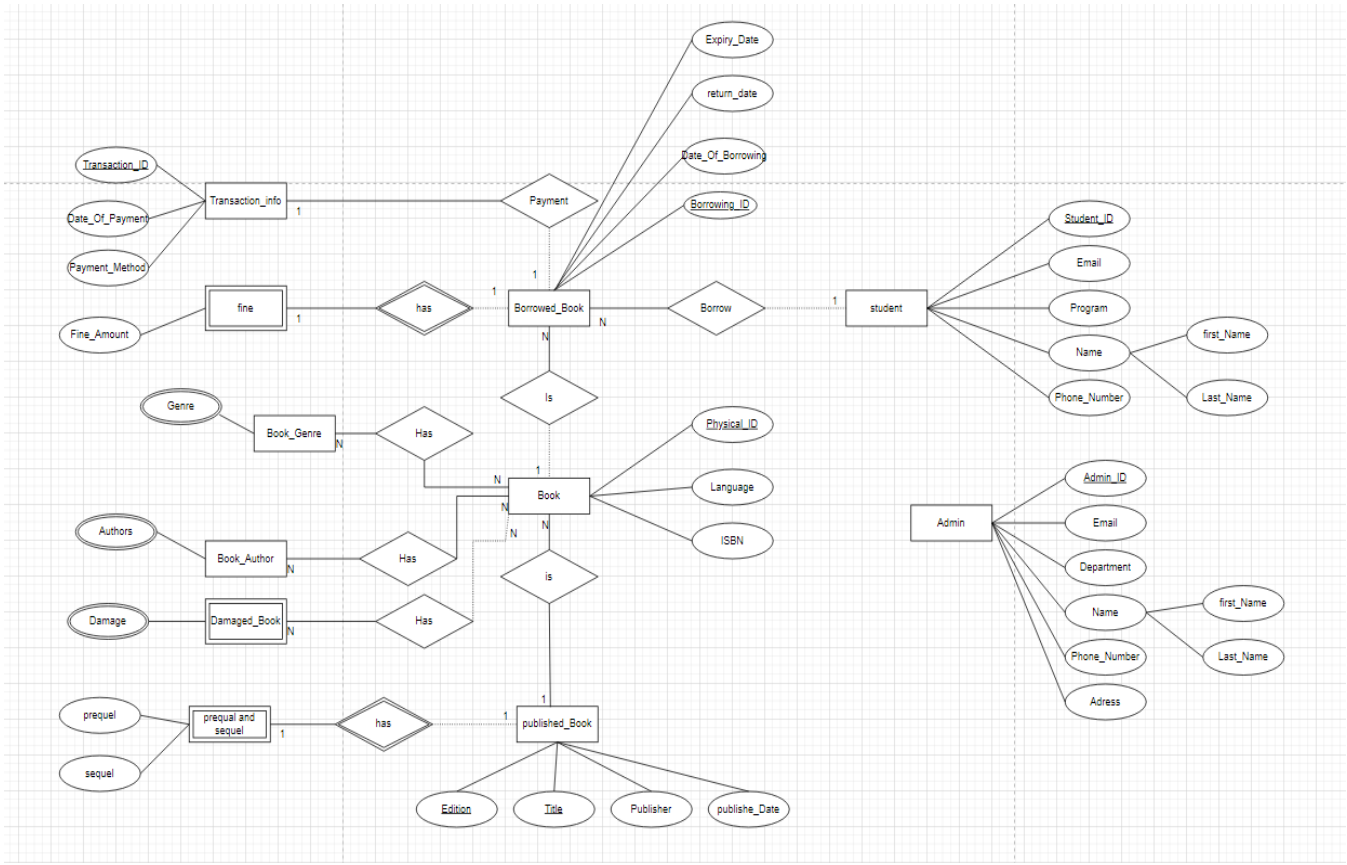
10- Fines(Borrowing_ID: integer, FineAmount: double)

11- Transaction_Information(Transaction_ID: integer, DateOFPayment: Date, PaymentMethod: string, Borrowing_ID)

Our model is already in BCNF due to our normalization of the database in homework 2 .

Homework 3
DD1368 HT21
Rafael Bechara & Paula Hanna

E/R diagram



Relation between Book and Borrowed_Book:

1) A Borrowed book is indeed a book and a copy of a book can be borrowed which means that it can be a borrowed book.
2) A copy of a book can exist without being borrowed but a borrowed book cannot exist without being a book from the first place. Which means that the existence of the Book entity is independent of the relation, while the existence of a BorrowedBook entity is totally independent on the relation.
3) One copy of a book can be borrowed many times.

Homework 3
DD1368 HT21
Rafael Bechara & Paula Hanna

According to that we chose to have the relation named “Is” between Book entity and BorrowedBook entity. It is one optional (Book) to many mandatory (BorrowedBook) type of relation (check 2,3) for explanation)

Relation between Book and published_Book:

1. A published book is indeed a book
2. The relation between book and published book have the mandatory trait on each side, because both entities are dependent on each other. A published book data can not exist in our database without having corresponding book data and vice versa .
3. one published book can have many books in our database, but a book can have only one published book instance.

According to that we chose to have the relation named “Is” between Book entity and PublishedBook entity. It is one mandatory (publishedBook) to many mandatory (Book) .

Relation between Book and Book_Author:

1. A book must have at least one author if not more.
2. The relation between book and book author has the mandatory trait on both sides, because both entities are dependent on each other, a book must have an author and vice versa.
3. One book can have many authors, and one author can have many books.

According to that we chose to have the relation named “Has” between Book entity and Book_Author entity and is many mandatory (Book_Author) to many mandatory (Book) .

Relation between Book and Book_Genre:

1. A book must have at least one genre if not more.
2. The relation between book and book genre has the mandatory trait on both sides, because both entities are dependent on each other, a book must have a genre and vice versa.
3. One book can have many genres, and one genre can relate to many books.

According to that we chose to have the relation named “Has” between Book entity and Book_Genre entity and is many mandatory (Book_Genre) to many mandatory (Book) .

Homework 3
DD1368 HT21
Rafael Bechara & Paula Hanna

Relation between Book and Damaged_Book:

1. A book doesn't necessarily have damage information .
2. The relation between book and Damaged_Book has the mandotary trait on the Book_Damage side, but optional participation on the Book side because a damage must relate to a book. But a book can exist without relating to the Damaged_Book entity
3. One book can have many damage information, and a single damage information can relate to many books.

According to that we chose to have the relation named “Has” between Book entity and Damaged_Book entity and is many mandotary (Damaged_Book) to many optional (Book) . And a book can exist normally without having a damage, the Damaged_Book entity is a weak entity.

Relation between Published_Book and Prequel_And_Sequel:

1) A copy of a book may or may not have a prequel or a sequel.
2) A copy of a book can exist without having a prequel or a sequel, but a prequel or a sequel cannot exist without having a copy of a book that has a prequel or sequel. Which means that the existence of the Published_Book entity is independent of the relation, while the existence of a Prequal_And_Sequal entity is totally independent on the relation.
3) The Prequel_And_Sequel entity has no attribute or set of attributes that uniquely identify it.
4) A Book may or may not have a prequel or a sequel.

According to that, we chose to have a relation named “Has” Between Published_Book entity and Prequel_And_Squel.
OBS: The Prequel_And_Sequel entity is a weak entity (for explanation check 3)).
The Relation according to our implementation is one optional (Published_Book) to one mandatory (Prequel_And_Sequel) . (check 2,3,4 for explanation).

Relation between Student and Borrowed_Book:

1) A student can borrow a book and a Book can be borrowed by a student.
2) A student can exist without borrowing any books, but a borrowed book cannot exist without a student borrowing it. Which means that the existence of the Student entity is independent of the relation, while the existence of a BorrowedBook entity is totally independent on the relation.
3) One student can borrow many books.

Based on that, we chose to have the relation named “Borrow” between Student entity and BorrowedBook entity. It is one optional (Student) to many mandatory (BorrowedBook) type of relation. (check 2,3) for explanation)

Relation between Transaction_Info and Borrowed_Book

1) A borrowed book instance can result in a payment, and a payment can be traced to a borrowed book.
2) A borrowed book can exist without a transaction , but a transaction cannot exist without a borrowed book that results in a transaction.
3) One borrowed book can result in only one transaction (one-one)

According to that we chose to have the relation named “payment” between Borrowed_Book entity and Transaction_Info entity. It is one mandatory (Borrowed_Book) to one optional(Transaction_Info) .

Relation between Borrowed_Book and Fines:

1. A borrowed book instance can result in a fine, and a specific fine can be traced to a specific borrowed book instance.
2. A borrowed book can exist without a fine , but a fine cannot exist without a borrowed book instance that might result in a fine.
3. One borrowed book can result in only one fine (one-one)
4. The Fines entity has no attribute or set of attributes that uniquely identify it which makes it a weak entity.

According to that we chose to have the relation named “has” between Borrowed_Book entity and Transaction_Info entity. It is one mandatory (Borrowed_Book) to one optional(Fines) .