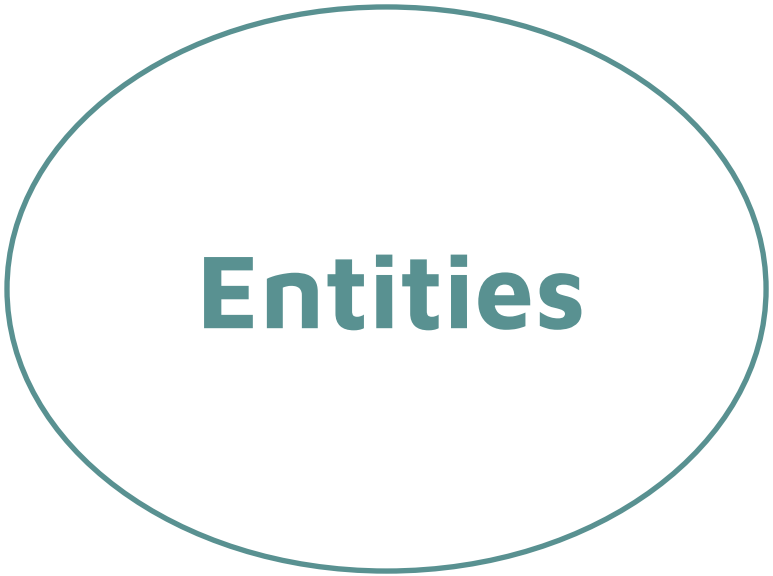
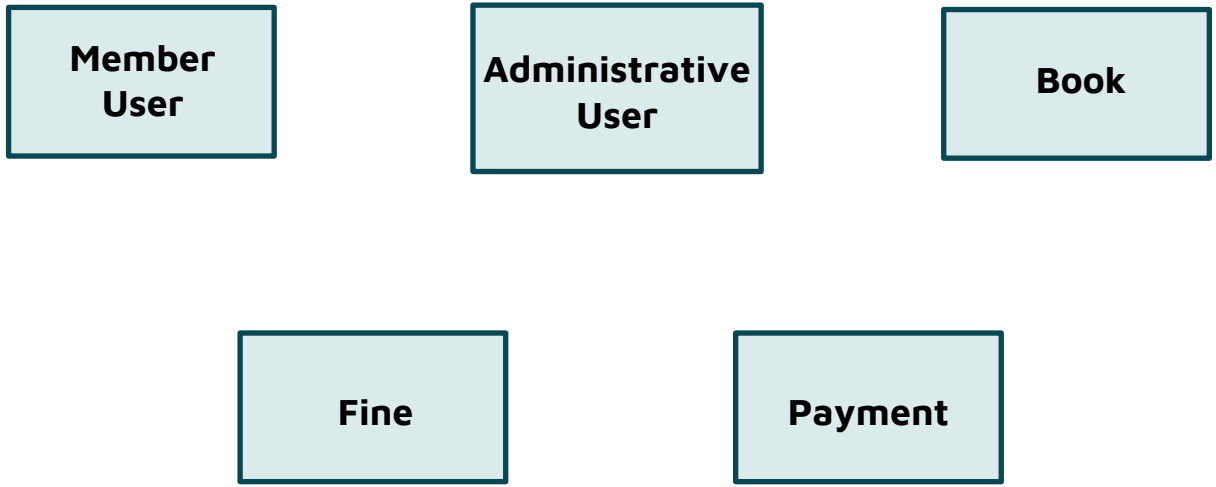
Conceptual Data Model

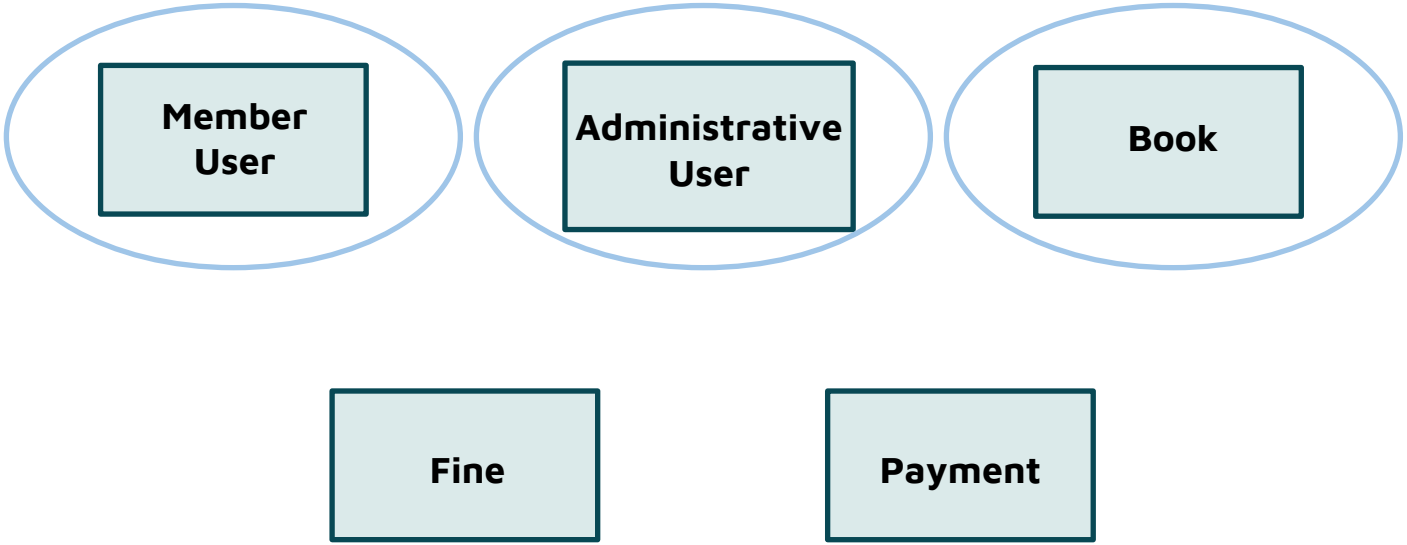

Entities

Entities

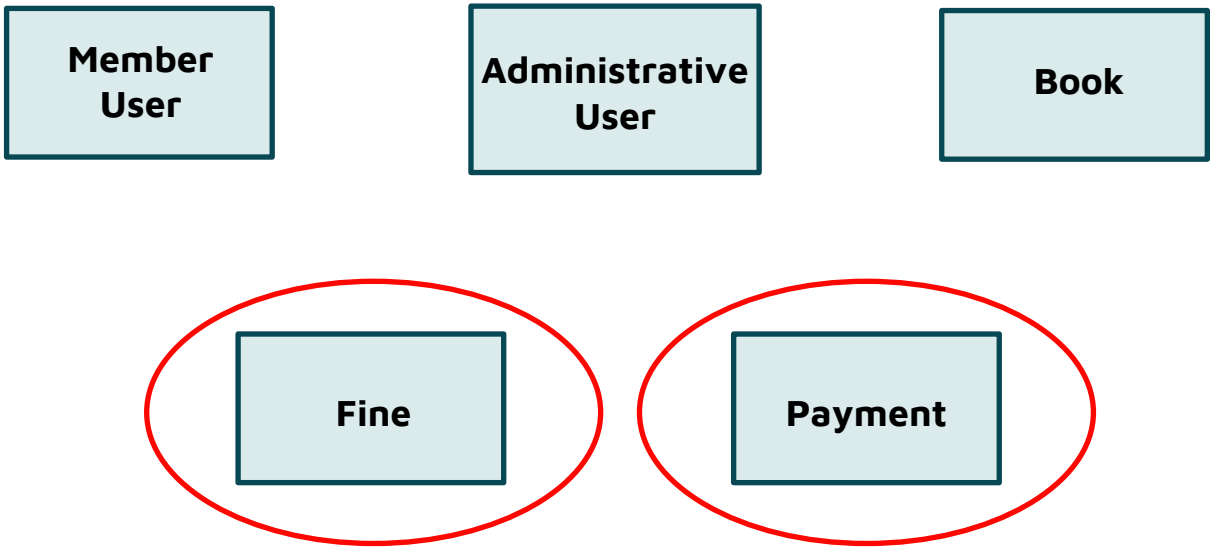


Entities

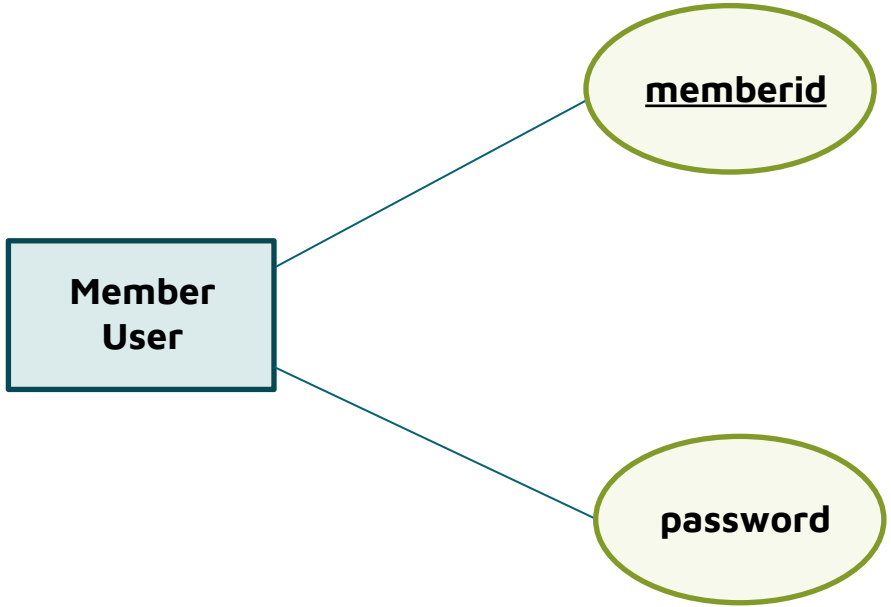
Strong Entities



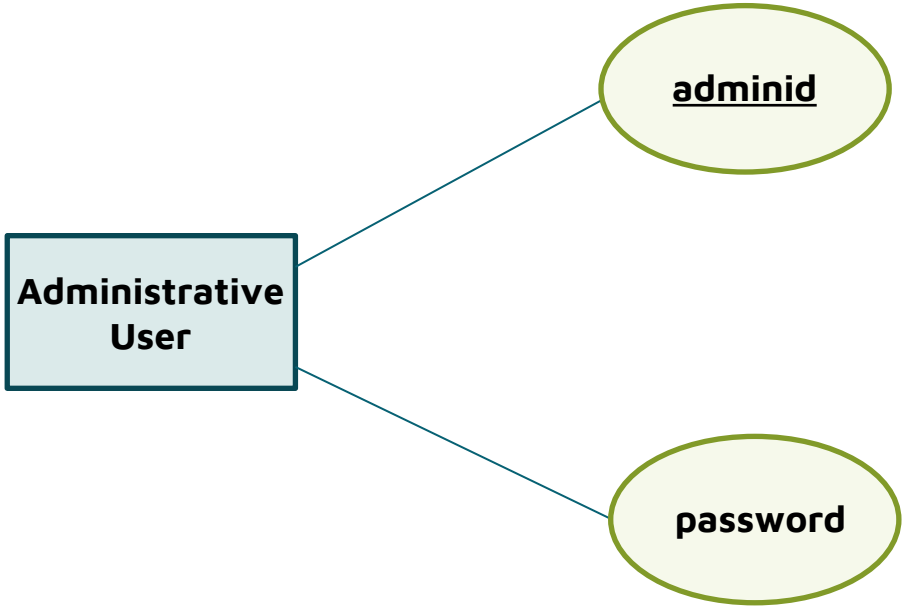
Entities


Member User
Administrative User
Book
Fine
Payment
Weak Entities

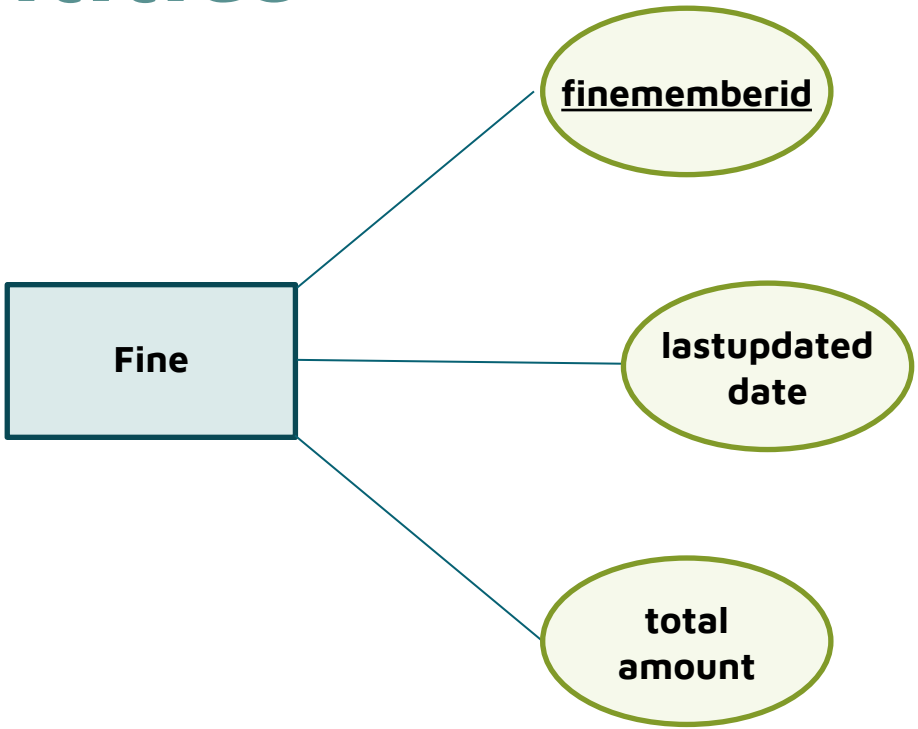
Entities



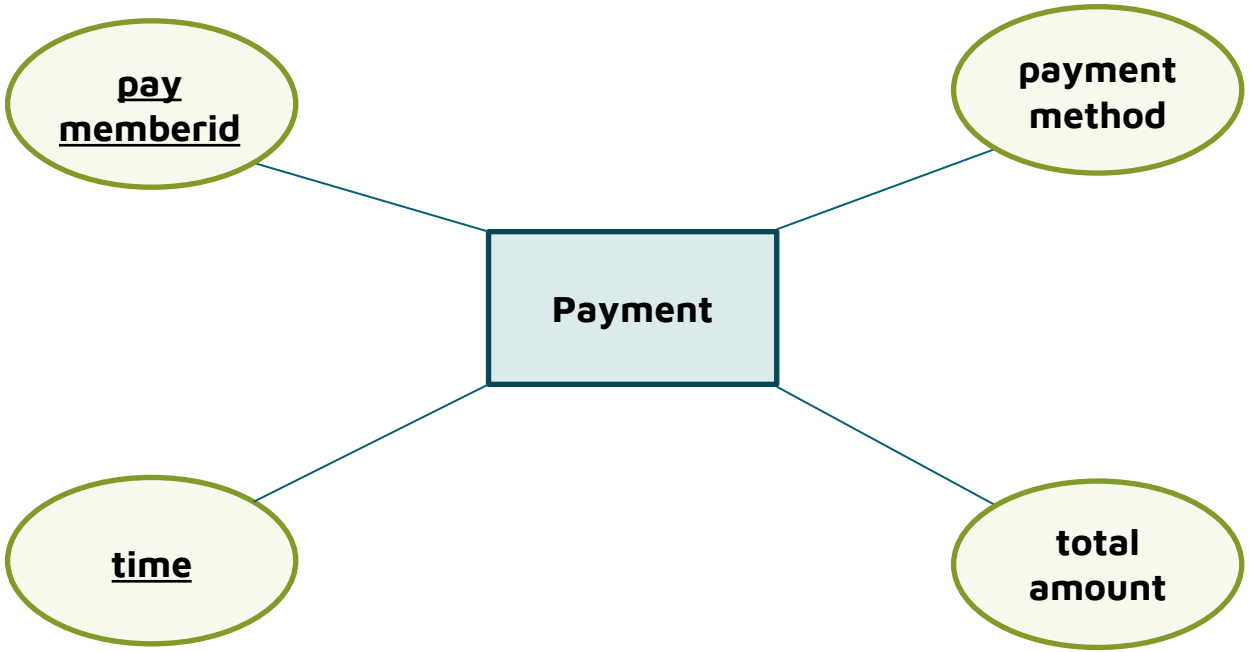
Entities



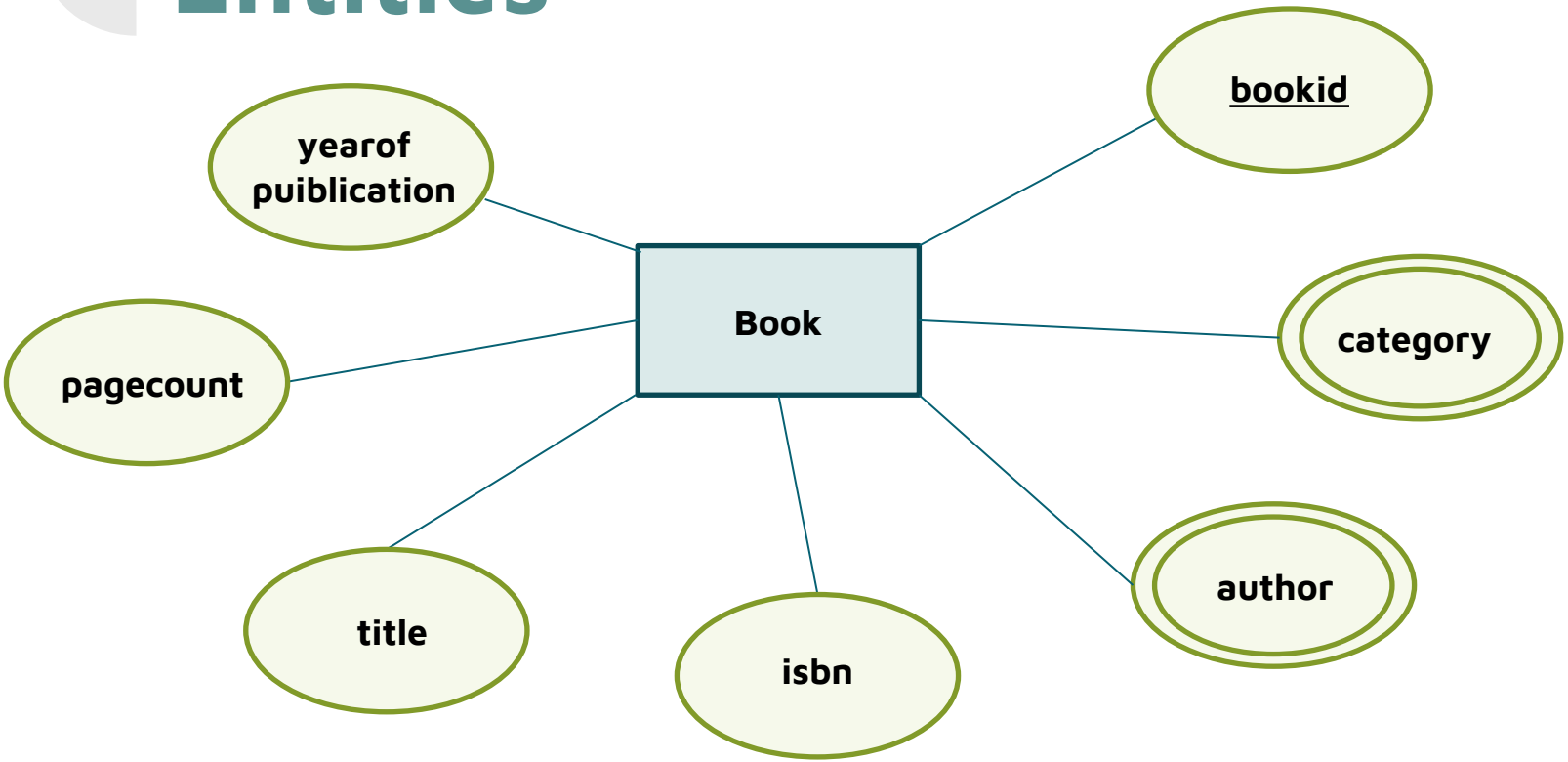
Entities



Entities

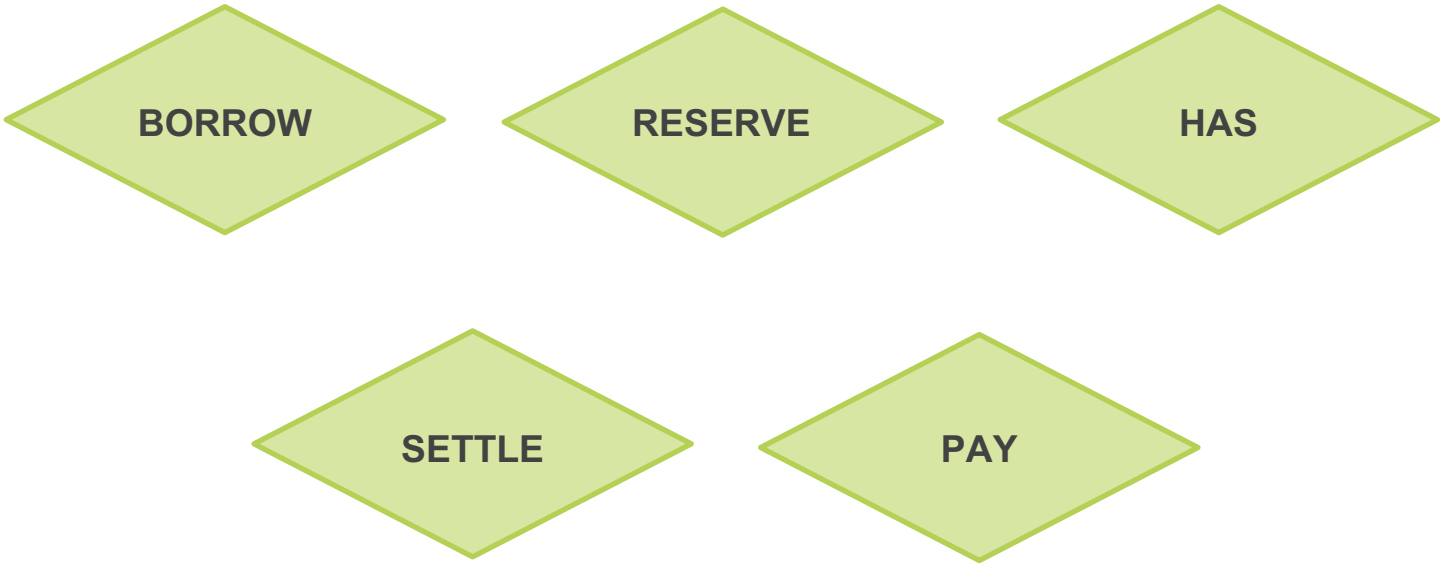


Entities

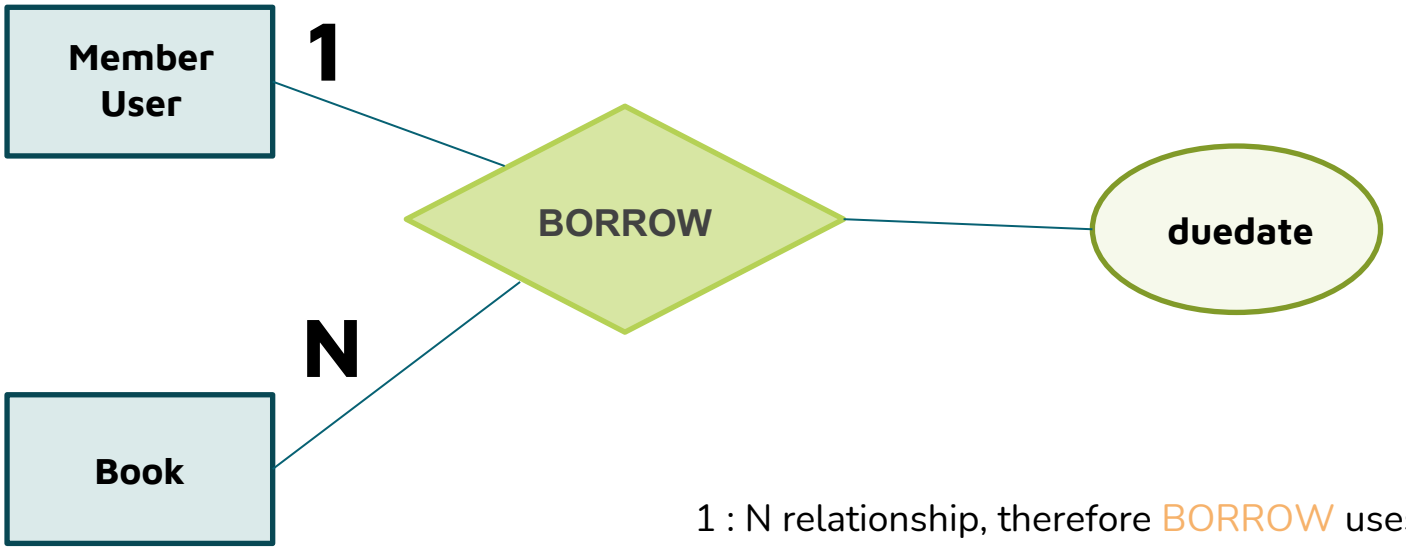

bookid
yearof puiblication
Book
category
pagecount
author
title
isbn


Relationships

Relationships

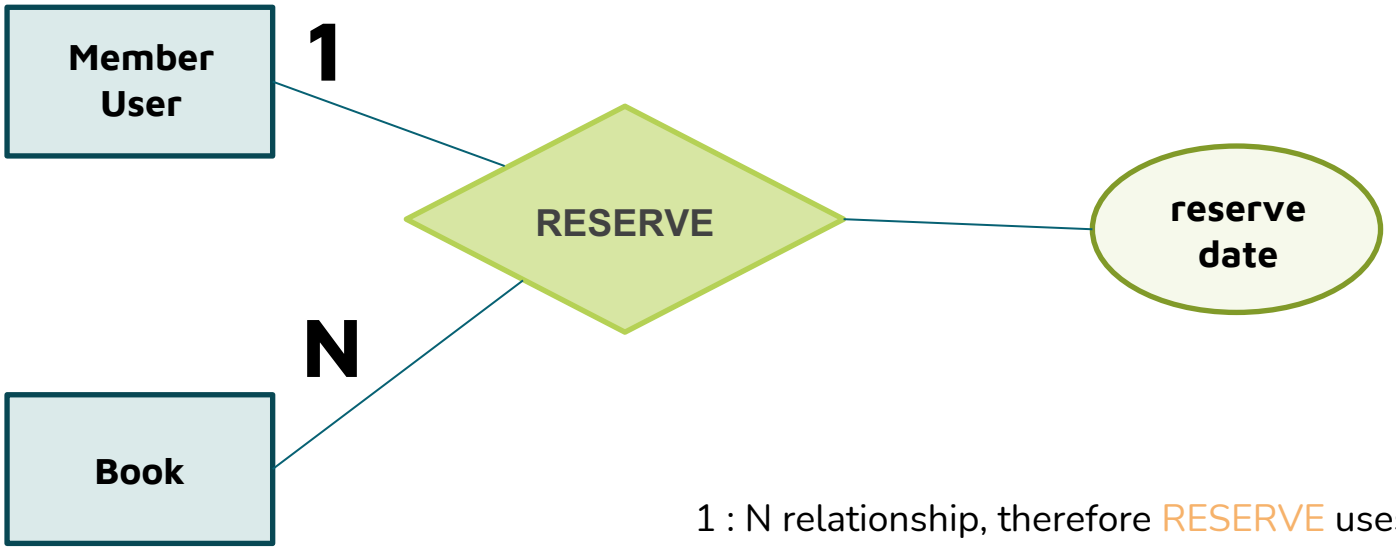


Relationships



1 : N relationship, therefore BORROW uses Book's key attribute as Primary Key

Relationships



1 : N relationship, therefore RESERVE uses Book's key attribute as Primary Key

Relationships

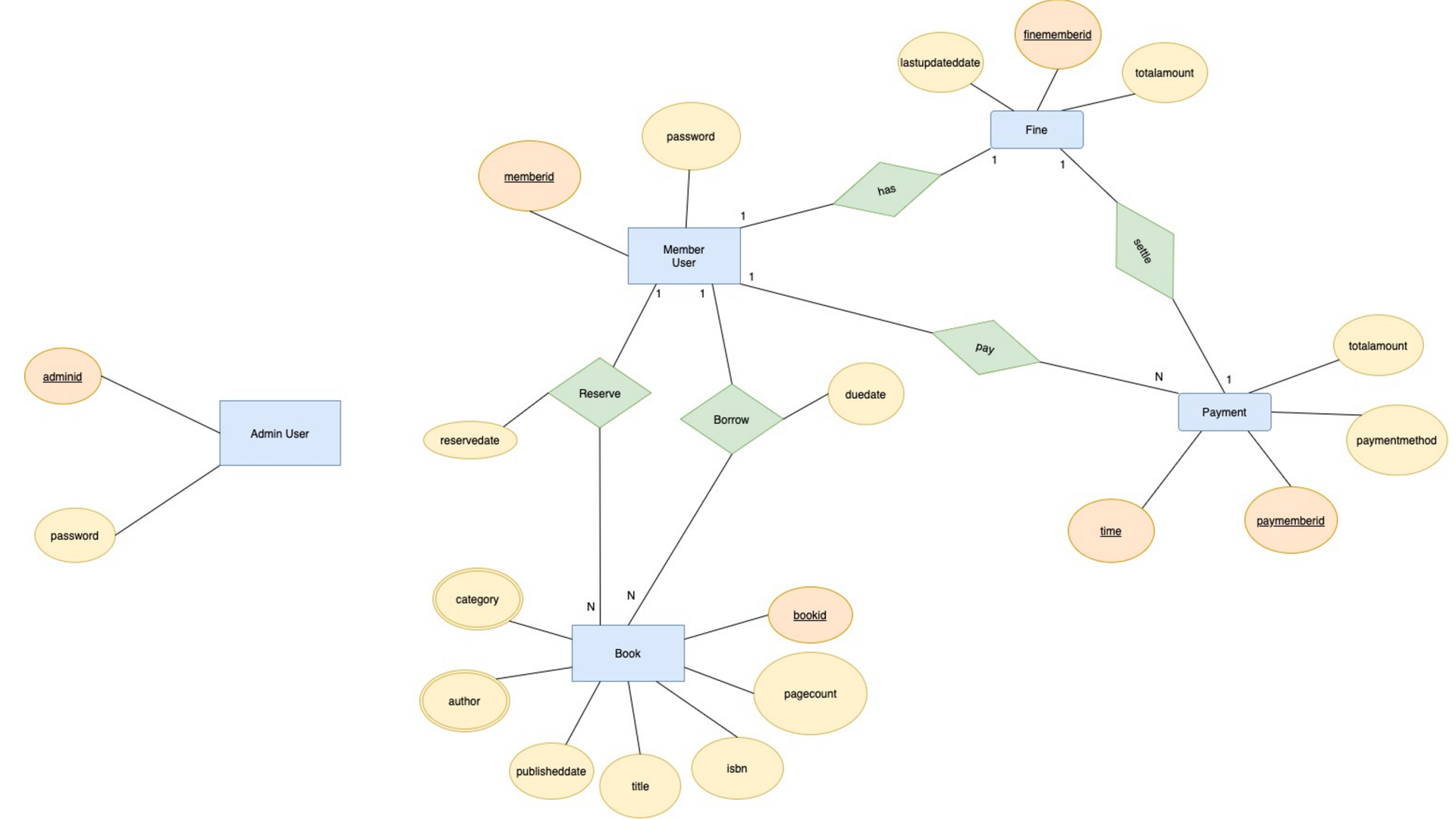
1 : 1 relationship, therefore HAS can use either Member user's or Fine's key attribute as Primary Key

Relationships

1 : N relationship, therefore PAY uses Payment's key attribute as Primary Key

Relationships

1 : 1 relationship, therefore SETTLE can use either Fine's or Payment's key attribute as Primary Key


adminid
Admin User
password
finememberid
lastupdateddate
totalamount
Fine
password
memberid
has
1
1
Member
User
1
1
1
1
settle
pay
N
1
totalamount
Payment
paymentmethod
time
paymemberid
Reserve
reservedate
Borrow
duedate
N
N
category
bookid
Book
author
pagecount
publisheddate
title
isbn

Create Relational Schema

Create relational schemas

RELATION FROM ENTITIES

Admin User	(adminid, password)
Member User	(memberid, password)
Book	(bookid, category, author, pagecount, title, isbn, yearofpublication)
Fine	(finememberid, totalamount, lastupdateddate)
Payment	(paymemberid, time, totalamount, paymethod)

Create relational schemas

RELATION FROM RELATIONSHIPS

Borrow (1 : N)	(bookid, *memberid*, duedate)
Reserve (1 : N)	(bookid, *memberid*, reservedate)
Has (1 ： 1)	(finememberid, *memberid*)
Pay (1 : N)	(paymemberid, time, *memberid*)
Settle (1 : 1)	(paymemberid, time, *finememberid*, *memberid*)

Refine Relational Schema

Multivalued Attributes : Author, Category

author and **category** are two multivalued attributes in Book
-create new entities Author and Category

Book (bookid, category, author, pagecount, title, isbn, yearofpublication)

- Book (bookid, pagecount, title, isbn, yearofpublication)
- Author (bookid, authorname)
- Category (bookid, categoryname)

Member User – Borrow - Book
Member User – Reserve - Book

Look for opportunities to combine relationships into entities:
Member User - Borrow - Book is a 1:N relationship
Member User - Reserve- Book is a 1:N relationship

-Participation of Book in Borrow and Reserve is total
-Integrate Borrow and Reserve relation into Book
-Primary Key doesn’t change, it is still Book’s PK. But now there is a Foreign Key connecting it to Member User

Member User – Borrow - Book
Member User – Reserve - Book

- Member User (memberid, password, totalamount)
- Borrow (bookid, duedate, ***memberid***)
- Reserve (bookid, reservedate, ***memberid***)
- Book (bookid, pagecount, title, isbn, yearofpublication)

- Member User (memberid, password, totalamount)
- Book (bookid, pagecount, title, isbn, yearofpublication, ***borrowedid***, borrowduedate, ***reservedid***, reservedate)

Member User -- Has -- Fine

Look for opportunities to combine relationships into entities:
Member User - Has - Fine is a 1:1 relationship

- integrate Has relation into Fine
- Primary Key doesn't change, it is still Fine's PK. But now there is a Foreign Key connecting it to Member User

- Member User (memberid, password)
- Has (finememberid, ***memberid***)
- Fine (finememberid, totalamount, lastupdateddate)

- Member User (memberid, password)
- Fine (finememberid, totalamount, lastupdateddate, ***memberid***)

Member User -- Pay -- Payment

Look for opportunities to combine relationships into entities:
Member User - Pay - Payment is a 1:N relationship

- Participation of Payment in Pay is total
- Integrate Pay relation into Payment
- Primary Key doesn't change, it is still Pay's PK. But now there is a Foreign Key connecting it to Member User

- Member User (memberid, password)
- Pay (paymemberid, time, ***memberid***)
- Payment (paymemberid, time, totalamount, paymethod)

- Member User (memberid, password, totalamount)
- Payment (paymemberid, time, totalamount, paymethod, ***memberid***)

Payment -- Settle -- Fine

Look for opportunities to combine relationships into entities:
Payment - Settle- Fine is a 1:1 relationship

- integrate Settle relation into Payment
- Primary Key doesn't change, it is still Payment's PK. But now there is a Foreign Key connecting it to Fine

- Payment (paymemberid, time, totalamount, paymethod, ***memberid***)
- Settle (paymemberid, time, ***finememberid***, ***memberid***)
- Fine (finememberid, totalamount, lastupdateddate, ***memberid***)

- Payment (paymemberid, time totalamount, paymethod, ***finememberid***, ***memberid***)
- Fine (finememberid, totalamount, lastupdateddate, ***memberid***)

Member User - Book

Look for opportunities to examine **dependency** in entities
In Book entity, borrowduedate is not dependent on bookid alone,
but dependent on (bookid and borrowedid) and is CALCULATED field
Same for Reserve

So there is **Transitive dependency** in Book
take borrowedid, borrowduedate, reservedid, reservedate out into BORROW and RESERVE

- Member User (memberid, password, totalamount)
- Book (bookid, pagecount, title, isbn, yearofpublication, ***borrowedid***, borrowduedate, ***reservedid***, reservedate)

- Member User (memberid, password, totalamount)
- Borrow (bookid, duedate, ***memberid***)
- Reserve (bookid, reservedate, ***memberid***)
- Book (bookid, pagecount, title, isbn, yearofpublication)

Our Final Schema

Admin User	(adminid, password)
Member User	(memberid, password)
Book	(bookid, pagecount, title, isbn, yearofpublication)
Author	(bookid, authorname)
Category	(bookid, categoryname)
Fine	(finememberid, totalamount, lastupdateddate, ***memberid***)
Payment	(paymemberid, time, totalamount, paymethod, ***memberid***, ***finememberid***)
Borrow	(bookid, duedate, ***memberid***)
Reserve	(bookid, reservedate, ***memberid***)

Logical Data Model

Logical Data Model

Member User (memberID, password)

Primary Key memberid

Admin User (adminid, password)

Primary Key adminid

Author (bookID, authorname)

Primary Key bookid, authorname

Book (bookid, pagecount, title, isbn, yearofpublication)

Primary Key bookid

Category (bookid, categoryname)

Primary Key bookid, categoryname

Borrow (bookid, duedate, ***memberid***)

Primary Key bookid
Foreign Key memberid **references** Member User (memberid)

Reserve (bookid, reservedate, ***memberid***)

Primary Key bookid
Foreign Key memberid **references** Member User (memberid)

Fine (finememberid, totalamount, lastupdateddate, ***memberid***)
Primary Key finememberid
Foreign Key memberid **references** Member User (memberid)

Payment (paymemberid, time, totalamount, paymethod, ***memberid, finememberid***)
Primary Key paymemberid, time
Foreign Key finememberid **references** Fine (finememberid)
Foreign Key memberid **references** Member User (memberid)