Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen · Bolzano

Library System Database

Emanuele Pippa (Student ID: 20009)

Introduction to Databases - 2024/2025
Bachelor in Computer Science

Table of contents

1. Conceptual Design

1.1 Requirements

We are interested in a system that allows the management of a Physical Municipal Library, keeping track of Books, Writers, Categories, Loans, Clients, and Publishers.

Each book has a title, a unique International Standard Book Number (ISBN-13), a year of publication, a total number of copies owned by the library, and the number of currently available copies. Every book belongs to one or more categories and can be written by one or multiple writers. Moreover, a book can be borrowed zero or multiple times, depending on the availability of its copies in the library.

A writer has a nationality, and a first and last name are required to identify them. They must have written at least one book but may have written multiple.

Each category has a name (drama, adventure, crime, etc.) and can optionally have a rank indicating its popularity score. A category can be associated with zero or multiple books.

Every publisher (publishing house) has a name (Mondadori, Giunti, Rizzoli, etc.), a foundation year, and a country of origin. A publisher can publish multiple books, while each book can be published by only one publisher. The relationship between the publisher and the book includes the edition attribute.

Each client has a name, a last name, an age, a unique client ID, a city of birth, a membership type (standard or premium for municipal employees), and a maximum number of books they can borrow at the same time. A client can have zero, one, or at most two phone numbers. A phone number can be associated with one or multiple clients. Each client can be a student, a worker, both, or neither. If they are a student, they are uniquely identified by a combination of student ID and university code. The course of study and year of enrolment must also be specified. If they are a worker, a job title and the name of the department where they work are required.

A client can apply for zero or multiple loans, provided their total number of active loans (loan Status) does not exceed their maximum (as defined by maxBookPerTime). Each loan involves a book, which can be borrowed only if at least one copy is currently available in the library, and a client who borrows the book. The loan is uniquely identified by the combination of start date, book, and client. A loan also includes an end date, an optional return date, and optional return notes (in case of delays or damages). The loan also stores its loan status (eg. active, returned, expired). Each loan must be authorized by a librarian. If a loan is created but not authorized, the book's available copies remain unchanged, and other clients can still borrow it.

Loans are either internal or external. Internal loans are for books that can only be consulted within the library, with a time limit of 15 hours. External loans have a due date (up to 30 days) and can be extended once. In case of a late return, fines are applied. A loan can only be registered if at least one copy of the book is currently available in the library.

Each librarian has a name, a last name, and a unique ID. Librarians can manage books and authorize loans.

1.2 External Constraints

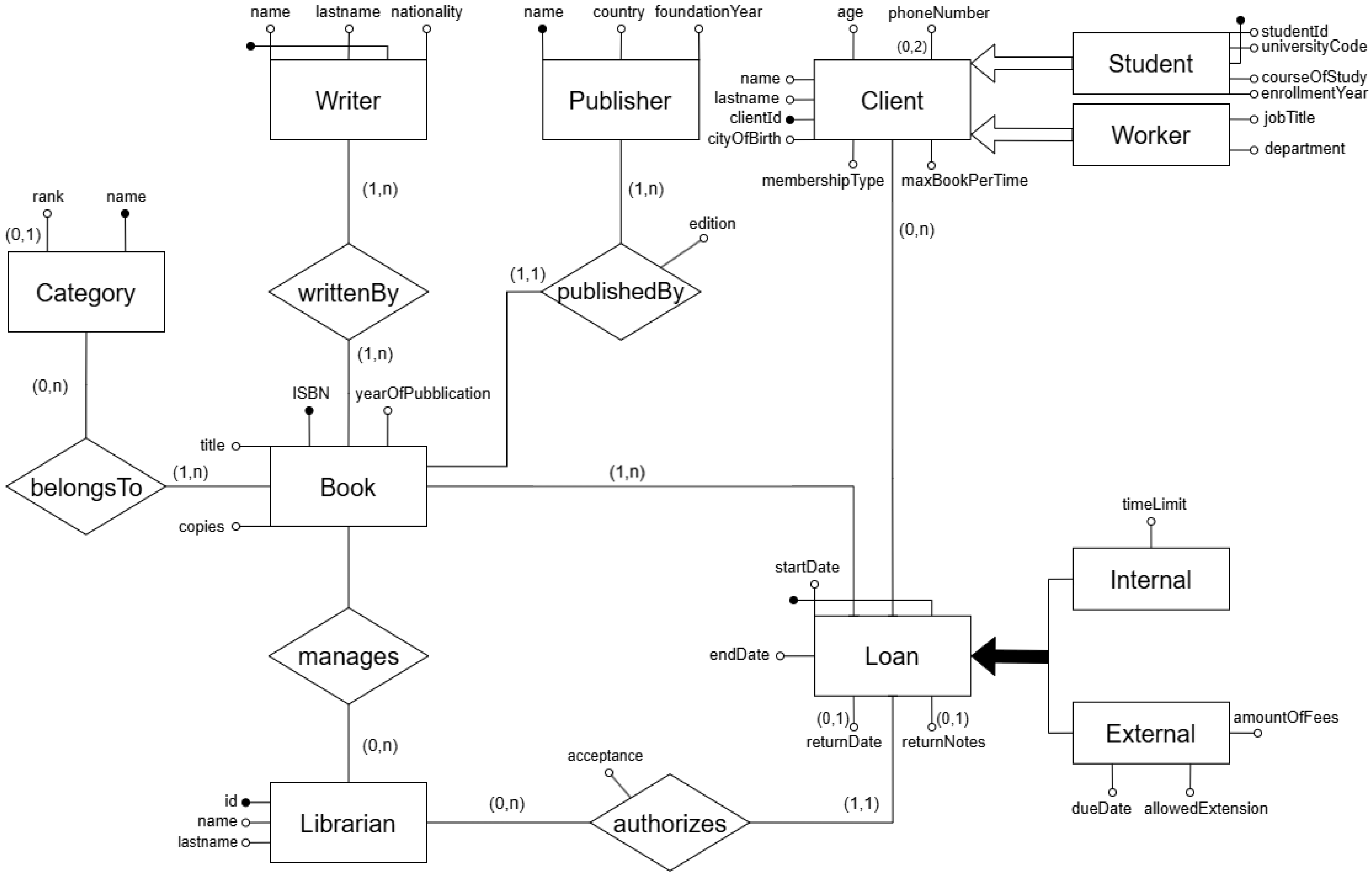
1. If a loan is created but not authorized by a librarian, the book associated with the loan should not be marked as unavailable. Copies of the book remain available to other customers until the loan is authorized.
2. A loan can only be registered if at least one copy of the book is currently available in the library.
3. A client can borrow new books only if the number of their current active loans is less than the maximum allowed by their membership.

1.3 Data Dictionary

Entity	Description	Attributes	Identifiers
Category	Category of the book's content	● rank ● name	{name}
Book	Book owned by the library	● copies ● title ● ISBN ● yearOfPubblication ● availableCopies	{ISBN}
Writer	Writer/s of a book	● name ● lastname ● nationality	{name, lastname}
Librarian	Manager of the books, authorizes the loans	● id ● name ● lastname	{id}
Publisher	Publishing house that retains the copyright e sells a book	● name ● country ● foundationYear	{name}
Client	Customer	● clientId ● name ● lastname ● cityOfBirth ● age ● membershipType ● maxBookPerTime ● phoneNumber	{clientId}
Student	Student	● studentId ● universityCode ● enrollmentYear ● courseOfStudy	{client, studentId, universityCode }
Worker	Worker	● jobTitle ● department	{client}
Loan	Loan to a client of a book's copy currently present	● startDate ● endDate ● returnDate	{startDate, book, client}

	in the library	● returnNotes ● loanStatus	
Internal	Loan of books that can only be consulted within the library	● timeLimit	{loan}
External	Loan of books that can be taken outside the library	● emountOfFees ● dueDate ● allowedExtension	{loan}

1.4 ER-Schema


name
lastname
nationality
Writer
name
country
foundationYear
Publisher
age
phoneNumber
(0,2)
name
lastname
clientId
cityOfBirth
Client
membershipType
maxBookPerTime
Student
studentId
universityCode
courseOfStudy
enrollmentYear
Worker
jobTitle
department
rank
name
(0,1)
Category
(1,n)
writtenBy
(1,n)
(1,n)
edition
(1,1)
publishedBy
(0,n)
(0,n)
belongsTo
(1,n)
ISBN
yearOfPubblication
title
Book
copies
(1,n)
timeLimit
Internal
startDate
endDate
Loan
(0,1)
returnDate
(0,1)
returnNotes
External
amountOfFees
dueDate
allowedExtension
manages
(0,n)
acceptance
id
name
lastname
Librarian
(0,n)
authorizes
(1,1)

1.5 Table of volumes

Concept	Construct	Volume	Notes
Category	Entity	20	
Book	Entity	4000	
Librarian	Entity	20	
Writer	Entity	3000	2 books x writer
Publisher	Entity	100	100 unique publishers
Client	Entity	2000	
Student	Entity	800	40% of clients
Worker	Entity	700	35% of clients
Loan	Entity	1500	
Internal	Entity	1500	Up to 100% of loans
External	Entity	1500	Up to 100% of loans
belongsTo	Relationship	6000	
writtenBy	Relationship	8000	2 writers x book
publishedBy	Relationship	4000	1 publisher per book
authorizes	Relationship	1500	Every loan is authorized
manages	Relationship	2000	100 books x librarian
Phone	Entity	2500	Some clients share numbers. up to 2 per client
HasPhone	Relationship	3000	Average 1.5 phone x client

Every loan is either internal or external, so both tables can contain up to 1500 loans, but no more than 1500 added together.

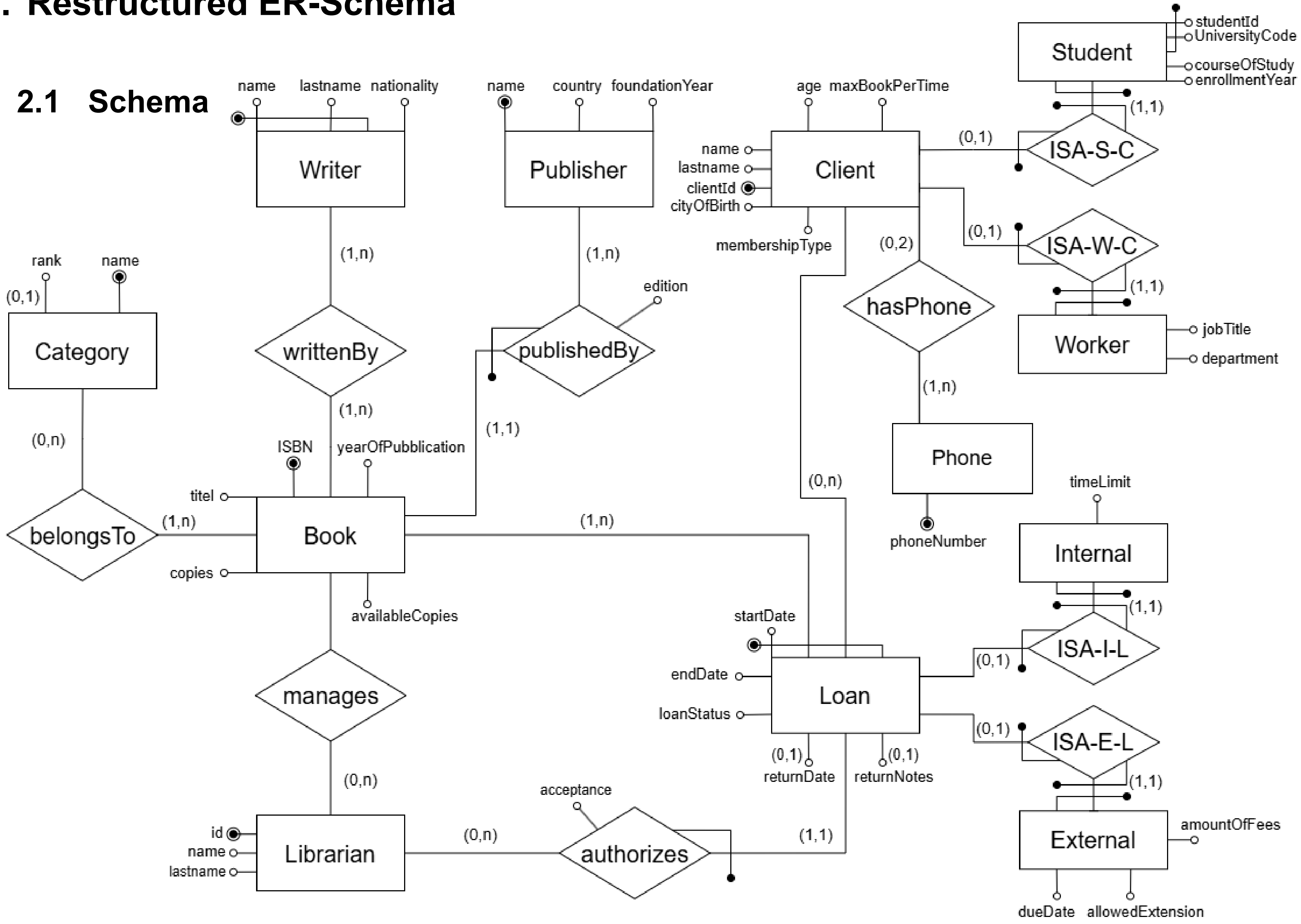
1.6 Table of Operations

Operation	Type	Frequency
Find books by a specific writer	Interactive	150 times/week
Find active loans of a client	Interactive	30 times/week
Create a new loan (internal or external)	Interactive	1200 times/month
Authorize a loan	Interactive	1000 times/month
Find available books by category	Interactive	300 times/week
Extend an external loan	Interactive	500 times/month
Register the return of a book	Interactive	1000 times/week
Check for overdue loans	Batch	700 times/week

Es. If 200 Clients borrow 6 books per month, then 200 x 6 loans = 1200 times/month

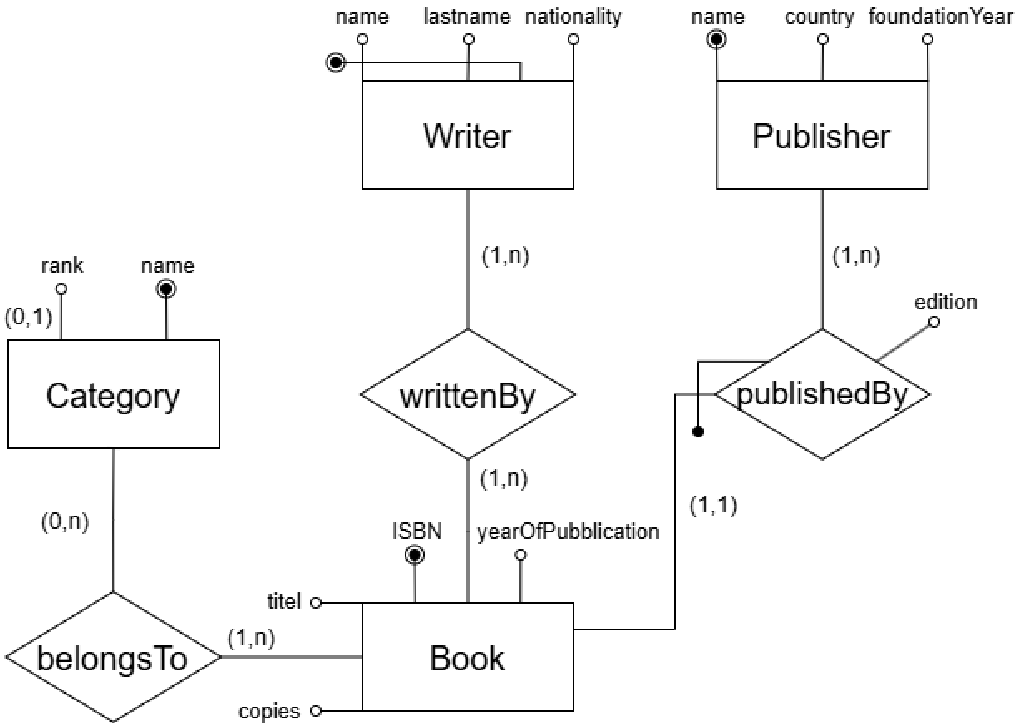
2. Restructured ER-Schema

2.1 Schema



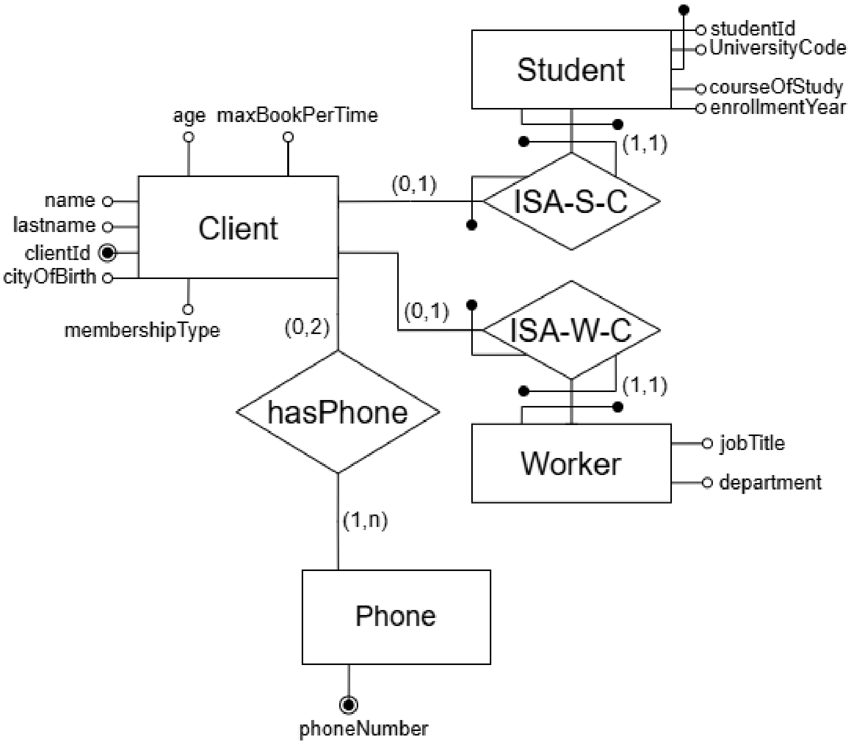
2.2 Access Tables

1. Visualize the books available in the library with title, author, publisher and category



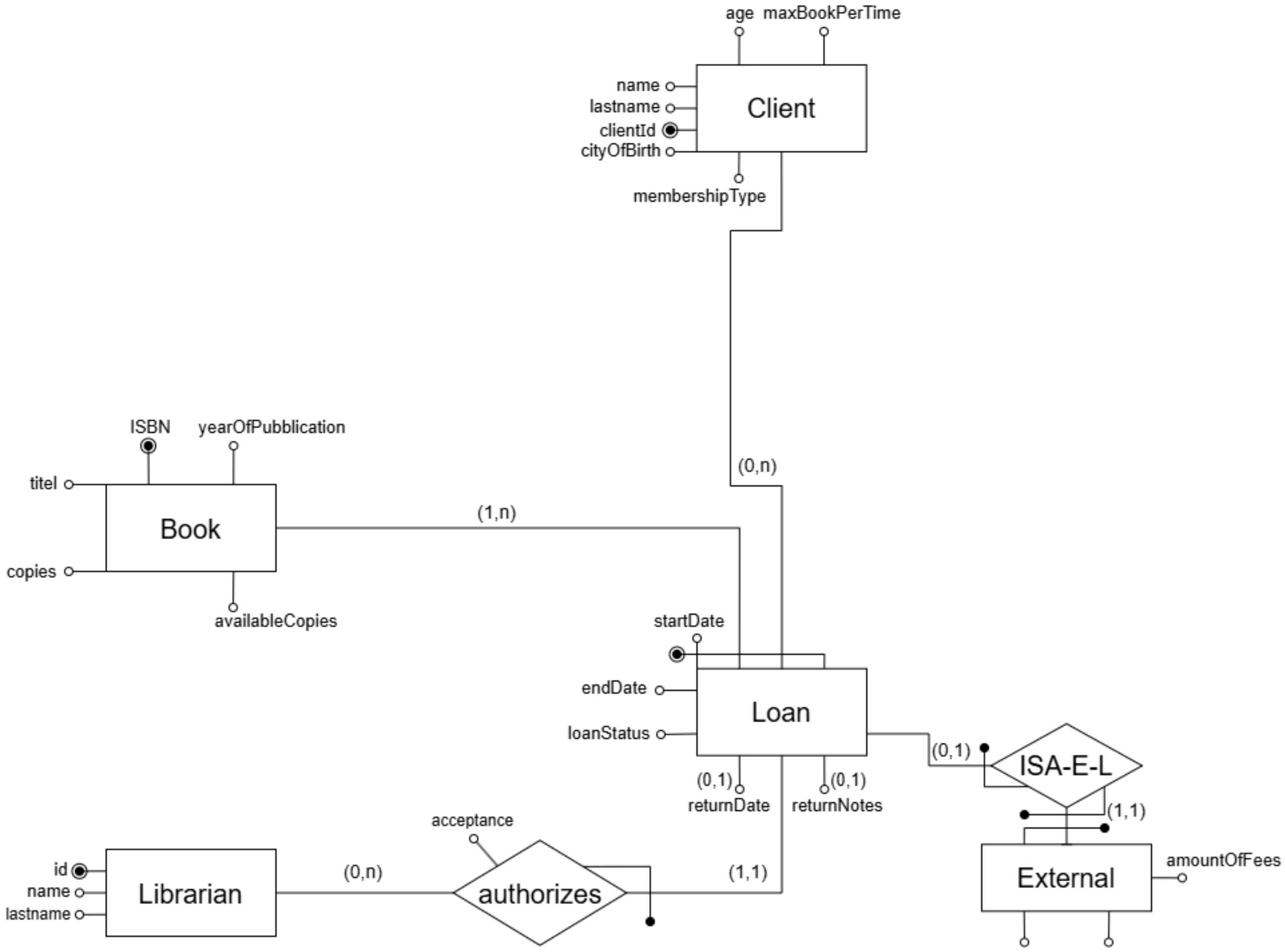
Concept	Construct	Access	Type
Book	Entity	4000	R
writtenBy	Relationship	4000	R
Writer	Entity	4000	R
Category	Entity	4000	R
belongsTo	Relationship	4000	R
Publisher	Entity	4000	R
publishedBy	Relationship	4000	R

2. Insert a new customer into the system with their personal information and phone numbers.



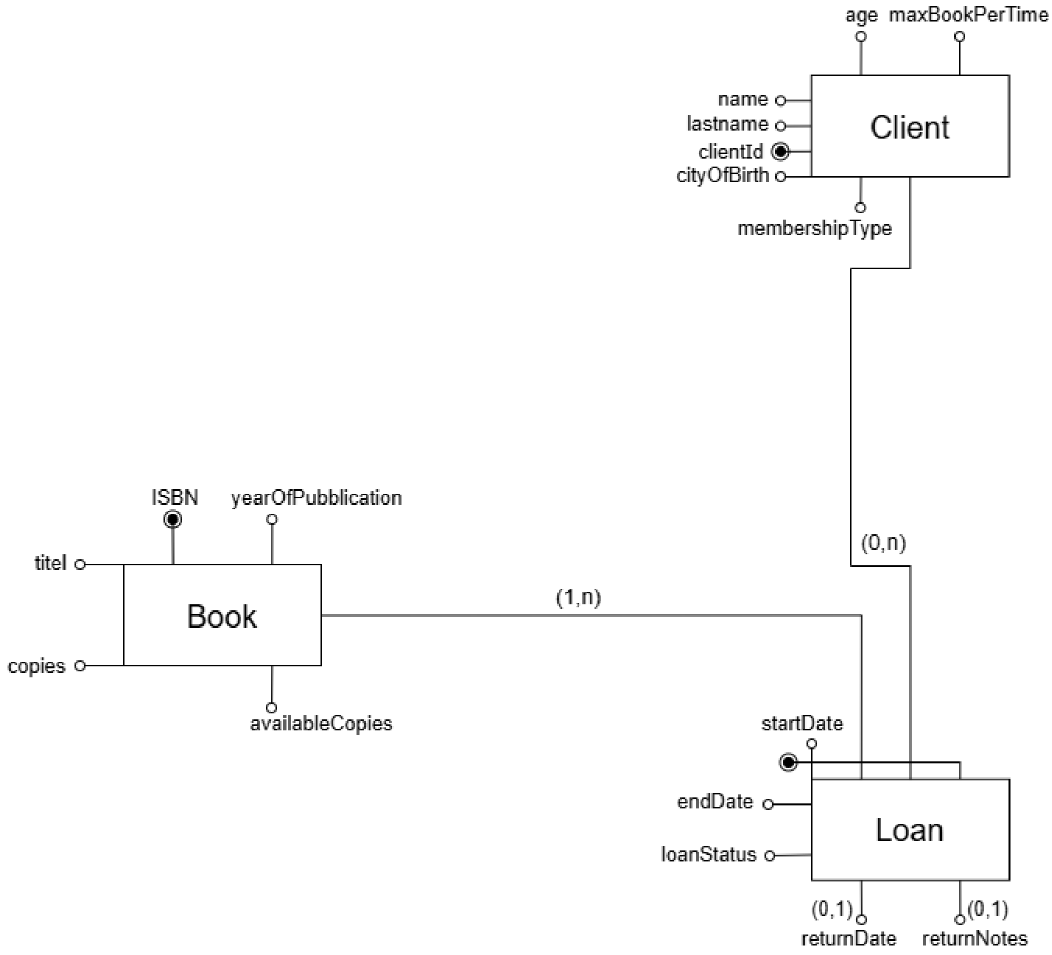
Concept	Construct	Access	Type
Client	Entity	1	W
Phone	Entity	2	W
HasPhone	Relationship	2	W
Student	ISA subclass	1	W
Worker	ISA subclass	1	W

3. Record a new external loan authorized by a librarian.



Concept	Construct	Access	Type
Loan	Entity	1	W
External	ISA subclass	1	W
Client	Entity	1	R
Book	Entity	2	R/W
Librarian	Entity	1	R
Authorizes	Relationship	1	W

4. Record the return of a book: update the number of copies available and enter any notes on the return.

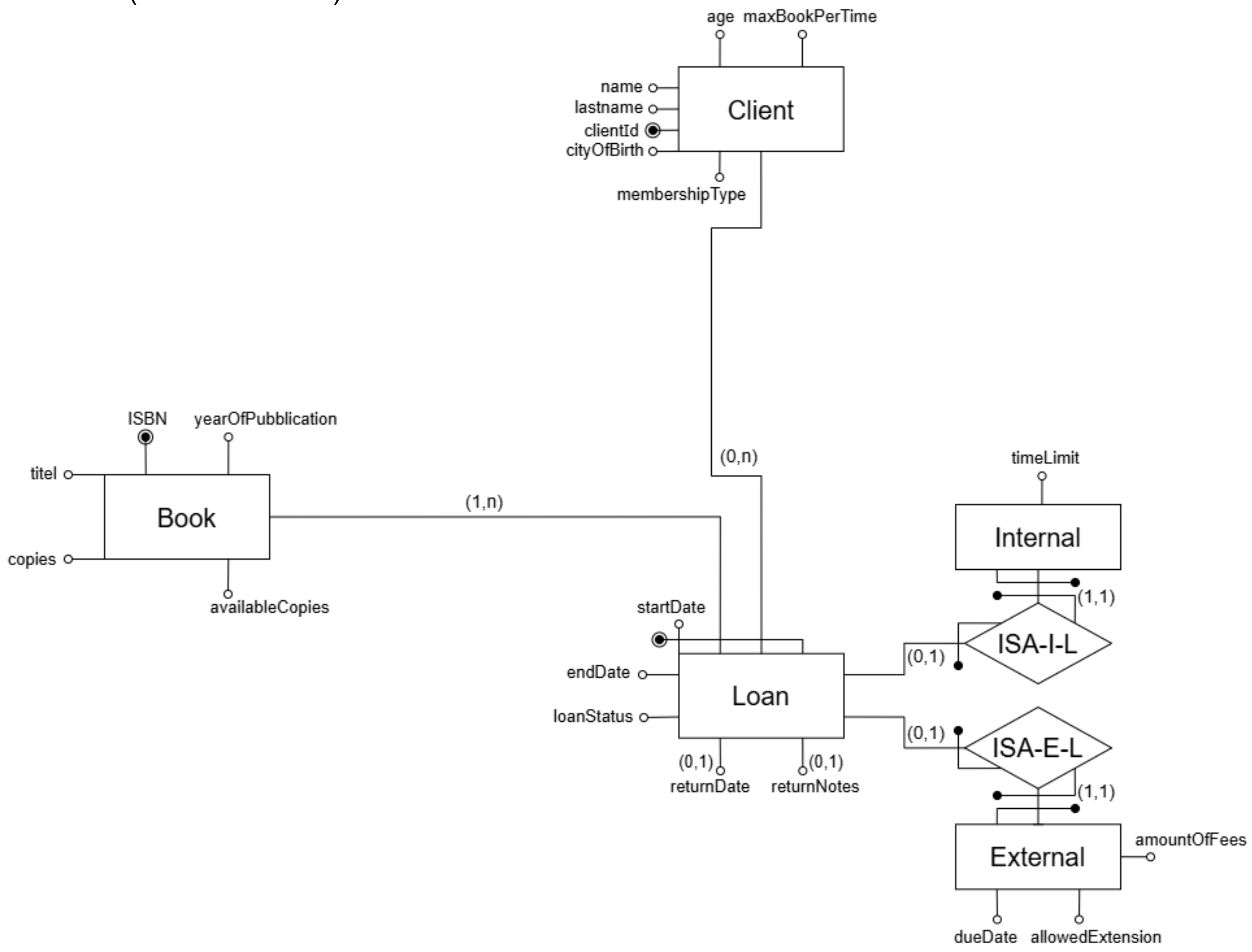


Concept	Construct	Access	Type
Loan	Entity	1	W
Book	Entity	1	W
Client	Entity	1	R

5. Get a list of all books currently on loan with the customer who borrowed them.

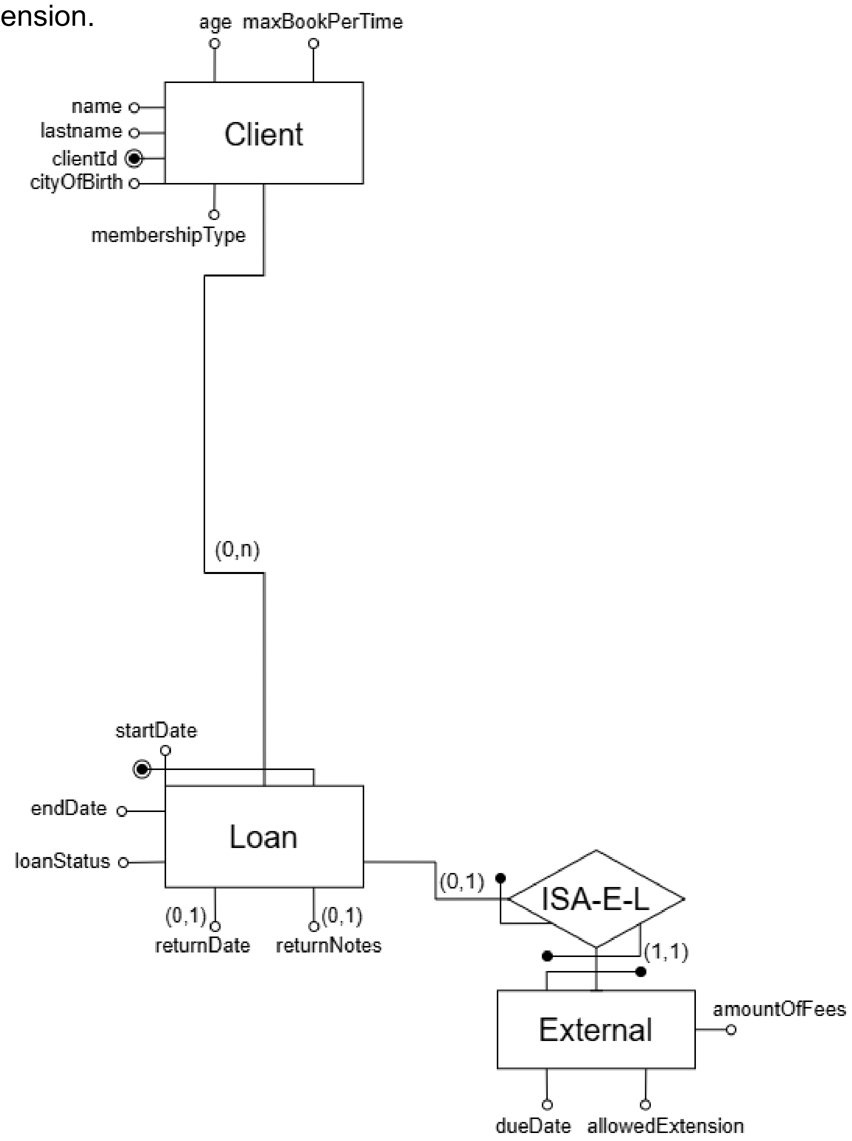
Concept	Construct	Access	Type
Loan	Entity	5	R
Client	Entity	1	R
Book	Entity	5	R

6. Create a new loan and assign it as internal (library consultation only) or external (take home loan).



Concept	Construct	Access	Type
Loan	Entity	1	W
Internal	ISA subclass	1	W
External	ISA subclass	1	W
Client	Entity	1	R
Book	Entity	2	R/W

7. Update the expiration date of an external loan, if the customer is still entitled to the extension.

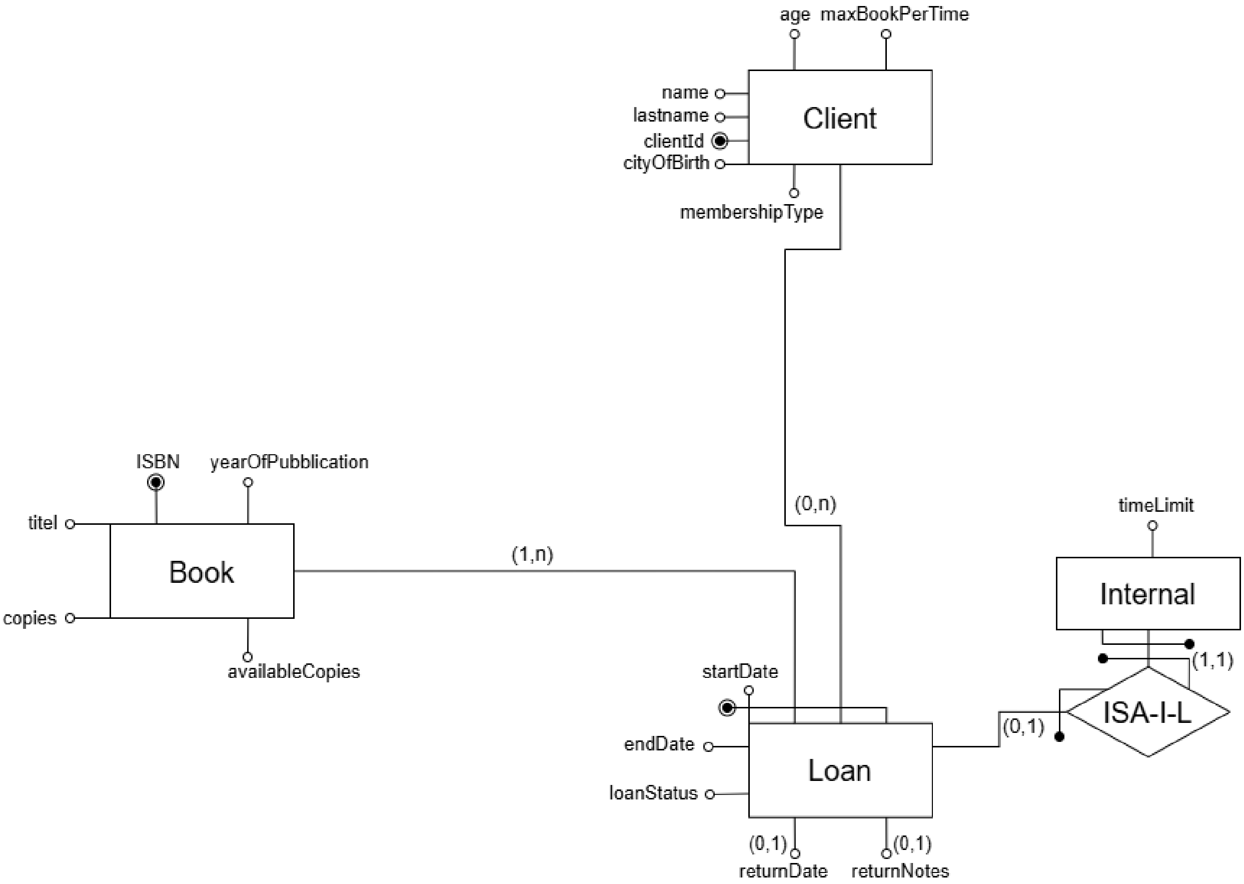


Concept	Construct	Access	Type
Loan	Entity	2	R/W
External	ISA subclass	1	W
Client	Entity	1	R

8. Add a fine to a customer who returned a book after its expiration date

Concept	Construct	Access	Type
Loan	Entity	1	R
External	ISA subclass	1	W
Client	Entity	1	R

9. Record the end of an internal loan and free the book for others



Concept	Construct	Access	Type
Loan	Entity	1	W
Internal	ISA subclass	1	W
Client	Entity	1	R
Book	Entity	1	W

3. Direct translation to the relational model

3.1 Relational schema

Book(ISBN, title, yearOfPublication, copies, availableCopies)
Primary key: ISBN

Category(name, rank*)
Primary key: name

BelongsTo(book, category)
Primary key: book, category
Foreign Key: BelongsTo[book] ⊆ Book[ISBN]
Foreign Key: BelongsTo[category] ⊆ Category[name]

Writer(name, lastname, nationality)
Primary key: name, lastname

WrittenBy(book, writerName, writerLastname)
Primary key: book, writerName, writerLastname
Foreign key: WrittenBy[book] ⊆ Book[ISBN]
Foreign key: WrittenBy[writerName, writerLastname] ⊆ Writer[name, lastname]

Publisher(name, country, foundationYear)
Primary key: name

PublishedBy(book, publisher, edition)
Primary key: book, edition
Foreign key: PublishedBy[book] ⊆ Book[ISBN]
Foreign key: PublishedBy[publisher] ⊆ Publisher[name]

Client(clientId, name, lastname, age, cityOfBirth, membershipType, maxBookPerTime)
Primary key: clientId

Phone(phoneNumber)
Primary key: phoneNumber

HasPhone(client, phone)
Primary key: client, phone
Foreign key: HasPhone[client] ⊆ Client[clientId]
Foreign key: HasPhone[phone] ⊆ Phone[phoneNumber]

Student(client, studentId, universityCode, courseOfStudy, enrollmentYear)

Primary key: client, studentId, universityCode

Foreign key: Student[client] ⊆ Client[clientId]

ISA: Student ⊆ Client

Worker(client, jobTitle, department)

Primary key: client

Foreign key: Worker[client] ⊆ Client[clientId]

ISA: Worker ⊆ Client

Loan(startDate, book, client, endDate, returnDate*, returnNotes*, loanStatus)

Primary key: startDate, book, client

Foreign key: Loan[book] ⊆ Book[ISBN]

Foreign key: Loan[client] ⊆ Client[clientId]

loanStatus can assume values like: pending, active, returned, expired.

Internal(startDate, book, client, timeLimit)

Primary key: startDate, book, client

Foreign key: Internal[startDate, book, client] ⊆ Loan[startDate, book, client]

ISA: Internal ⊆ Loan

External(startDate, book, client, dueDate, allowedExtension, amountOfFees)

Primary key: startDate, book, client

Foreign key: External [startDate, book, client] ⊆ Loan[startDate, book, client]

ISA: External ⊆ Loan

Librarian(id, name, lastname)

Primary key: id

Manages(librarian, book)

Primary key: librarian, book

Foreign key: Manages[librarian] ⊆ Librarian[id]

Foreign key: Manages[book] ⊆ Book[ISBN]

Authorizes(librarian, startDate, book, client, acceptance)

Primary key: startDate, book, client

Foreign key: Authorizes[startDate, book, client] ⊆ Loan[startDate, book, client]

Foreign key: Authorizes[librarian] ⊆ Librarian[id]

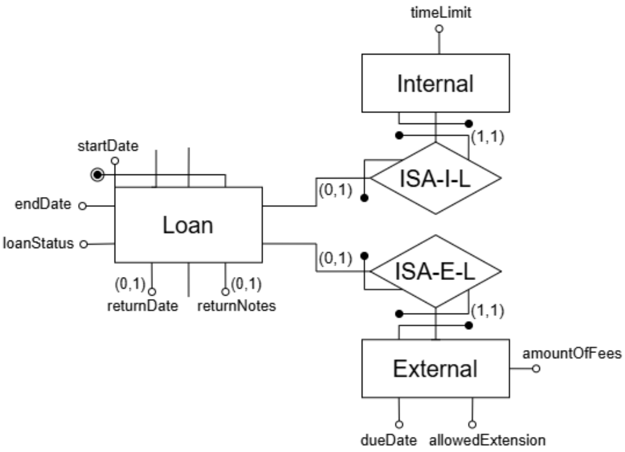
4. Restructuring of the relational schema

4.1 Restructured relational -> restructuring steps

To produce a cleaner and more efficient relational schema, the following restructuring steps have been applied:

- **When accessing loan data, it is our interest to know whether it is internal or external.**

To represent internal and external loans, we apply an ISA structure.
We keep Loan as the parent entity, and define Internal and External as subclasses, each with their specific attributes.



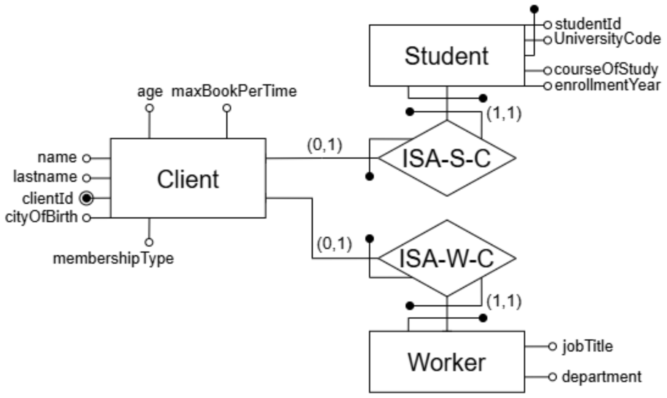
Loan(startDate, book, client, endDate, returnDate*, returnNotes*, loanStatus)
Primary key: startDate, book, client
Foreign key: Loan[book] ⊆ Book[ISBN]
Foreign key: Loan[client] ⊆ Client[clientId]

Internal(startDate, book, client, timeLimit)
Primary key: startDate, book, client
Foreign key: Internal[startDate, book, client] ⊆ Loan[startDate, book, client]

External(startDate, book, client, dueDate, allowedExtension, amountOfFees)
Primary key: startDate, book, client
Foreign key: External[startDate, book, client] ⊆ Loan[startDate, book, client]

- **When accessing data about Clients, it's our interest to know if they are students or workers.**

We use ISA relationships to represent the specializations Student and Worker, which both inherit from the general entity Client.



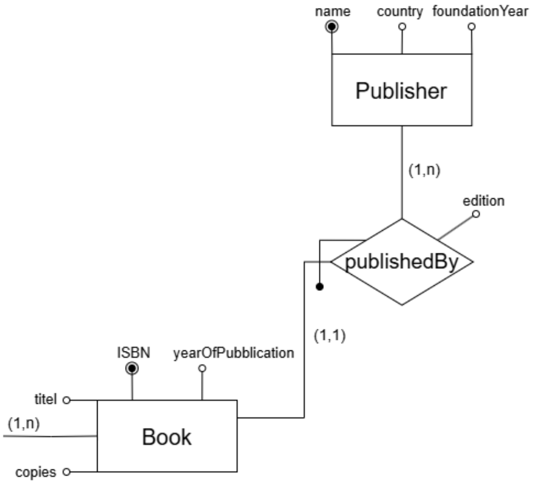
Client(clientId, name, lastname, age, cityOfBirth, membershipType, maxBookPerTime)

Student(client, studentId, universityCode, courseOfStudy, enrollmentYear)
Primary key: client, studentId, universityCode
Foreign key: Student[client] ⊆ Client[clientId]

Worker(client, jobTitle, department)
Primary key: client
Foreign key: Worker[client] ⊆ Client[clientId]

- **When accessing data about books (Book), we immediately want to know the details of the publisher.**

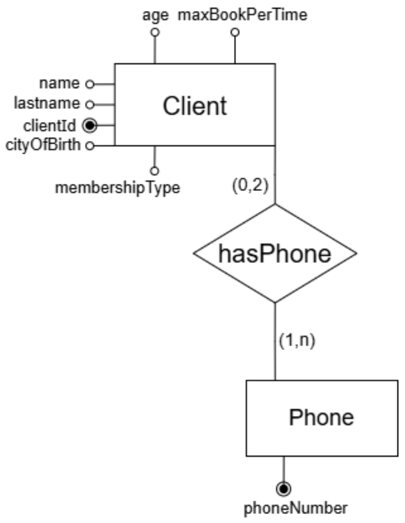
The relationship between a book and its publisher is represented by the PublishedBy relation.
We avoid merging publisher attributes directly into Book to prevent redundancy, especially when a publisher publishes multiple books.



Book(ISBN, title, yearOfPublication, copies, availableCopies)
Primary key: ISBN

PublishedBy(book, publisher, edition)
Primary key: book
Foreign key: PublishedBy[book] ⊆ Book[ISBN]
Foreign key: PublishedBy[publisher] ⊆ Publisher[name]

- **When accessing data on phones, we want a simplified phone management**



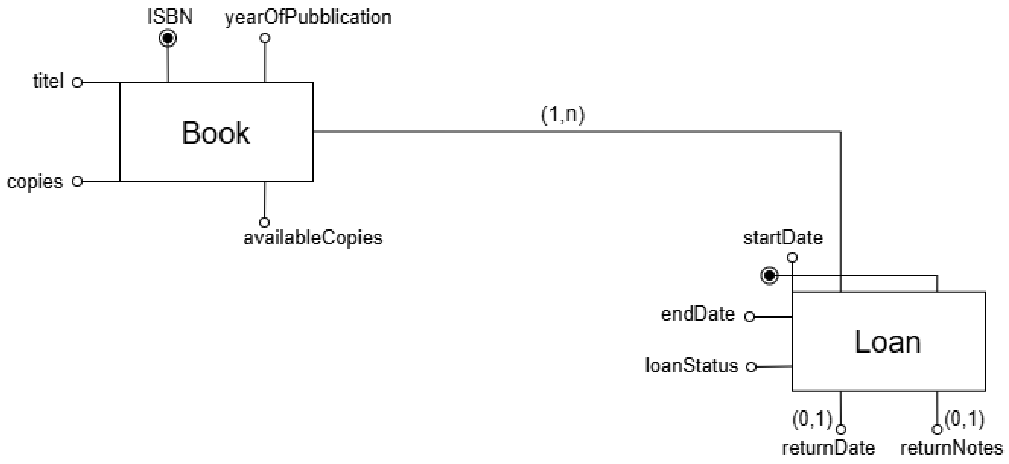
We keep a separate relation HasPhone(client, phone) to correctly handle the (0,2) cardinality between Client and Phone.
This allows the same phone number to be reused and avoids unnecessary NULL values in the Client table.

Client(clientId, name, lastname, age, cityOfBirth, membershipType, maxBookPerTime)

Phone(phoneNumber)
  Primary key: phoneNumber

HasPhone(client, phone)
  Primary key: client, phone
  Foreign key: HasPhone[client] ⊆ Client[clientId]
  Foreign key: HasPhone[phone] ⊆ Phone[phoneNumber]

- **When accessing data about Loans, it is always our interest to know the availability of book's copies in the library without additional joins**



All attributes related to the loan (such as loanStatus, and the availability of copies) are handled directly within the Loan or Book entities.

4.2 External constraints

External Constraint 1:

A loan becomes active only if it is authorized by a librarian. Until that moment, Book.availableCopies remains unchanged.

- *IF NOT EXISTS (SELECT * FROM Authorizes WHERE Authorizes.startDate = NEW.startDate AND Authorizes.book = NEW.book AND Authorizes.client = NEW.client) THEN Book.availableCopies REMAINS UNCHANGED;*

External Constraint 2:

Before inserting a new loan, a check must verify that Book.availableCopies > 0.
Once the loan is authorized, availableCopies is decremented by 1.

- *CHECK (EXISTS (SELECT * FROM Book WHERE Book.ISBN = NEW.book AND Book.availableCopies > 0))*

External Constraint 3:

A client can borrow a new book only if their number of active loans is below their allowed maximum (Client.maxBookPerTime).

- *CHECK ((SELECT COUNT(*) FROM Loan WHERE Loan.client = NEW.client AND Loan.loanStatus = 'active') < (SELECT maxBookPerTime FROM Client WHERE Client.clientId = NEW.client))*