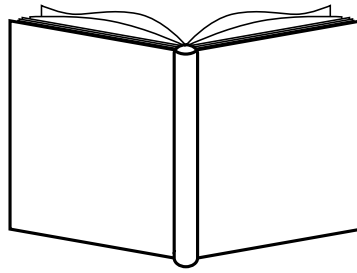


Biblio



Documentation

issues 3 open forks 0 stars 2 license GPL-3.0

DB Postgres Container Docker

WebFramework Flask Framework Bootstrap

Language Python Language HTML Language CSS

Biblio is a tool for your private library. It manages your books and magazines.

The code can be found here: <https://github.com/BennerLukas/biblio>. It is recommended to do a `git clone` and look at the code like this. But its via the zip-folder or github.com possible as well.

Composed Docker Containers

Otherwise, you can use the docker-image provided in the repository.

The first time you use this docker-compose you must initialise the containers with:

```
cd biblio
docker compose up
```

The Frontend is now visible under `localhost:5000`

Afterwards, the container may be started with

```
docker compose start
```

To shut down the container you can either use a different terminal and use

```
docker compose stop
```

or by pressing Ctrl + C in the terminal used to initialise the container.

About the project

Team

- [Lukas Benner](#) (6550912)
- [Phillip Lange](#) (5920414)
- [Alina Buss](#) (4163246)

Target

With Biblio we want to build a system to keep track of your private book collection like its a real library.

Biblio is your tool to manage your own private book collection. Whether you want to keep track of all your books. Find gems you forgot about or keep track of your reading list. With Biblio you can also invite friends and family to your private library. It allows you to share and borrow books between each other.

We think its important to know your possessions and keep it managed. We can help you to focus on reading rather than searching. On the other hand its very important to share, so everybody can enjoy reading and learning new stuff. Biblio helps you to organize this process easily.

Tools

For developing Biblio we used Python, Flask, Postgres and a little bit of bootstrap for easier styling.

Specification

'Biblio' will be a webapp for organizing your private book collection.

Group-member: Alina Buss (4163246), Phillip Lange (5920414), Lukas Benner (6550912)

The used database schema und functionalities are described in the following:

- Books: Every book is stored in the database with following attributes: title, ISBN, recommended Age, language, publishing date, genre, edition, wether its available and a unique Identifier.
- Author & Publisher: Every Book has a authoter and Publishier assigned. These contain information like address, name and a unique identifier.
- Location: Every book has a location assigned, where its supposed to be. The Location has the following attributes: address, floor, room, shelf, compartment and a unique identifier.
- User: Every user of the database has a name, a user_id and his/her date of birth is stored.
- Borrow: A User can lend severall books. For every loan the timestamp will be stored. For every borrowed item the duration and the Book_ID will be used to assign it to a loan.
- The database can tracks which books a particular user already read.

- The Address is handled in a separate table and has the following attributes: city, country, street, house number, zipcode and a unique identifier.

Presentation & Model

Biblio

Sign Out

Home

Books

Add new book

Profile

Return book

Reading history

LOANS

Active loans

Loan History

EXTRA

About

Loans

Search Book

List of all your active loans

User's first name	User's last name	loan_id	loan_timestamp	loan_duration	loan_active	book_isbn	book_title	book_edition
Ben	Hell	4	2021-03-27 18:03:33.050095	14	True	9780575097568	Rivers of London	1

Biblio

Sign Out

Home

Books

Add new book

Profile

Return book

Reading history

LOANS

Active loans

Loan History

EXTRA

About

Home

benni

What is Biblio?

Biblio is your tool to manage your own private book collection. Whether you want to keep track of all your books. Find gems you forgot about or keep track of your reading list. With Biblio you can also invite friends and family to your private library. It allows you to share and borrow books between each other.

Why Biblio?


We think its important to know your possessions and keep it managed. We can help you to focus on reading rather than searching. On the other hand its very important to share, so everybody can enjoy reading and learning new stuff. Biblio helps you to organize this process easily.


More


You want to know more? Check out the [About](#) page


3 / 7


Biblio


 Home

 Books


 Add new book


 Profile


 Return book

 Reading history


LOANS




 Active loans

 Loan History

EXTRA



 About

Return Book

Return via isbn:

isbn

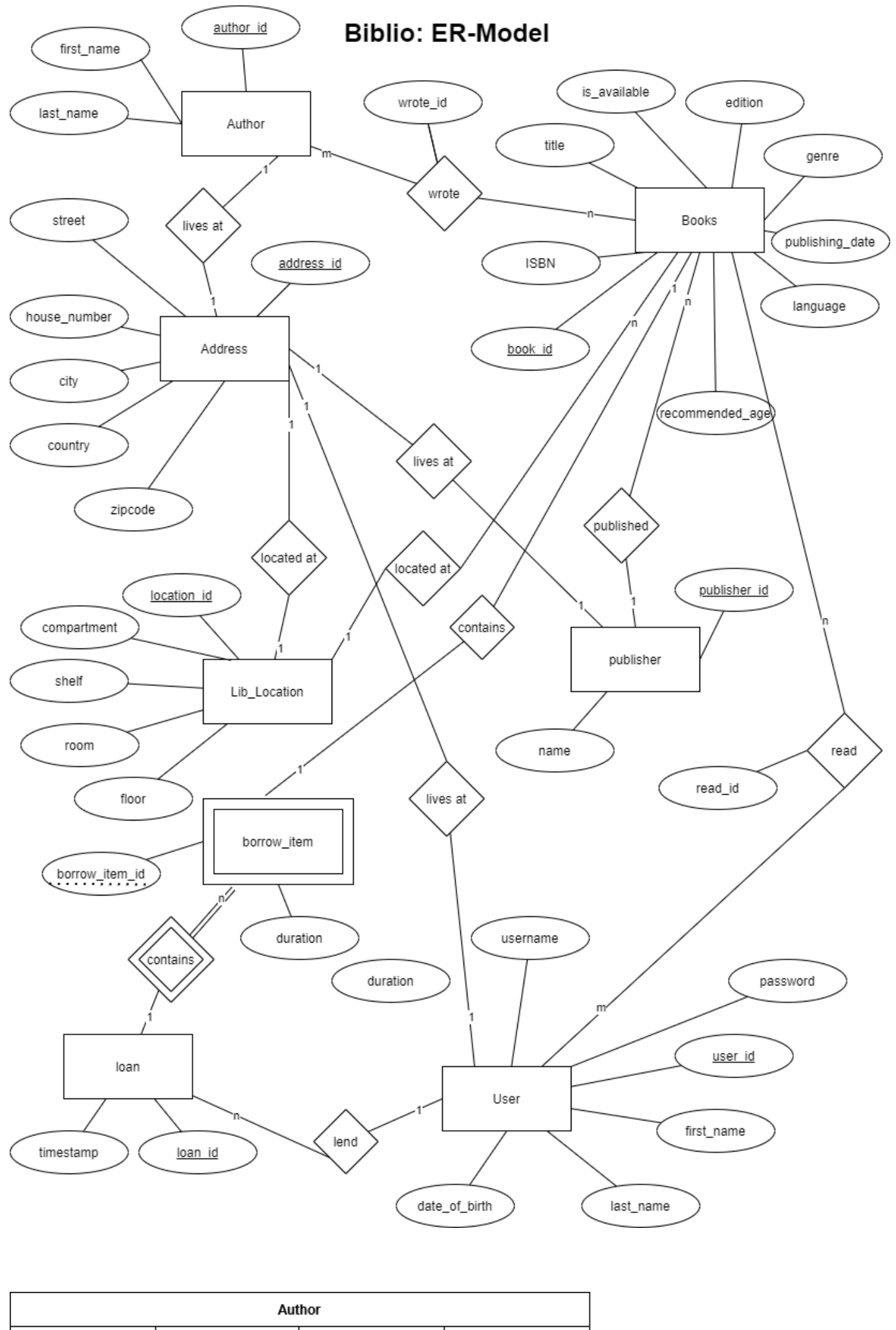
Return Book

Return via titel and edition:

title

edition

Return Book



author_id	first_name	last_name	address_id
0	Lukas	Benner	12
1	Phillip	Lange	5
2	Alina	Buss	31

Publisher			Lib_Location		
publisher_id	name	address_id	location_id	...	address_id
0	test	12	0	...	12
1	guteBücher	5	1	...	5
2	abcde	31	2	...	31

Address					
address_id	street	house_number	city	country	zipcode
0	test	12	Mannheim	Germany	13245
1	guteBücher	5	Stuttgart	Germany	98765
2	abcde	31	New York	USA	1234567

Books				
book_id	ISBN	...	publisher_id	location_id
0	012345	...	0	54
1	684354	...	5	8
2	123416	...	7	69

Borrow_item			
borrow_item_id	duration	book_id	loan_id
0	1	5	0
1	2	4	0
2	1	3	1

Loan			Read_books		
loan_id	timestamp	user_id	read_books_id	book_id	user_id
0	2020-10-20	5	0	87	5
1	2021-01-01	7	1	65	7
2	2020-04-08	10	2	74	10

User				
user_id	first_name	last_name	...	address_id
0	Lukas	Benner		12
1	Phillip	Lange	...	Null
2	Alina	Buss		31

wrote		
wrote_id	book_id	author_id

0	0	5
1	0	7
2	1	7

Database Modeling Explanation

1. atomic attributes

- There are only Single Valued Attributes.
- Attribute Domain does not change.
- There is a Unique name for every Attribute/Column.
- The order in which data is stored, does not matter.

Therefore, we did following:

- split address
- split names
- allowed several authors
- unique names inside a table

2. remove the repeated information

- Second Normal Form (2NF) is based on the concept of full functional dependency.

Therefore, we did following:

- delete address attributes
- make separate address table

3. No non-primary-key attribute is transitively dependent on the primary key

- A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

Therefore, we did following:

- made separate tables for author, publisher and connected it with the books table

Video

The video can be found on YouTube here: https://youtu.be/rB1QyN_s4V4

Links

- [Queries](#) (more can be found in the code)
- [SQL-File](#)
- [Github](#)