

NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Data Science and Business Analytics"

Library relational database

Made by:

Khaibrakhmanov Timur

Moscow 2023

1 Planned end users, users' needs

1.1 Librarians

- Keep track of all books in their responsibility
- See borrowed books
- See available books
- Recognize author and date of punishment of a particular book

1.2 Patrons

- Know which books are assigned to them
- See date when the book was borrowed
- See date when the book shall be returned
- Search database for available books

2 Functional requirements

- Adding/deleting new book
- Updating book properties
- Searching for a book by title/author/publisher
- Viewing all available books
- Adding/deleting patrons
- Updating patron's attributes
- Assign book to patron
- View due date for each assigned book

3 Data restrictions

- Book can have only one author (but there can be different publishers for one book)
- Book's status should be in available, unavailable
- All id's should be intra-unique
- Patrons with overdue should not be able to take a new book
- Due date can not be before the borrowing date
- Once a book is assigned to patron it cannot be reassigned to another patron, unless there are copies
- Patron's opportunity to take a book should be in able, unable - the status attribute

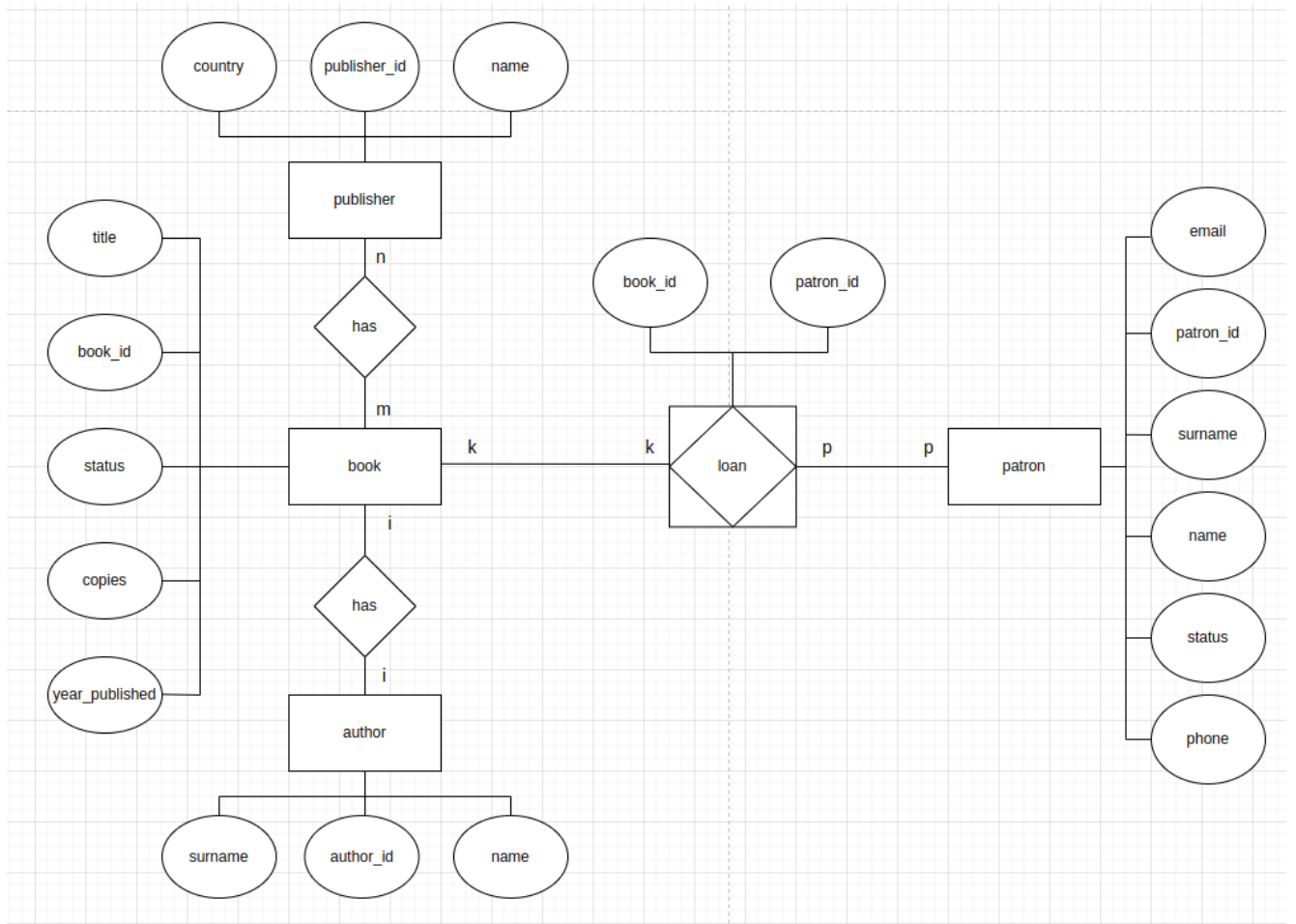
4 None-functional requirements

- Since the data on patrons is stored, it should be secure
- The data should be consistent and reliable
- High capacity, since there can be a lot of data

5 Functional and multi-valued dependencies

- book is in BCNF: $\text{book_id} \longrightarrow \text{title, status, copies, year_published}$
- author is in BCNF: $\text{author_id} \longrightarrow \text{name, surname}$
- publisher is in BCNF: $\text{publisher_id} \longrightarrow \text{name, country}$
- patron is in BCNF: $\text{patron_id} \longrightarrow \text{name, surname, phone, email, status}$
- loan is in BCNF: $\text{book_id, patron_id} \longrightarrow \text{book_id, patron_id}$

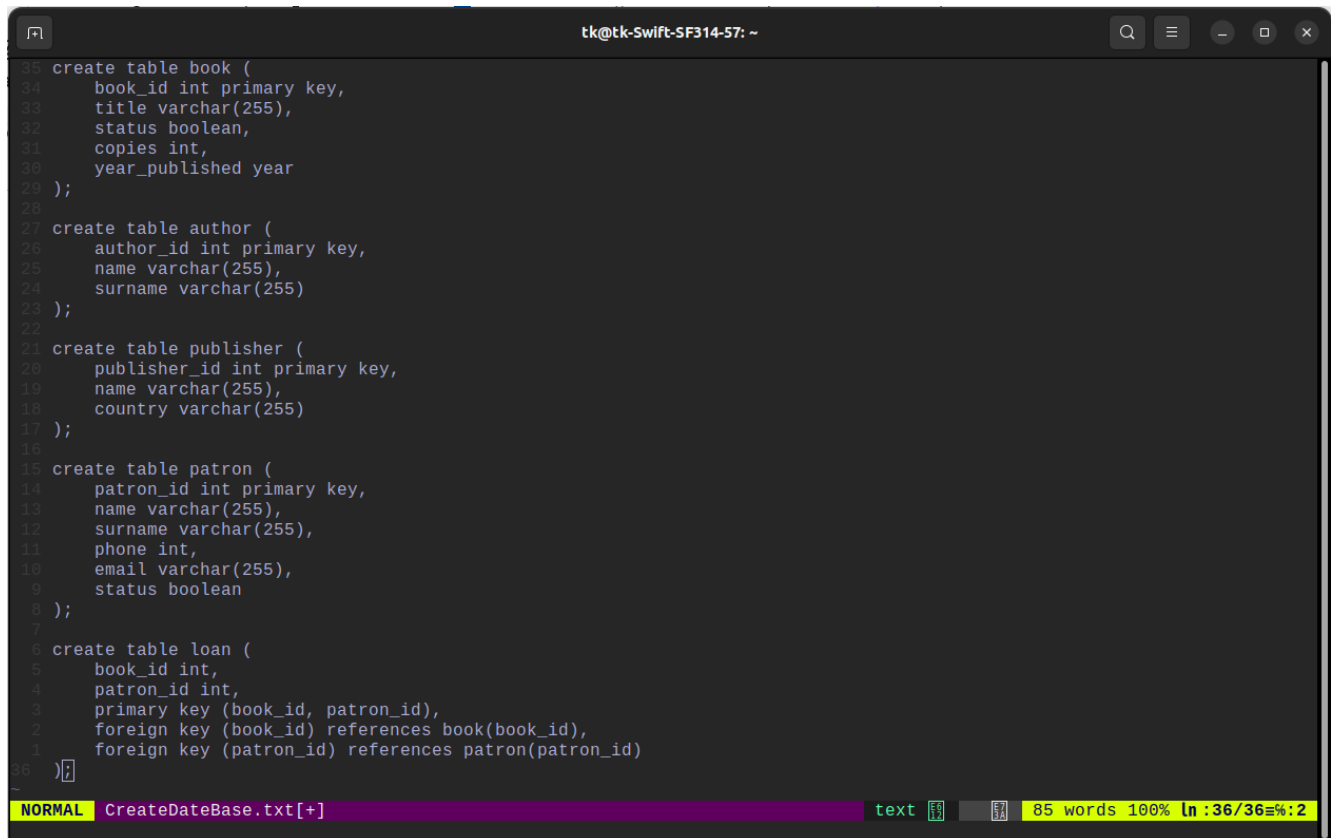
6 E/R Diagram



7 Normalized database scheme

- book: book_id, title, status, copies, year_published
- author: author_id, name, surname
- publisher: publisher_id, name, country
- patron: patron_id, name, surname, phone, email, status
- loan: book_id, patron_id

8 SQL DDL script for database creation based on normalized schema



```
tk@tk-Swift-SF314-57: ~  
35 create table book (  
34     book_id int primary key,  
33     title varchar(255),  
32     status boolean,  
31     copies int,  
30     year_published year  
29 );  
28  
27 create table author (  
26     author_id int primary key,  
25     name varchar(255),  
24     surname varchar(255)  
23 );  
22  
21 create table publisher (  
20     publisher_id int primary key,  
19     name varchar(255),  
18     country varchar(255)  
17 );  
16  
15 create table patron (  
14     patron_id int primary key,  
13     name varchar(255),  
12     surname varchar(255),  
11     phone int,  
10     email varchar(255),  
9     status boolean  
8 );  
7  
6 create table loan (  
5     book_id int,  
4     patron_id int,  
3     primary key (book_id, patron_id),  
2     foreign key (book_id) references book(book_id),  
1     foreign key (patron_id) references patron(patron_id)  
36 );
```

NORMAL CreateDateBase.txt[+] text 85 words 100% ln :36/36=2

9 SQL DML queries that implement functional requirements

9.1 Adding new book and deleting already existing book functional requirement

```
25
24 -- Insert book
23 create procedure AddBook (
22     in new_book_id id int,
21     in new_title varchar(255),
20     in new_status boolean,
19     in new_copies int,
18     in new_year_published int
17 )
16
15 begin
14     insert into book(book_id, title, status, copies, year_published)
13     values (new_book_id, new_title, new_status, new_copies, new_year_published);
12 end;
11
10
9 -- Delite book
8 create procedure DellBook (
7     in existing_book_id int
6 )
5
4 begin
3     delete from book
2     where book_id = existing_book_id;
1 end;
0
33
1
2
```

NORMAL CreateDateBase.txt[+] text 144 words 88% ln :63/71≡%:1

9.2 Adding new patron and deleting already existing patron functional requirement

```
9
10 -- Add patron
11 create procedure AddPatron (
12     in new_patron_id int primary key,
13     in new_name varchar(255),
14     in new_surname varchar(255),
15     in new_phone int,
16     in new_email varchar(255),
17     in new_status boolean
18 )
19 begin
20     insert into patron(patron_id, name, surname, phone, email, status)
21     values (new_patron_id, new_name, new_surname, new_phone, new_email, new_status);
22 end;
23
24 -- Delete patron
25 create procedure DellPatron (
26     in existing_patron_id int primary key
27 )
28 begin
29     delete from patron
30     where patron_id = existing_patron_id;
31 end;
32
33
34
35
36
```

NORMAL CreateDateBase.txt[+] text 215 words 57% ln :52/91=1 [50]mix...

9.3 Updating attributes of already existing patron functional requirement

```
17 -- Update patron
16 create procedure UpdatePatron (
15     in existing_patron_id int primary key,
14     in new_name varchar(255),
13     in new_surname varchar(255),
12     in new_phone int,
11     in new_email varchar(255),
10     in new_status boolean
9 )
8 begin
7     update patron
6     set name = new_name,
5         surname = new_surname,
4         phone = new_phone,
3         email = new_email,
2         status = new_status
1     where patron_id = existing_patron_id;
103 end;|
1
2
3
```

INSERT CreateDateBase.txt[+] text 267 words 87% ln :103/118=5 [87]tra...
-- INSERT --