



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт ИКБ направление 09.03.02 «Информационные системы и технологии»

Кафедра КБ-14 «Цифровые технологии обработки данных»

Дисциплина «Программные средства манипулирования данными»

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент: 3 курса, группа: БСБО-05-22

1. Тема: «Разработка базы данных для обмена книгами»

2. Срок представления проекта (работы) к защите _____ г.

3. Исходные данные для разработки

4. Содержание пояснительной записки:

Задание

Содержание

Описание предметной области

Список использованных источников

Приложения

Руководитель работы Котилевец И.Д.

Ф.И.О

подпись, дата

Задание принял к исполнению Домашова Е.А., Таранова В.В.

Ф.И.О

Москва 2024г.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«МИРЭА – Российский технологический университет»
РТУ МИРЭА**

Институт ИКБ направление 09.03.02 «Информационные системы и
технологии»

Кафедра КБ-14 «Цифровые технологии обработки данных»

Дисциплина «Программные средства манипулирования данными»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе (проект) на тему:
«Разработка базы данных для обмена книгами»

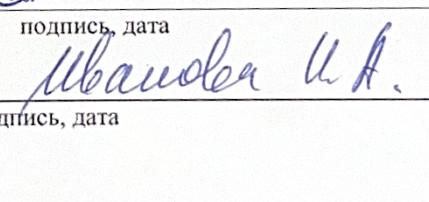
Студент Домашова Екатерина Алексеевна  Ф.И.О.
подпись, дата

Студент Таранова Валерия Викторовна  Ф.И.О.
подпись, дата

Группа БСБО-05-22

Работа защищена на оценку 5

Руководитель работы Котилевец И.Д.  Ф.И.О.
подпись, дата

Члены комиссии Марочкин  Иванова И.А.  Ф.И.О.
подпись, дата

Москва 2024г.

Содержание

Введение	4
Глава 1. Анализ предметной области.....	5
1.1. Общая характеристика	5
1.2. Функциональная структура	5
Глава 2. Модель предметной области	6
2.1. ER-диаграмма системы	6
2.2. Сущности.....	6
Глава 3. Проектирование базы данных.....	9
3.1. Создание базы данных	9
3.2. Заполнение таблиц данными	10
3.3. Реализация многопользовательского доступа.....	10
3.4. Автоматизация базы данных.....	13
Глава 4. Проектирование графического интерфейса.....	18
Заключение.....	24
Список использованных источников	26
Приложение.....	27
Приложение А. Листинг кода	27
Приложение Б. Интерфейс.....	35

Введение

Реляционная база данных – это способ организации данных, в котором данные организованы в логически связанные таблицы. Реляционная модель данных позволяет эффективно решать задачи безопасного хранения и обработки данных о сущностях, имеющих общую связь, подвергая изменениям только нужные таблицы и не нарушая целостность всей базы данных.

В рамках рассматриваемой предметной области уделяется особое внимание процессам, связанным с обменом книгами: добавлению и удалению книг из обменных списков, поиску возможных обменов и управлению информацией о совершенных обменах.

Целью настоящей курсовой работы является создание базы данных для обмена книгами, а также реализация графического интерфейса для упрощенного взаимодействия с базой данных. В процессе разработки базы данных будут рассмотрены следующие этапы:

- 1) анализ предметной области;
- 2) определение функциональных требований;
- 3) моделирование и проектирование базы данных;
- 4) реализация графического интерфейса.

Глава 1. Анализ предметной области

В настоящей главе проводится анализ выбранной предметной области, описывается общая характеристика и функциональная структура.

1.1. Общая характеристика

Социальное движение буккроссинг – это система, в рамках которой происходит свободный обмен книгами в специально обозначенных точках, находящихся в общественных местах, например, кафе, магазинах или библиотеках. При этом не обязательен контакт между лицом, отдающим книгу и лицом, принимающим книгу. Через специальные интернет-сервисы пользователь регистрирует книгу, после чего ей присваивается уникальный код (англ. BCID – Book Crossing IDentification) для дальнейшего отслеживания.

Упрощенной моделью буккроссинга является прямой обмен книгами между двумя людьми по предварительной договоренности. Однако такой способ имеет явный недостаток в виде необходимости самостоятельно искать обмены на специализированных форумах. С целью создать программное решение для такого способа обмена книгами в данной курсовой работе будет разработана информационная система и графический интерфейс для нее.

1.2. Функциональная структура

Разрабатываемое веб-приложение в выбранной предметной области должно предоставлять следующий функционал:

- регистрация нового пользователя,
- авторизация пользователя,
- добавление книг в список желаемых,
- добавление книг в список имеющихся,
- удаление книг из списка желаемых,
- удаление книг из списка имеющихся,
- добавление данных о совершенном обмене,
- просмотр доступных обменов,
- просмотр личной страницы других пользователей.

Глава 2. Модель предметной области

Во второй главе реализуется моделирование системы, создается ER-диаграмма системы, описываются ее сущности и их атрибуты.

2.1. ER-диаграмма системы

На основе анализа предметной области, проведенного в первой главе, была составлена ER-диаграмме разрабатываемой системы (см. рис. 1).

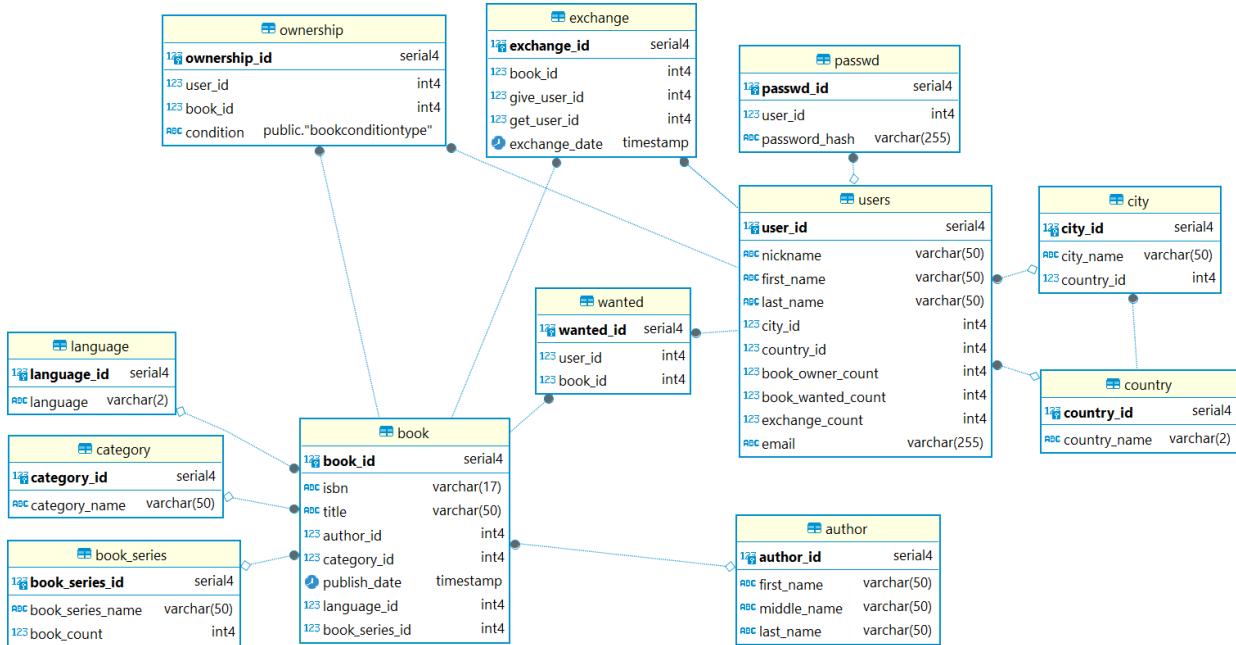


Рисунок 1. ER-диаграмма базы данных

2.2. Сущности

В таблице 1 представлена информация о всех сущностях в системе и описание к ним.

Таблица 1. Спецификация сущностей

№	Название сущности	Описание
1	book	Сущность содержит информацию о книге
2	author	Информация об авторе книги
3	category	Сущность для категории книги
4	book series	Информация о серии книг
5	language	Язык книги
6	users	Сущность содержит информацию о пользователе
7	passwd	Сущность для хранения пароля пользователя
8	wanted	Информация о желаемых книгах пользователя
9	ownership	Информация об имеющихся книгах у пользователя
10	country	Сущность для страны
11	city	Сущность для города
12	exchange	Информация о состоявшемся обмене между пользователями

В таблице 2 приведено описание всех атрибутов сущностей в базе данных.

Таблица 2. Атрибуты сущностей

Сущность	Название атрибута	Назначение атрибута	Тип	Примечание
book	book_id	Уникальный идентификатор книги	serial	Первичный ключ
	isbn	Международный стандартный книжный номер	varchar(17)	Обязательный, уникальный
	title	Название книги	varchar(50)	Обязательный
	author_id	ID автора книги	integer	Внешний ключ
	category_id	ID категории книги	integer	Внешний ключ
	publish_date	Дата публикации книги	timestamp	
	language_id	ID языка книги	integer	Внешний ключ
	book_series_id	ID серии книги	integer	Внешний ключ
author	author_id	Уникальный идентификатор автора	serial	Первичный ключ
	first_name	Имя автора	varchar(50)	
	middle_name	Отчество автора	varchar(50)	
	last_name	Фамилия автора	varchar(50)	Обязательный
category	category_id	Уникальный идентификатор категории	serial	Первичный ключ
	category_name	Название категории	varchar(50)	Обязательный
book_series	book_series_id	Уникальный идентификатор серии книг	serial	Первичный ключ
	book_series_name	Название серии книг	varchar(50)	Обязательный, уникальный
	book_series_count	Количество книг в серии	integer	
language	language_id	Уникальный идентификатор языка	serial	Первичный ключ
	language_name	Код языка ISO 639-1	varchar(2)	Обязательный
users	user_id	Уникальный идентификатор пользователя	serial	Первичный ключ
	nickname	Имя пользователя	varchar(50)	Обязательный, уникальный
	first_name	Имя	varchar(50)	
	last_name	Фамилия	varchar(50)	
	city_id	ID города пользователя	integer	Внешний ключ, обязательный
	country_id	ID страны пользователя	integer	Внешний ключ, обязательный
	book_owner_count	Количество имеющихся книг у пользователя	integer	
	book_wanted_count	Количество желаемых книг пользователя	integer	

	exchange_count	Количество совершенных обменов пользователя	integer	
	email	Электронный адрес пользователя	varchar(255)	Обязательный
passwd	passwd_id	Уникальный идентификатор пароля	serial	Первичный ключ
	user_id	ID пользователя	integer	
	password_hash	Хэш пароля пользователя	varchar(255)	Обязательный
wanted	wanted_id	Уникальный идентификатор записи о желаемой книге пользователя	serial	Первичный ключ
	user_id	ID пользователя	integer	Внешний ключ, обязательный
	book_id	ID книги	integer	Внешний ключ, обязательный
ownership	ownership_id	Уникальный идентификатор записи о имеющейся книге	serial	Первичный ключ
	user_id	ID пользователя	integer	Внешний ключ, обязательный
	book_id	ID книги	integer	Внешний ключ, обязательный
	condition	Состояние книги	BookConditionType	Внешний ключ, обязательный
country	country_id		serial	Первичный ключ
	country_name	Код страны ISO 3166-1 alpha-2	varchar(2)	Обязательный, уникальный
city	city_id	Уникальный идентификатор города	serial	Первичный ключ
	city_name	Название города	varchar(50)	Обязательный
	country_id	ID страны города	integer	Внешний ключ, обязательный
exchange	exchange_id	Уникальный идентификатор записи об обмене книги	serial	Первичный ключ
	book_id	ID книги	integer	Внешний ключ, обязательный
	give_user_id	ID пользователя, отдавшего книгу	integer	Внешний ключ, обязательный
	get_user_id	ID пользователя, получившего книгу	integer	Внешний ключ, обязательный
	exchange_date	Дата обмена	timestamp	

Глава 3. Проектирование базы данных

В данной главе будет рассмотрено проектирование базы данных для обмена книгами: создание базы данных, заполнение таблиц данными, реализация многопользовательского доступа и автоматизация базы данных.

3.1. Создание базы данных

Первым шагом была создана база данных book_exchange (см. листинг 1).

Листинг 1. Создание базы данных

```
CREATE DATABASE book_exchange;
```

Далее был создан собственный тип данных, отражающий состояние книги. Тип данных BookConditionType может принимать одно из пяти значений: новый, как новый, хороший, приемлемый, плохой (см. листинг 2).

Листинг 2. Создание нового типа данных

```
CREATE TYPE BookConditionType AS ENUM (
    'new',
    'like_new',
    'good',
    'acceptable',
    'bad'
);
```

В соответствии с составленной в предыдущей главе ER-диаграммой (см. рис. 1) были созданы связанные между собой таблицы. Создание таблицы показано на примере таблицы Book (см. листинг 3). Поле ISBN имеет длину 17 символов и является уникальным, название книги, поле title, не может быть длиннее 50 символов, поля author_id, category_id, language_id, book_series_id являются внешними ключами соответствующих таблиц Author, Category, Language и Book_series. Запросы на создание таблиц базы данных представлены в приложении А.1. Создание базы данных.

Листинг 3. Запрос на создание таблицы Book

```
CREATE TABLE book(
    book_id SERIAL PRIMARY KEY,
    isbn VARCHAR(17) UNIQUE NOT NULL,
    title VARCHAR(50) NOT NULL,
    author_id INTEGER REFERENCES author(author_id) ON DELETE SET NULL,
    category_id INTEGER REFERENCES category(category_id) ON DELETE SET NULL,
    publish_date TIMESTAMP,
```

```
language_id INTEGER REFERENCES language(language_id) ON DELETE SET NULL,
book_series_id INTEGER REFERENCES book_series(book_series_id) ON DELETE SET
NULL
);
```

3.2. Заполнение таблиц данными

После создания базы данных было выполнено заполнение таблиц данными в основном по 3 кортежа. Заполнение данными показано на примере таблицы Book (см. листинг 4). Запросы на заполнение таблиц данными представлены в приложении А.2. Заполнение таблиц данными.

Листинг 4. Заполнение данными таблицы Book

```
INSERT INTO book(isbn, title, author_id, category_id, publish_date, language_id,
book_series_id)
VALUES
    ('978-5-17-157450-5', 'Графики, которые убеждают всех', 1, 1, '2020-04-07', 1,
5),
    ('978-5-9268-2917-1', 'Стихотворения', 2, 2, '2020-12-21', 1, 4),
    ('978-5-04-120314-6', 'Принцип 80/20', 3, 1, '2012-01-01', 1, 3),
    ('978-5-17-112546-2', 'Братья Карамазовы', 4, 3, '2018-01-01', 1, 1),
    ('978-5-17-080085-8', 'О дивный новый мир', 5, 2, '2016-12-28', 1, 2);
```

3.3. Реализация многопользовательского доступа

Были созданы роли администратора, аналитика и системного пользователя (см. листинги 5-8).

Листинг 5. Создание роли администратора

```
CREATE ROLE administrator WITH
    LOGIN
    ENCRYPTED PASSWORD '8dmin'
    NOSUPERUSER
    NOCREATEDB
    CREATEROLE
    INHERIT
    NOREPLICATION
    CONNECTION LIMIT -1;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO administrator;
REVOKE TRUNCATE ON ALL TABLES IN SCHEMA public FROM administrator;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO administrator;
GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN SCHEMA public TO administrator;
```

Листинг 6. Создание роли аналитика

```
CREATE ROLE analyst WITH
    LOGIN
    ENCRYPTED PASSWORD '4n8lys'
    NOSUPERUSER
    NOCREATEDB
    NOCREATEROLE
    INHERIT
    NOREPLICATION
    CONNECTION LIMIT -1;

GRANT SELECT ON ALL TABLES IN SCHEMA public TO analyst;
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO analyst;
REVOKE ALL ON passwd FROM analyst;
```

Листинг 7. Создание групповых ролей для аналитиков

```
CREATE ROLE analyst_group;
GRANT analyst_group TO analyst;
```

Листинг 8. Создание роли системного пользователя

```
CREATE ROLE db_user WITH
    LOGIN
    ENCRYPTED PASSWORD 'postgres'
    NOSUPERUSER
    NOCREATEDB
    NOCREATEROLE
    NOINHERIT
    NOREPLICATION;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO db_user;
GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO db_user;
REVOKE UPDATE, DELETE ON passwd FROM db_user;
```

Для проверки корректности созданных прав доступа был открыт терминал SQL Shell (psql), в котором поочередно был осуществлен вход в каждую из созданных ролей.

На рисунке 2 представлена таблица с существующими ролями.

Имя роли	Список ролей	
	Атрибуты	
administrator	Создаёт роли, Создаёт БД	
analyst	Вход запрещён	
analyst_group		
db_user	Не наследуется	
postgres	Суперпользователь, Создаёт роли, Создаёт БД, Репликация, Пропускать RLS	

Рисунок 2. Таблица существующих ролей

По умолчанию кодировка консоли 866, ее необходимо изменить на 1251 для корректного отображения кириллицы. Это было реализовано с помощью команды: `! chcp 1251`.

Была осуществлена проверка прав доступа у роли администратора. Он может делать все в базе данных, кроме очищения таблиц командой `truncate`. Ему были даны такие права, так как структура базы данных сервиса может измениться, для чего могут понадобиться команды `create` и `drop` (см. рис. 3).

```
book_exchange=# \c - administrator
Пароль пользователя administrator:

Вы подключены к базе данных "book_exchange" как пользователь "administrator".
book_exchange=> TRUNCATE TABLE language
book_exchange=> ;
ОШИБКА: нет доступа к таблице language
book_exchange=> SELECT * FROM language;
language_id | language
-----+-----
 1 | ru
 2 | en
 3 | de
(3 строки)

book_exchange=>
```

Рисунок 3. Проверка роли администратора

Аналитик может только просматривать данные во всех таблицах, кроме `passwd`, которая содержит зашифрованные пароли пользователей (см. рис. 4).

```
book_exchange=> \c - analyst
Пароль пользователя analyst:

Вы подключены к базе данных "book_exchange" как пользователь "analyst".
book_exchange=> select * from passwd;
ОШИБКА: нет доступа к таблице passwd
book_exchange=> drop table author;
ОШИБКА: нужно быть владельцем таблицы author
book_exchange=> |
```

Рисунок 4. Проверка роли аналитика

Роль `db_user` была создана для взаимодействия с базой данных через веб-приложение. Данная роль имеет доступ к таблице `passwd`, но может только добавлять и просматривать данные таблицы (см. рис. 5).

```
book_exchange=> \c - db_user
Вы подключены к базе данных "book_exchange" как пользователь "db_user".
book_exchange=> select * from passwd;
passwd_id | user_id | password_hash
-----+-----+
(0 строк)

book_exchange=> drop table passwd;
ОШИБКА: нужно быть владельцем таблицы passwd
book_exchange=> |
```

Рисунок 5. Проверка роли `db_user`

3.4. Автоматизация базы данных

Была проведена автоматизация базы данных с использованием следующих инструментов: триггеры, функции, процедуры, представления.

Триггеры необходимы для автоматического подсчета значений в полях book_count (сколько книг в коллекции), exchange_count (сколько совершено обменов пользователем), book_wanted_count (сколько книг ищет пользователь), book_owner_count (сколько книг есть у пользователя) таблиц Book_series и Users. Создание триггера представлено на примере update_book_series_count, который учитывает создание и удалении книги и в зависимости от этого добавляет или убирает общее значение поля (см. листинг 9). Создание триггеров представлено в приложении А.3. Создание триггеров.

Листинг 9. Создание триггера update_book_series_count

```
CREATE OR REPLACE FUNCTION update_book_series_count()
RETURNS TRIGGER AS $$

BEGIN
    IF (TG_OP = 'INSERT') THEN
        UPDATE book_series
        SET book_count = book_count + 1
        WHERE book_series_id = NEW.book_series_id;
    ELSIF (TG_OP = 'DELETE') THEN
        UPDATE book_series
        SET book_count = book_count - 1
        WHERE book_series_id = OLD.book_series_id;
    END IF;
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER book_series_count_trigger
AFTER INSERT OR DELETE ON book
FOR EACH ROW
EXECUTE FUNCTION update_book_series_count();
```

Были написаны две функции: для извлечения информации о книге по ISBN – get_book_info, и для извлечения информации о пользователе по никнейму – get_user_info. В качестве примера рассмотрим создание функции get_book_info (см. листинг 10). Для отображения полной информации о книге к исходной таблице Book присоединяют другие необходимые таблицы по

внешним ключам. В приложении А.4. Создание функций также приведено создание функции get_user_info.

Листинг 10. Создание функции get_book_info.

```
CREATE OR REPLACE FUNCTION get_book_info(isbn_input VARCHAR)
RETURNS TABLE (
    isbn VARCHAR,
    title VARCHAR,
    author_name VARCHAR,
    category_name VARCHAR,
    publish_date TIMESTAMP,
    language VARCHAR,
    series_name VARCHAR,
    series_book_count INT
) AS $$

BEGIN
    RETURN QUERY
    SELECT
        b.isbn::VARCHAR,
        b.title::VARCHAR,
        CONCAT(a.first_name, ' ', COALESCE(a.middle_name, ''), ' ',
        a.last_name)::VARCHAR AS author_name,
        COALESCE(c.category_name, 'Unknown')::VARCHAR AS category_name,
        b.publish_date::TIMESTAMP,
        l.language::VARCHAR,
        COALESCE(bs.book_series_name, 'No series')::VARCHAR AS series_name,
        COALESCE(bs.book_count, 0)::INT AS series_book_count
    FROM book b
    LEFT JOIN author a ON b.author_id = a.author_id
    LEFT JOIN category c ON b.category_id = c.category_id
    LEFT JOIN language l ON b.language_id = l.language_id
    LEFT JOIN book_series bs ON b.book_series_id = bs.book_series_id
    WHERE b.isbn = isbn_input;
END;
$$ LANGUAGE plpgsql;
```

Проверим корректность выполнения функции получения информации о книге. Для этого выполним запрос, представленный на листинге 11.

Листинг 11. Вызов функции get_book_info

```
SELECT * FROM get_book_info('978-5-17-082803-6');
```

Результат выполнения функции представлен на рисунке 6.

	isbn character varying	title character varying	author_name character varying	category_name character varying	publish_date timestamp without time zone	language character varying	series_name character varying	series_book_count integer
1	978-5-17-082803-6	Тошнота	Жан Поль Сартр	Fiction	2019-08-28 00:00:00	ru	Эксклюзивная классика	2

Рисунок 6. Результат выполнения функции

Для автоматизации добавления книги была создана процедура add_book_bulk (см. листинг 12), которая добавляет записи, если таковых еще не было, в таблицы Author, Category, Language и Book_series – только после этого информация о новой книге будет добавлена в таблицу Book.

Листинг 12. Создание процедуры добавления книги

```
CREATE OR REPLACE PROCEDURE add_books_bulk(
    books JSONB
)
LANGUAGE plpgsql
AS $$

DECLARE
    book JSONB;
    author_id INT;
    category_id INT;
    language_id INT;
    book_series_id INT;

BEGIN
    FOR book IN SELECT * FROM jsonb_array_elements(books)
    LOOP
        INSERT INTO author (first_name, middle_name, last_name)
        VALUES (
            book->>'first_name',
            book->>'middle_name',
            book->>'last_name'
        )
        ON CONFLICT (first_name, middle_name, last_name) DO NOTHING;

        SELECT a.author_id INTO author_id
        FROM author a
        WHERE last_name = book->>'last_name';

        INSERT INTO category (category_name)
        VALUES (book->>'category')
        ON CONFLICT (category_name) DO NOTHING;

        SELECT c.category_id INTO category_id
        FROM category c
        WHERE category_name = book->>'category';

        INSERT INTO language (language)
        VALUES (book->>'language')
        ON CONFLICT (language) DO NOTHING;

        SELECT l.language_id INTO language_id
        FROM language l
        WHERE language = book->>'language';
    END LOOP;
END;
```

```

    INSERT INTO book_series (book_series_name)
    VALUES (book->>'book_series')
    ON CONFLICT (book_series_name) DO NOTHING;

    SELECT bs.book_series_id INTO book_series_id
    FROM book_series bs
    WHERE book_series_name = book->>'book_series';

    INSERT INTO book (isbn, title, author_id, category_id, publish_date,
language_id, book_series_id)
    VALUES (
        book->>'isbn',
        book->>'title',
        author_id,
        category_id,
        TO_TIMESTAMP(book->>'publish_date', 'YYYY-MM-DD HH24:MI:SS'),
        language_id,
        book_series_id
    )
    ON CONFLICT (isbn) DO NOTHING;
END LOOP;
END;
$$;

```

Проверим корректность работы процедуры путем ее вызова (см. листинг 13).

Листинг 13. Вызов процедуры

```

CALL add_books_bulk(
  [
    {
      "isbn": "978-5-6046934-9-0",
      "title": "The Invisible Man",
      "first_name": "Herbert",
      "middle_name": "George",
      "last_name": "Wells",
      "category": "Novel",
      "publish_date": "2021-12-24",
      "language": "en",
      "book_series": "MovieBook"
    }
  ] ::JSONB
);

```

После вызова процедуры с параметрами новой книги проверим корректность заполнения таблиц данными с помощью функции просмотра информации о книге, созданной ранее (см. листинг 14).

Листинг 14. Вызов функции get_book_info

```
SELECT * FROM get_book_info('978-5-6046934-9-0');
```

Результат выполнения функции get_book_info представлен на рисунке 7.

	isbn character varying	title character varying	author_name character varying	category_name character varying	publish_date timestamp without time zone	language character varying	series_name character varying	series_book_count integer
1	978-5-6046934-9-0	The Invisible Man	Herbert George Wells	Novel	2021-12-24 00:00:00	en	MovieBook	1

Рисунок 7. Проверка результата вызова процедуры

Было добавлено представление exchange_statistics в качестве отчета для использования аналитиком (см. листинг 15).

Листинг 15. Создание представления exchange_statistics

```
CREATE VIEW exchange_statistics AS
SELECT
    u.user_id,
    COALESCE(give_books.books_given, 0) AS books_given,
    COALESCE(get_books.books_received, 0) AS books_received
FROM users u
LEFT JOIN (
    SELECT
        give_user_id AS user_id,
        COUNT(*) AS books_given
    FROM exchange
    GROUP BY give_user_id
) AS give_books ON u.user_id = give_books.user_id
LEFT JOIN (
    SELECT
        get_user_id AS user_id,
        COUNT(*) AS books_received
    FROM exchange
    GROUP BY get_user_id
) AS get_books ON u.user_id = get_books.user_id;
```

Наполнение представления представлено на рисунке 8.

	user_id integer	books_given bigint	books_received bigint
1	1	1	1
2	3	2	2
3	2	1	1

Рисунок 8. Представление exchange_statistics

Глава 4. Проектирование графического интерфейса

Серверная часть веб-приложения для обмена книгами была написана на Node.js с использованием фреймворка Express, JSONWebToken для авторизации и ORM Sequelize для работы с базой данных.

Клиентская часть была разработана с использованием HTML, CSS и JavaScript. Для работы с CSS была использована методология БЭМ [9].

При открытии веб-приложения пользователь попадает на приветственную страницу с кнопками «вход» и «регистрация» (см. рис. 9).



Рисунок 9. Страница входа

После нажатия на кнопку «регистрация» открывается окно с регистрацией нового пользователя (см. рис. 10). Пользователь должен заполнить поля регистрационной формы, обязательными из которых являются имя пользователя, адрес электронной почты и пароль.

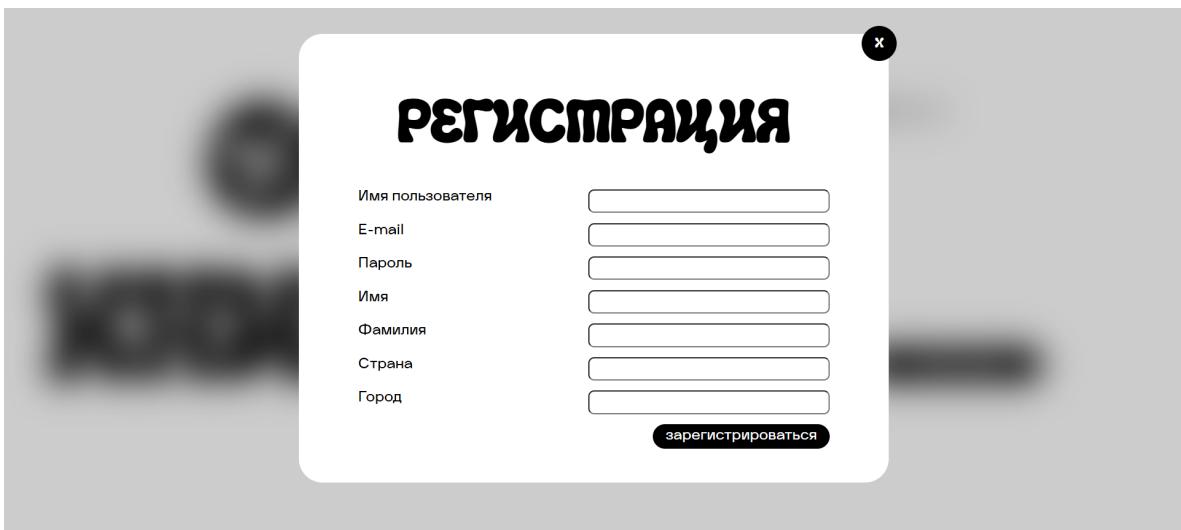


Рисунок 10. Окно регистрации

Окно входа открывается при нажатии на соответствующую кнопку (см. рис. 11). В данном окне пользователю необходимо вести логин, имя пользователя и пароль, которые он указывал при регистрации.

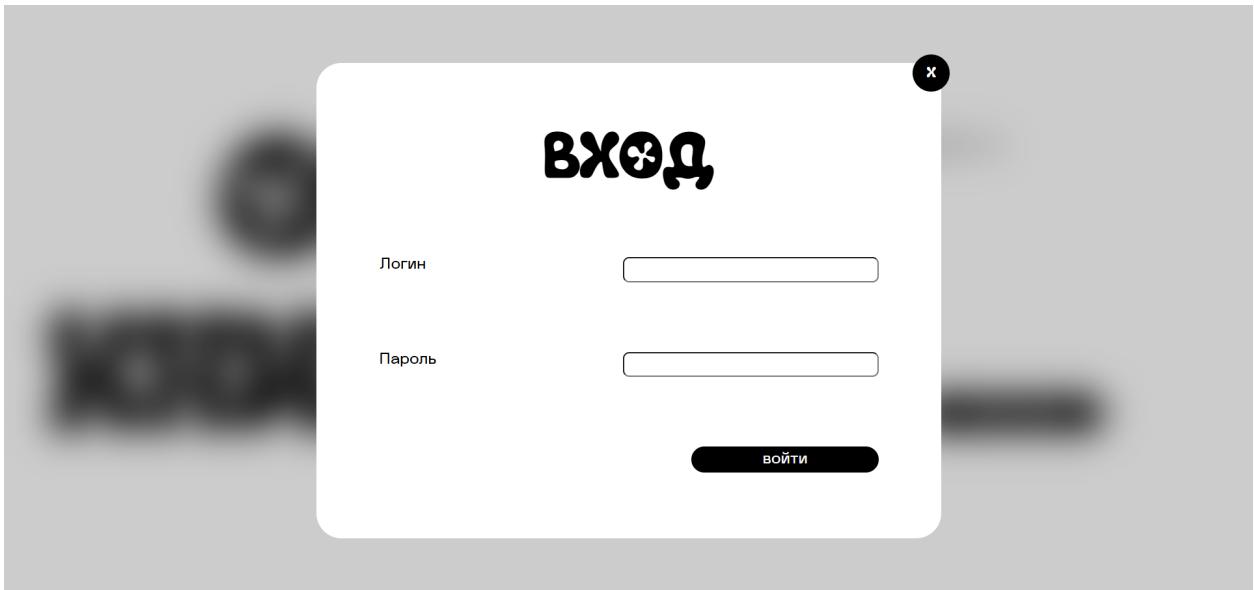


Рисунок 11. Окно входа

После успешной регистрации и входа в систему пользователь попадает на страницу со своим профилем, на которой отображается информация о пользователе, а также информация о желаемых и имеющихся книгах (см. рис. 12-14).

A screenshot of a user profile page titled "ОБМЕН КНИГАМИ" (Book Exchange). The page has a light gray background. At the top, there are two links: "поиск обменов" (Search for swaps) and "мой профиль" (My profile). The main feature is a large, bold title "MARSS4". Below it, there are three statistics: "Книг отдает: 1", "Книг ищет: 2", and "Всего обменов: 3". On the right side, there is a user information block with the text "Нет имени" (No name), "mars@d.ru", "RU", and "Москва" (Moscow). At the bottom, there are two buttons: "добавить обмен" (Add swap) and "выйти из профиля" (Logout).

Рисунок 12. Профиль авторизованного пользователя

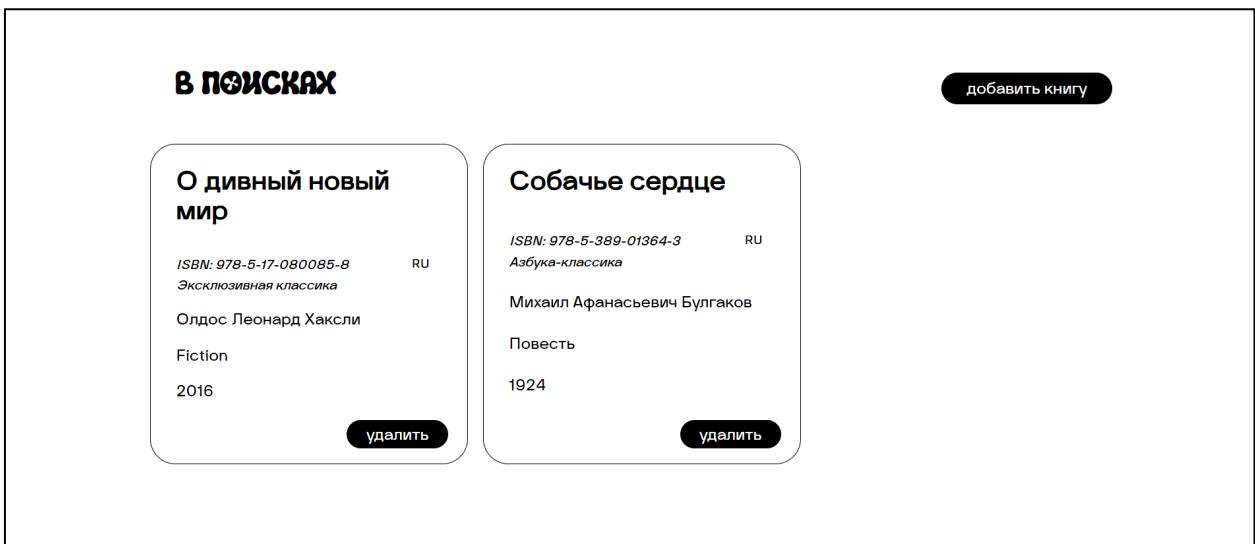


Рисунок 13. Раздел желаемых книг авторизованного пользователя

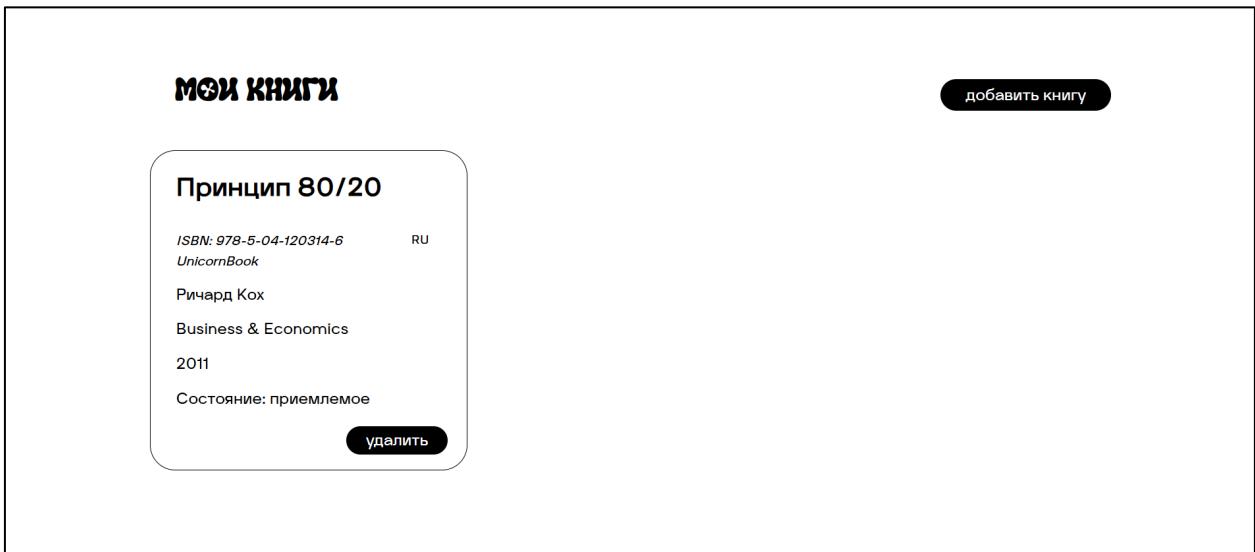


Рисунок 14. Имеющиеся книги авторизованного пользователя

При нажатии на кнопку «выйти из профиля» будет осуществлен выход пользователя из профиля и откроется страница входа, описанная выше (см. рис. 12).

На странице «мой профиль» авторизованный пользователь может добавить обмен, нажав на кнопку «добавить обмен» (см. рис. 12). В открывшемся окне пользователю необходимо ввести имя пользователя, с которым был совершен обмен, и ISBN книги, участвовавшей в обмене (см. рис. 15). После этого новая запись об обмене будет добавлена в базу данных и количество обменов у задействованных в нем пользователей обновится.



Рисунок 15. Добавление нового обмена

После нажатия на кнопку «добавить книгу» в профиле авторизованного пользователя в разделе «в поисках» (см. рис. 13) откроется окно для добавления желаемой книги (см. рис. 16).

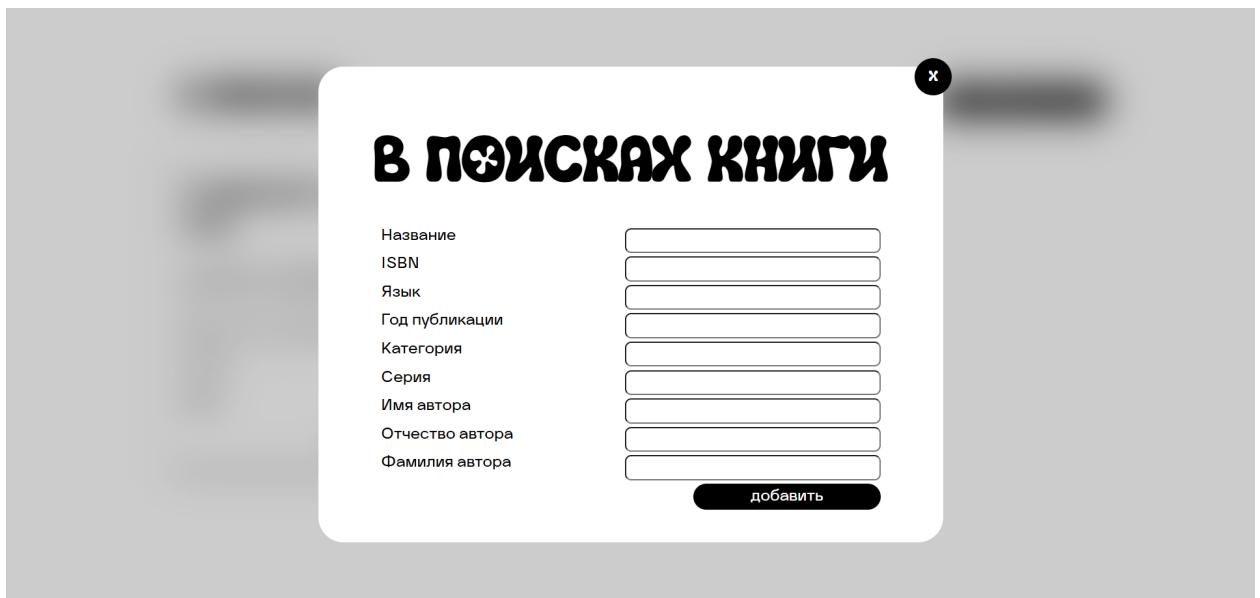


Рисунок 16. Добавление новой желаемой книги

При нажатии на кнопку «добавить книгу» в разделе «мои книги» в профиле авторизованного пользователя (см. рис. 14) откроется окно для добавление новой книги в имеющиеся (см. рис. 17). Пользователю необходимо заполнить информацию о книге, в том числе указать ее состояние.



Рисунок 17. Окно добавления новой книги в имеющиеся

В шапке на странице профиля авторизованного пользователя содержится название системы и навигация в виде двух кнопок: «поиск обменов» и «мой профиль» (см. рис. 12). Последняя окрашена в серый цвет, что дает пользователю понять, что он уже находится на этой странице. При нажатии на кнопку «поиск обменов» будет загружена страница с возможными обменами (см. рис. 18). На странице представлены профили других пользователей и книги, которые они меняют и ищут.

ОБМЕН КНИГАМИ		
		поиск обменов мой профиль
Пользователь:	Меняет:	Ищет:
aidar RU, Москва aidar007agent@mail.ru Совершено обменов: 3	Принцип 80/20	О дивный новый мир
Alex337 RU, Москва alex337@gmail.com Совершено обменов: 5	Вам ничего не нужно	О дивный новый мир

Рисунок 18. Страница поиска обменов

При нажатии на имя пользователя в списке будет совершен переход на его профиль (см. рис. 19-21). В профиле другого пользователя доступен адрес его электронной почты для связи с целью осуществления обмена. Также в

профиле доступны обменные списки данного пользователя, включающие в себя доступные к обмену и желаемые книги.

The screenshot shows a user profile page. At the top, there is a navigation bar with the text 'ОБМЕН КНИГАМИ' on the left and 'поиск обменов' and 'мой профиль' on the right. Below the navigation bar, the word 'AIDAR' is displayed in large, bold, black letters. To the left of 'AIDAR', there is some user statistics: 'Книг отдает: 2', 'Книг ищет: 1', and 'Всего обменов: 3'. To the right of 'AIDAR', there is contact information: 'Айдар', 'aidar007agent@mail.ru', 'RU', and 'Москва'.

Рисунок 19. Профиль неавторизованного пользователя

The screenshot shows the 'Having' section of the user profile. The title 'МЕНЯЕМ' is at the top. Below it, a box contains the title 'Принцип 80/20'. Inside the box, there are details about the book: ISBN: 978-5-04-120314-6, RU, UnicornBook, author: Ричард Кох, category: Business & Economics, year: 2011.

Рисунок 20. Раздел имеющихся книг неавторизованного пользователя

The screenshot shows the 'Want' section of the user profile. The title 'ИЩЕМ' is at the top. Below it, two boxes are shown. The first box is titled 'Графики, которые убеждают всех' and contains details about the book: ISBN: 978-5-17-157450-5, RU, Бизнес-наставник, author: Александр Богачев, category: Business & Economics, year: 2020, condition: плохое. The second box is titled 'О дивный новый мир' and contains details about the book: ISBN: 978-5-17-080085-8, RU, Эксклюзивная классика, author: Олдос Леонард Хаксли, category: Fiction, year: 2016, condition: новая.

Рисунок 21. Раздел желаемых книг неавторизованного пользователя

Заключение

В настоящей курсовой работе была разработана база данных для обмена книгами и реализован графический интерфейс, предоставляющий упрощенное взаимодействие и многопользовательский доступ к ней.

В первой главе была рассмотрена общая характеристика предметной области и определена функциональная структура веб-приложения.

Во второй главе было рассмотрено моделирование системы. На основе анализа предметной области, выполненного в первой главе, была создана ER-диаграмма системы. Были описаны 12 сущностей системы: book, author, category, book_series, language, users, passwd, wanted, ownership, country, city, exchange. Также дано описание всем атрибутам сущностей, включая информацию о том, какое назначение и тип имеет атрибут.

В третьей главе была спроектирована база данных. Создание таблиц базы данных было выполнено в соответствии описанным сущностям и атрибутам во второй главе. Созданные таблицы были заполнены данными по три и более кортежей. Был реализован многопользовательский доступ: созданы роли администратора, аналитика и системного пользователя, а также проверена корректность созданных ролей. База данных была автоматизирована с помощью триггеров для автоматических подсчетов полей таблиц, функций для просмотра полной информации о книгах и пользователях, процедуры для добавления записи о новой книге, представления для использования аналитиком в качестве отчета и индексации. Также в третьей главе были представлены примеры используемых листингов, которые более подробно можно рассмотреть в приложении А.

В четвертой главе было описано проектирование графического интерфейса, а также продемонстрированы реализованные механики взаимодействия. Для наглядности были прикреплены скриншоты интерфейса, более подробно представленные в приложении Б.

Все поставленные задачи выполнены, цель достигнута – была создана база данных для обмена книгами с использованием СУБД реляционного типа – PostgreSQL.

Список использованных источников

1. PostgreSQL: Documentation [Электронный ресурс] URL: <https://www.postgresql.org/docs/> Дата обращения (01.11.2024).
2. Node.JS | Паттерн MVC. Контроллеры [Электронный ресурс] URL: <https://metanit.com/web/nodejs/7.1.php> Дата обращения (17.12.2024).
3. Sequelize, Node.js и PostgreSQL: примеры использования ORM [Электронный ресурс] URL: <https://proglib.io/p/gibkaya-orm-dlya-node-js-sequelize-2022-10-12> Дата обращения (17.12.2024).
4. Идеальный инструмент для работы с СУБД без SQL для Node.js или Все, что вы хотели знать о Sequelize. Часть 1 / Хабр [Электронный ресурс] URL: <https://habr.com/ru/articles/565062/> Дата обращения (17.12.2024).
5. Идеальный инструмент для работы с СУБД без SQL для Node.js или Все, что вы хотели знать о Sequelize. Часть 2 / [Электронный ресурс] URL: Хабр <https://habr.com/ru/articles/566036/> Дата обращения (17.12.2024).
6. Associations | Sequelize [Электронный ресурс] URL: <https://sequelize.org/docs/v6/core-concepts/assocs/> Дата обращения (17.12.2024).
7. Raw Queries | Sequelize [Электронный ресурс] URL: <https://sequelize.org/docs/v6/core-concepts/raw-queries/> Дата обращения (17.12.2024).
8. Атрибут autocomplete — HTML — Дока [Электронный ресурс] URL: <https://doka.guide/html/autocomplete/> Дата обращения (17.12.2024).
9. Быстрый старт / Методология / БЭМ [Электронный ресурс] URL: <https://ru.bem.info/methodology/quick-start/> Дата обращения (19.12.2024).
- 10.<nav>: The Navigation Section element - HTML: HyperText Markup Language | MDN [Электронный ресурс] URL: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav> Дата обращения (19.12.2024).
- 11.[Электронный ресурс] URL: Дата обращения (19.12.2024).

Приложение

Приложение А. Листинг кода

Приложение А.1. Создание базы данных

```
CREATE TABLE category(
    category_id SERIAL PRIMARY KEY,
    category_name VARCHAR(50) NOT NULL UNIQUE
);

CREATE TABLE language(
    language_id SERIAL PRIMARY KEY,
    language VARCHAR(2) NOT NULL UNIQUE
);

CREATE TABLE book_series(
    book_series_id SERIAL PRIMARY KEY,
    book_series_name VARCHAR(50) NOT NULL UNIQUE,
    book_count INTEGER DEFAULT 0 CONSTRAINT positive_value_book_count CHECK
(book_count >= 0)
);

CREATE TABLE country(
    country_id SERIAL PRIMARY KEY,
    country_name VARCHAR(2) NOT NULL UNIQUE
);

CREATE TABLE city(
    city_id SERIAL PRIMARY KEY,
    city_name VARCHAR(50) NOT NULL,
    country_id INTEGER NOT NULL REFERENCES country(country_id) ON DELETE CASCADE
);

CREATE TABLE author(
    author_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    middle_name VARCHAR(50),
    last_name VARCHAR(50) NOT NULL
);

CREATE TABLE book(
    book_id SERIAL PRIMARY KEY,
    isbn VARCHAR(17) UNIQUE NOT NULL,
    title VARCHAR(50) NOT NULL,
    author_id INTEGER REFERENCES author(author_id) ON DELETE SET NULL,
    category_id INTEGER REFERENCES category(category_id) ON DELETE SET NULL,
    publish_date TIMESTAMP,
    language_id INTEGER REFERENCES language(language_id) ON DELETE SET NULL,
    book_series_id INTEGER REFERENCES book_series(book_series_id) ON DELETE SET
NULL
```

```

);

CREATE TABLE users(
    user_id SERIAL PRIMARY KEY,
    nickname VARCHAR(50) NOT NULL UNIQUE,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    city_id INTEGER NOT NULL REFERENCES city(city_id) ON DELETE SET NULL,
    country_id INTEGER NOT NULL REFERENCES country(country_id) ON DELETE SET NULL,
    book_owner_count INTEGER DEFAULT 0 CHECK (book_owner_count >= 0),
    book_wanted_count INTEGER DEFAULT 0 CHECK (book_wanted_count >= 0),
    exchange_count INTEGER DEFAULT 0 CHECK (exchange_count >= 0),
    email VARCHAR(255) NOT NULL
);

CREATE TABLE passwd(
    passwd_id SERIAL PRIMARY KEY,
    user_id INTEGER REFERENCES users(user_id) ON DELETE CASCADE,
    password_hash VARCHAR(255) NOT NULL
);

CREATE TABLE wanted(
    wanted_id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
    book_id INTEGER NOT NULL REFERENCES book(book_id) ON DELETE CASCADE
);

CREATE TABLE ownership(
    ownership_id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
    book_id INTEGER NOT NULL REFERENCES book(book_id) ON DELETE CASCADE,
    condition BookConditionType NOT NULL
);

CREATE TABLE exchange(
    exchange_id SERIAL PRIMARY KEY,
    book_id INTEGER NOT NULL REFERENCES book(book_id) ON DELETE SET NULL,
    give_user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE SET NULL,
    get_user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE SET NULL,
    exchange_date TIMESTAMP
);

```

Приложение А.2. Заполнение таблиц данными

```

INSERT INTO category(category_name)
VALUES
    ('Business & Economics'),
    ('Fiction'),
    ('Novel');

INSERT INTO language(language)

```

```

VALUES
('ru'),
('en'),
('de');

INSERT INTO book_series(book_series_name)
VALUES
('Лучшая мировая классика'),
('Эксклюзивная классика'),
('UnicornBook'),
('Малая Классика Речи'),
('Бизнес-наставник');

INSERT INTO country(country_name)
VALUES
('RU'),
('US'),
('GB'),
('DE');

INSERT INTO city(city_name, country_id)
VALUES
('Москва', 1),
('Санкт-Петербург', 1),
('Калининград', 1),
('Краснодар', 1),
('Ухта', 1);

INSERT INTO author(first_name, middle_name, last_name, country_id)
VALUES
('Александр', NULL, 'Богачев', 1),
('Осип', NULL, 'Мандельштам', 1),
('Ричард', NULL, 'Кох', 3),
('Федор', 'Михайлович', 'Достоевский', 1),
('Олдос', 'Леонард', 'Хаксли', 3);

INSERT INTO book(isbn, title, author_id, category_id, publish_date, language_id,
book_series_id)
VALUES
('978-5-17-157450-5', 'Графики, которые убеждают всех', 1, 1, '2020-04-07', 1,
5),
('978-5-9268-2917-1', 'Стихотворения', 2, 2, '2020-12-21', 1, 4),
('978-5-04-120314-6', 'Принцип 80/20', 3, 1, '2012-01-01', 1, 3),
('978-5-17-112546-2', 'Братья Карамазовы', 4, 3, '2018-01-01', 1, 1),
('978-5-17-080085-8', 'О дивный новый мир', 5, 2, '2016-12-28', 1, 2);

INSERT INTO users(nickname, first_name, last_name, city_id, country_id, email)
VALUES
('Orenwolf56', 'Владимир', 'Вареньев', 2, 1, 'varenik_v@bk.com'),
('aidar', 'Айдар', NULL, 1, 1, 'aidar007agent@mail.ru'),

```

```

('Alex337', NULL, NULL, NULL, 1, 'alex337@gmail.com');

INSERT INTO wanted(user_id, book_id)
VALUES
(1, 1),
(2, 3),
(3, 4);

INSERT INTO ownership(user_id, book_id, condition)
VALUES
(1, 3, 'new'),
(3, 1, 'like_new'),
(1, 4, 'good'),
(3, 5, 'acceptable'),
(3, 2, 'bad');

INSERT INTO exchange(book_id, give_user_id, get_user_id, exchange_date)
VALUES
(2, 2, 3, '2024-01-01'),
(5, 1, 3, '2023-05-07'),
(3, 3, 1, '2024-11-11');

```

Приложение А.3. Создание триггеров

```

CREATE OR REPLACE FUNCTION update_book_series_count()
RETURNS TRIGGER AS $$

BEGIN
    IF (TG_OP = 'INSERT') THEN
        UPDATE book_series
        SET book_count = book_count + 1
        WHERE book_series_id = NEW.book_series_id;
    ELSIF (TG_OP = 'DELETE') THEN
        UPDATE book_series
        SET book_count = book_count - 1
        WHERE book_series_id = OLD.book_series_id;
    END IF;
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER book_series_count_trigger
AFTER INSERT OR DELETE ON book
FOR EACH ROW
EXECUTE FUNCTION update_book_series_count();

```

```

CREATE OR REPLACE FUNCTION update_exchange_count()
RETURNS TRIGGER AS $$

BEGIN
    UPDATE users
    SET exchange_count = exchange_count + 1

```

```

        WHERE user_id = NEW.give_user_id OR user_id = NEW.get_user_id;
        RETURN NULL;
    END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER exchange_count_trigger
AFTER INSERT ON exchange
FOR EACH ROW
EXECUTE FUNCTION update_exchange_count();

CREATE OR REPLACE FUNCTION update_wanted_count()
RETURNS TRIGGER AS $$

BEGIN
    IF (TG_OP = 'INSERT') THEN
        UPDATE users
        SET book_wanted_count = book_wanted_count + 1
        WHERE user_id = NEW.user_id;
    ELSIF (TG_OP = 'DELETE') THEN
        UPDATE users
        SET book_wanted_count = book_wanted_count - 1
        WHERE user_id = OLD.user_id;
    END IF;
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER wanted_count_trigger
AFTER INSERT OR DELETE ON wanted
FOR EACH ROW
EXECUTE FUNCTION update_wanted_count();

CREATE OR REPLACE FUNCTION update_owner_count()
RETURNS TRIGGER AS $$

BEGIN
    IF (TG_OP = 'INSERT') THEN
        UPDATE users
        SET book_owner_count = book_owner_count + 1
        WHERE user_id = NEW.user_id;
    ELSIF (TG_OP = 'DELETE') THEN
        UPDATE users
        SET book_owner_count = book_owner_count - 1
        WHERE user_id = OLD.user_id;
    END IF;
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER owner_count_trigger

```

```

AFTER INSERT OR DELETE ON ownership
FOR EACH ROW
EXECUTE FUNCTION update_owner_count();

```

Приложение А.4. Создание функций

```

CREATE OR REPLACE FUNCTION get_book_info(isbn_input VARCHAR)
RETURNS TABLE (
    isbn VARCHAR,
    title VARCHAR,
    author_name VARCHAR,
    category_name VARCHAR,
    publish_date TIMESTAMP,
    language VARCHAR,
    series_name VARCHAR,
    series_book_count INT
) AS $$

BEGIN
    RETURN QUERY
    SELECT
        b.isbn::VARCHAR,
        b.title::VARCHAR,
        CONCAT(a.first_name, ' ', COALESCE(a.middle_name, ''), ' ',
        a.last_name)::VARCHAR AS author_name,
        COALESCE(c.category_name, 'Unknown')::VARCHAR AS category_name,
        b.publish_date::TIMESTAMP,
        l.language::VARCHAR,
        COALESCE(bs.book_series_name, 'No series')::VARCHAR AS series_name,
        COALESCE(bs.book_count, 0)::INT AS series_book_count
    FROM book b
    LEFT JOIN author a ON b.author_id = a.author_id
    LEFT JOIN category c ON b.category_id = c.category_id
    LEFT JOIN language l ON b.language_id = l.language_id
    LEFT JOIN book_series bs ON b.book_series_id = bs.book_series_id
    WHERE b.isbn = isbn_input;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION get_users_info(nickname_input VARCHAR)
RETURNS TABLE (
    nickname VARCHAR,
    email VARCHAR,
    name VARCHAR,
    city VARCHAR,
    country VARCHAR,
    book_owner_count INT,
    book_wanted_count INT,
    exchange_count INT
) AS $$

BEGIN
    RETURN QUERY

```

```

SELECT
    u.nickname::VARCHAR,
    u.email::VARCHAR,
    CONCAT(COALESCE(u.first_name, 'No name'), ' ', COALESCE(u.last_name, 'No
last name'))::VARCHAR AS name,
    COALESCE(a.city_name, 'Unknown')::VARCHAR AS city,
    COALESCE(c.country_name, 'Unknown')::VARCHAR AS country,
    u.book_owner_count::INT,
    u.book_wanted_count::INT,
    u.exchange_count::INT
FROM users u
LEFT JOIN city a ON u.city_id = a.city_id
LEFT JOIN country c ON u.country_id = c.country_id
WHERE u.nickname = nickname_input;
END;
$$ LANGUAGE plpgsql;

```

Приложение А.5. Создание процедуры добавления книги

```

CREATE OR REPLACE PROCEDURE add_books_bulk(
    books JSONB
)
LANGUAGE plpgsql
AS $$

DECLARE
    book JSONB;
    author_id INT;
    category_id INT;
    language_id INT;
    book_series_id INT;
BEGIN
    FOR book IN SELECT * FROM jsonb_array_elements(books)
    LOOP
        INSERT INTO author (first_name, middle_name, last_name)
        VALUES (
            book->>'first_name',
            book->>'middle_name',
            book->>'last_name')
        ON CONFLICT (first_name, middle_name, last_name) DO NOTHING;

        SELECT a.author_id INTO author_id
        FROM author a
        WHERE last_name = book->>'last_name';
        INSERT INTO category (category_name)
        VALUES (book->>'category')
        ON CONFLICT (category_name) DO NOTHING;

        SELECT c.category_id INTO category_id
        FROM category c
        WHERE category_name = book->>'category';

```

```

INSERT INTO language (language)
VALUES (book->>'language')
ON CONFLICT (language) DO NOTHING;

SELECT l.language_id INTO language_id
FROM language l
WHERE language = book->>'language';

INSERT INTO book_series (book_series_name)
VALUES (book->>'book_series')
ON CONFLICT (book_series_name) DO NOTHING;

SELECT bs.book_series_id INTO book_series_id
FROM book_series bs
WHERE book_series_name = book->>'book_series';

INSERT INTO book (isbn, title, author_id, category_id, publish_date,
language_id, book_series_id)
VALUES (
    book->>'isbn',
    book->>'title',
    author_id,
    category_id,
    TO_TIMESTAMP(book->>'publish_date', 'YYYY-MM-DD HH24:MI:SS'),
    language_id,
    book_series_id
)
ON CONFLICT (isbn) DO NOTHING;
END LOOP;
END;
$$;

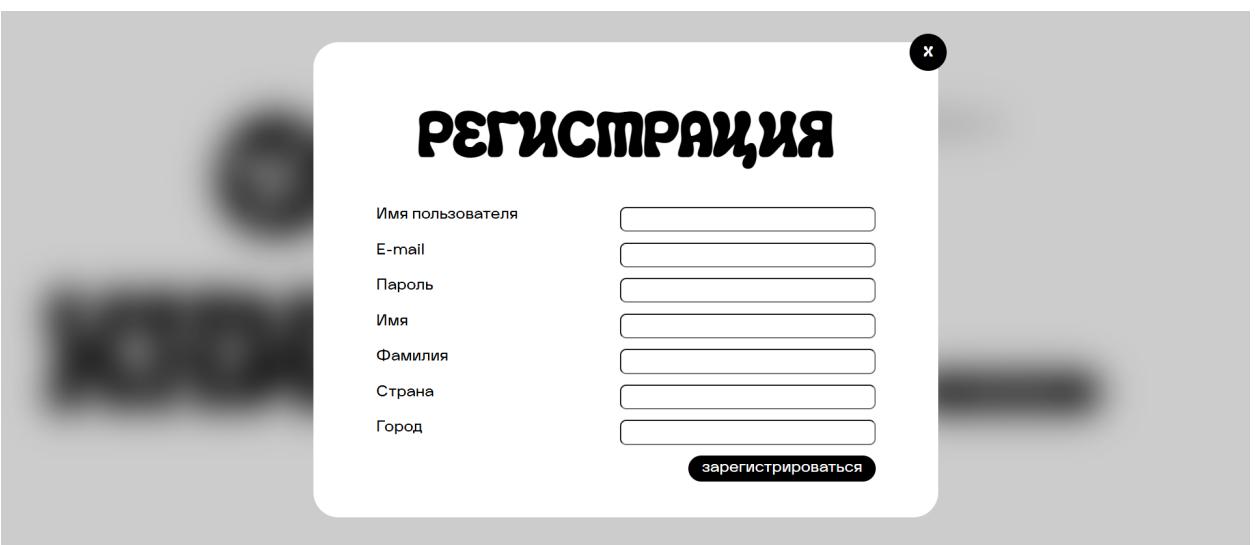
```

Приложение Б. Интерфейс

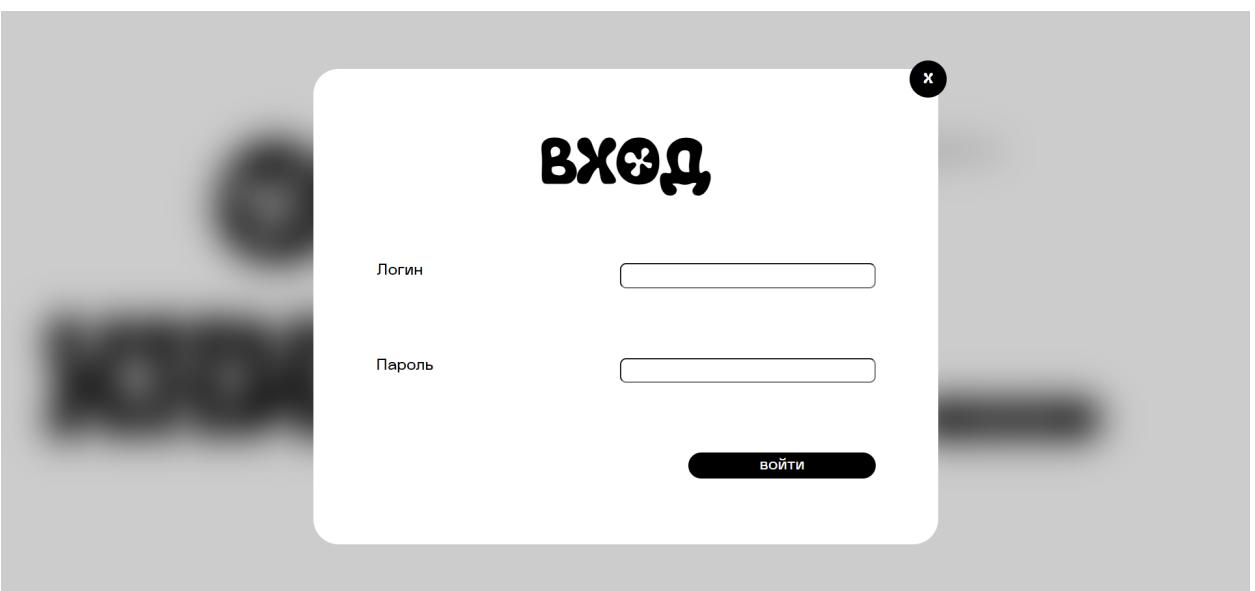
Приложение Б.1. Страница входа в веб-приложение



Приложение Б.2. Окно регистрации нового пользователя



Приложение Б.3. Окно входа в веб-приложение



Приложение Б.4. Профиль авторизованного пользователя

The screenshot shows a user profile page. At the top, there's a navigation bar with the text 'ОБМЕН КНИГАМИ' on the left and 'поиск обменов' and 'мой профиль' on the right. The main title 'MARSS4' is displayed prominently in large, bold, black letters. Below the title, there are three status indicators: 'Книг отдает: 1', 'Книг ищет: 2', and 'Всего обменов: 3'. To the right, there's contact information: 'Нет имени', 'mars@d.ru', 'RU', and 'Москва'. Two buttons at the bottom are 'добавить обмен' on the left and 'выйти из профиля' on the right.

Приложение Б.5. Желаемые книги авторизованного пользователя

This screenshot shows the 'Wanted books' section. The heading 'В ПОИСКАХ' is at the top. On the right, there's a button 'добавить книгу'. Below, two book entries are listed in boxes: 'О дивный новый мир' by Олдос Леонард Хаксли (ISBN: 978-5-17-080085-8, RU) and 'Собачье сердце' by Михаил Афанасьевич Булгаков (ISBN: 978-5-389-01364-3, RU). Each entry includes the author's name, genre ('Fiction' or 'Повесть'), year ('2016' or '1924'), and a 'удалить' (Delete) button.

Приложение Б.6. Имеющиеся книги авторизованного пользователя

This screenshot shows the 'My books' section. The heading 'МОИ КНИГИ' is at the top. On the right, there's a button 'добавить книгу'. One book entry is visible: 'Принцип 80/20' by Ричард Кох (ISBN: 978-5-04-120314-6, RU, UnicornBook). The entry includes the author's name, genre ('Business & Economics'), year ('2011'), and a 'удалить' (Delete) button.

Приложение Б.7. Добавление нового обмена

Новый обмен

Имя пользователя, с которым был обмен

ISBN книги

добавить

This screenshot shows a modal window titled "Новый обмен" (New Swap). It contains two input fields: one for the user's name and another for the ISBN of the book being swapped. A "добавить" (add) button is located at the bottom right.

Приложение Б.8. Добавление новой желаемой книги

В поисках книги

Название
ISBN
Язык
Год публикации
Категория
Серия
Имя автора
Отчество автора
Фамилия автора

добавить

This screenshot shows a modal window titled "В поисках книги" (In search of a book). It lists several fields for entering book details: title, ISBN, language, publication year, category, series, author's first name, middle name, and last name. A "добавить" (add) button is at the bottom right.

Приложение Б.9. Добавление имеющейся книги

Моя новая книга

Название
ISBN
Язык
Год публикации
Состояние книги
Категория
Серия
Имя автора
Отчество автора
Фамилия автора

новая

добавить

This screenshot shows a modal window titled "Моя новая книга" (My new book). It includes fields for entering book details: title, ISBN, language, publication year, condition (with "новая" selected), category, series, author's first name, middle name, and last name. A dropdown menu for condition is shown with "новая" selected. A "добавить" (add) button is at the bottom right.

Приложение Б.10. Страница поиска обменов

ОБМЕН КНИГАМИ

поиск обменов мой профиль

Пользователь:	Меняет:	Ищет:
aidar RU, Москва aidar007agent@mail.ru Совершено обменов: 3	Принцип 80/20	О дивный новый мир
Alex337 RU, Москва alex337@gmail.com Совершено обменов: 5	Вам ничего не нужно	О дивный новый мир

Приложение Б.11. Профиль неавторизованного пользователя

ОБМЕН КНИГАМИ

поиск обменов мой профиль

AIDAR

Книг отдает: 2 Айдар
Книг ищет: 1 aidar007agent@mail.ru
Всего обменов: 3 RU
 Москва

Приложение Б.12. Желаемые книги неавторизованного пользователя

МЕНЯЕТ

Принцип 80/20

ISBN: 978-5-04-120314-6 RU
UnicornBook

Ричард Кох

Business & Economics

2011

Приложение Б.13. Имеющиеся книги неавторизованного пользователя

ищет

Графики, которые убеждают всех	О дивный новый мир
<i>ISBN: 978-5-17-157450-5</i>	<i>ISBN: 978-5-17-080085-8</i>
<i>Бизнес-наставник</i>	<i>Эксклюзивная классика</i>
Александр Богачев	Олдос Леонард Хаксли
Business & Economics	Fiction
2020	2016
Состояние: плохое	Состояние: новая