

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
РГУ им. А. Н. Косыгина  
(Технологии. Дизайн. Искусство.)

Институт Информационных технологий и цифровой трансформации

Кафедра «Искусственного интеллекта,  
прикладной математики и программирования»

КУРСОВОЙ ПРОЕКТ  
по дисциплине «Базы данных и программирование»

на тему: Задачи управления базой данных

**Выполнил:** студент группы ИТПМ-124

Молчанов Даниил Максимович

(Фамилия, имя, отчество)

(подпись)

**Принял:** доцент кафедры ИИПМиП

Мокряков Алексей Викторович (Фамилия, имя, отчество)

(подпись)

**Оценка:**

**Дата:**

Москва, 2025

# Содержание

<b>1 Введение</b>	<b>4</b>
<b>2 Техническое задание</b>	<b>4</b>
<b>3 Проектирование базы данных</b>	<b>4</b>
3.1 Создание базы данных . . . . .	4
3.2 Создание таблицы author . . . . .	5
3.3 Создание таблицы publishing_house . . . . .	5
3.4 Создание таблицы book . . . . .	5
3.5 Создание таблицы reader . . . . .	6
3.6 Создание таблицы book_instance . . . . .	6
3.7 Создание таблицы issuance . . . . .	6
3.8 Создание таблицы booking . . . . .	7
<b>4 Реализация бизнес-логики</b>	<b>7</b>
4.1 Управление авторами (CRUD операции) . . . . .	7
4.2 Управление издательствами . . . . .	9
4.3 Управление информацией о книгах . . . . .	11
4.4 Управление читателями . . . . .	13
4.5 Управление экземплярами книг . . . . .	14
4.6 Сценарий выдачи книги . . . . .	15
4.7 Сценарий возврата книги . . . . .	16
4.8 Представление выданных книг . . . . .	17
4.9 Представление просроченных выдач . . . . .	17
4.10 Модификация выдачи с проверкой просрочек . . . . .	18
4.11 Сценарий бронирования книги . . . . .	19
4.12 Сценарий отмены бронирования . . . . .	20
4.13 Модификация выдачи с проверкой бронирований . . . . .	22
4.14 Функция поиска местоположений книги . . . . .	24
4.15 Представление доступных экземпляров . . . . .	24
4.16 Представление долгосрочных выдач . . . . .	25
4.17 Создание таблицы логов . . . . .	26
4.18 Реализация триггеров для логгирования . . . . .	27

5	Заключение	30
6	Список использованных источников	31

# 1 Введение

В данном разделе описывается цель курсовой работы, решаемые задачи и структура документа.

## 2 Техническое задание

Разработка системы управления библиотекой на основе реляционной базы данных PostgreSQL. Система должна обеспечивать:

- Управление каталогом книг, авторов и издательств
- Учет читателей и их истории выдачи
- Управление физическими экземплярами книг с учетом их состояния
- Бронирование книг читателями с ограничениями
- Логирование операций в системе
- Формирование отчетов и представлений
- Проверку корректности операций (просрочки, доступность)
- Автоматическое обновление статусов экземпляров
- Поиск книг по различным критериям

## 3 Проектирование базы данных

### 3.1 Создание базы данных

```
1 CREATE DATABASE t01_library;
```

Листинг 1: Создание базы данных t01\_library

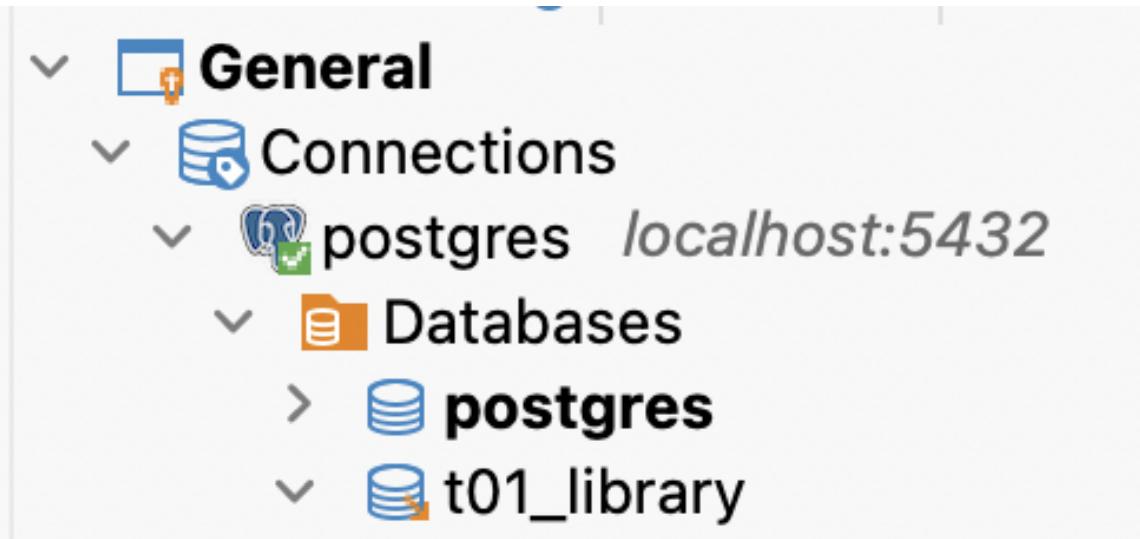


Рис. 1: Результат создания базы данных

### 3.2 Создание таблицы author

```
1 create table public.author(
2   id serial primary key,
3   first_name varchar(80) not null,
4   last_name varchar(80) not null
5 );
```

Листинг 2: Создание таблицы author

### 3.3 Создание таблицы publishing\_house

```
1 create table public.publishing_house(
2   id SERIAL primary key,
3   name varchar(80) not null,
4   city varchar(80) not null
5 );
```

Листинг 3: Создание таблицы publishing\_house

### 3.4 Создание таблицы book

```
1 create table public.book(
2   id serial primary key,
3   name_book varchar(80) not null,
4   author int references public.author(id),
```

```
5 publisher_code int references public.publishing_house(id),  
6 edition int not null,  
7 year_publication int,  
8 circulation int  
9 );
```

Листинг 4: Создание таблицы book

### 3.5 Создание таблицы reader

```
1 create type sex as enum ('male', 'female');  
2  
3 create table public.reader(  
4 id serial primary key,  
5 first_name varchar(80) not null,  
6 last_name varchar(80) not null,  
7 date_of_birth date,  
8 gender sex,  
9 date_of_registration date  
10 );
```

Листинг 5: Создание таблицы reader

### 3.6 Создание таблицы book\_instance

```
1 create type availability as enum ('in stock', 'issued', 'reserved');  
2 create type state as enum ('excellent', 'good', 'satisfactory', 'dilapidated', 'lost'  
3 );  
4  
4 create table public.book_instance(  
5 id serial primary key,  
6 book_information int references public.book(id),  
7 state state,  
8 status availability,  
9 location varchar(80) not null  
10 );
```

Листинг 6: Создание таблицы book\_instance

### 3.7 Создание таблицы issuance

```
1 create table public.issuance(
2     reader_id int references public.reader(id),
3     book_id int references public.book_instance(id),
4     date_time timestamp,
5     date_return_expected date not null,
6     date_return_fact date NULL
7 );
```

Листинг 7: Создание таблицы issuance

### 3.8 Создание таблицы booking

```
1 create table booking(
2     id serial primary key,
3     number_reader_ticket int references public.reader(id),
4     book_information int references public.book(id),
5     data_time timestamp,
6     min_state state
7 );
```

Листинг 8: Создание таблицы booking

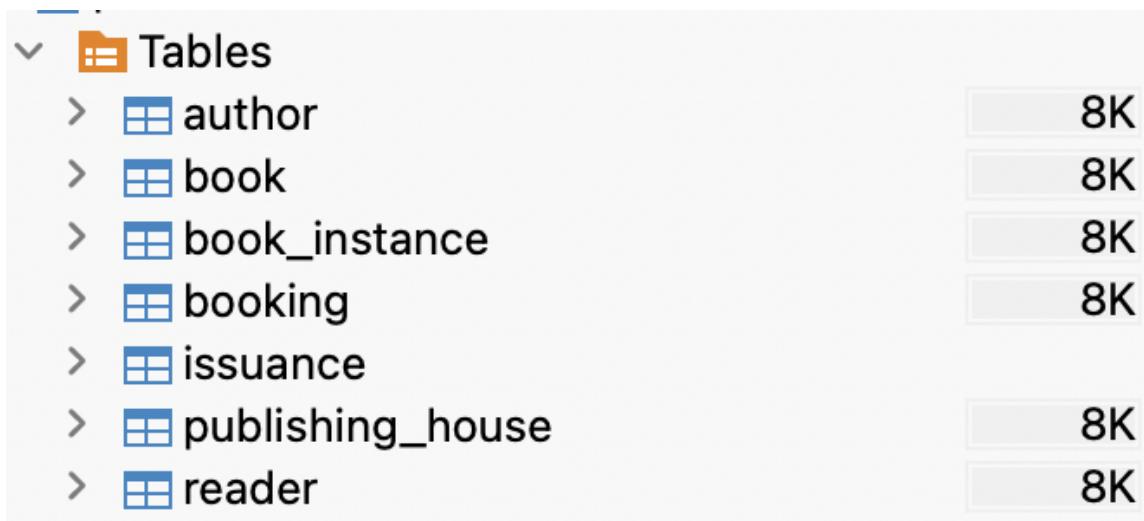


Рис. 2: Результат создания таблиц

## 4 Реализация бизнес-логики

### 4.1 Управление авторами (CRUD операции)

```

1  INSERT INTO author (first_name, last_name) VALUES
2  ('Александр', 'Пушкин'),
3  ('Михаил', 'Булгаков'),
4  ('Лев', 'Толстой'),
5  ('Федор', 'Достоевский'),
6  ('Антон', 'Чехов'),
7  ('Николай', 'Гоголь'),
8  ('Иван', 'Тургенев'),
9  ('Владимир', 'Набоков'),
10 ('Сергей', 'Есенин'),
11 ('Анна', 'Ахматова')
12 RETURNING id;
13
14 UPDATE author SET last_name='Ахматова-Горенко' WHERE first_name='Анна';
15
16 DELETE FROM author WHERE id IN (2, 5);

```

Листинг 9: CRUD операции для таблицы author

	123 ↗ id	A-Z first_name	A-Z last_name
1	1	Александр	Пушкин
2	2	Михаил	Булгаков
3	3	Лев	Толстой
4	4	Федор	Достоевский
5	5	Антон	Чехов
6	6	Николай	Гоголь
7	7	Иван	Тургенев
8	8	Владимир	Набоков
9	9	Сергей	Есенин
10	10	Анна	Ахматова

Рис. 3: Результат операции INSERT

	123 ↞ id	A-Z first_name	A-Z last_name
1	1	Александр	Пушкин
2	2	Михаил	Булгаков
3	3	Лев	Толстой
4	4	Федор	Достоевский
5	5	Антон	Чехов
6	6	Николай	Гоголь
7	7	Иван	Тургенев
8	8	Владимир	Набоков
9	9	Сергей	Есенин
10	10	Анна	Ахматова-Горенко

Рис. 4: Результат операции UPDATE

	123 ↞ id	A-Z first_name	A-Z last_name
1	1	Александр	Пушкин
2	3	Лев	Толстой
3	4	Федор	Достоевский
4	6	Николай	Гоголь
5	7	Иван	Тургенев
6	8	Владимир	Набоков
7	9	Сергей	Есенин
8	10	Анна	Ахматова-Горенко

Рис. 5: Результат операции DELETE

## 4.2 Управление издательствами

```

1 INSERT INTO publishing_house (name, city) VALUES
2 ('Солнышко', 'Москва'),
3 ('АСТ', 'Москва'),
4 ('Эксмо', 'Санкт-Петербург'),
5 ('Просвещение', 'Москва'),
6 ('Росмэн', 'Казань'),
7 ('Дрофа', 'Москва'),

```

```

8   ('Фламинго', 'Санкт-Петербург'),
9   ('Олма-Пресс', 'Новосибирск'),
10  ('Азбука', 'Москва'),
11  ('Амфора', 'Екатеринбург')
12 RETURNING id;
13
14 UPDATE publishing_house SET name='Просвещение+' WHERE id=4;
15
16 DELETE FROM publishing_house WHERE id IN (2, 4, 6);

```

Листинг 10: CRUD операции для таблицы publishing\_house

	123 ↗ id	A-Z name	A-Z city
1	1	Солнышко	Москва
2	2	АСТ	Москва
3	3	Эксмо	Санкт-Петербург
4	4	Просвещение	Москва
5	5	Росмэн	Казань
6	6	Дрофа	Москва
7	7	Фламинго	Санкт-Петербург
8	8	Олма-Пресс	Новосибирск
9	9	Азбука	Москва
10	10	Амфора	Екатеринбург

Рис. 6: Результат операций с издательствами

	123 ↗ id	A-Z name	A-Z city
1	1	Солнышко	Москва
2	2	АСТ	Москва
3	3	Эксмо	Санкт-Петербург
4	5	Росмэн	Казань
5	6	Дрофа	Москва
6	7	Фламинго	Санкт-Петербург
7	8	Олма-Пресс	Новосибирск
8	9	Азбука	Москва
9	10	Амфора	Екатеринбург
10	4	Просвещение+	Москва

Рис. 7: Дополнительный результат для издательств

	123 ↗ id	A-Z name	A-Z city
1	1	Солнышко	Москва
2	3	Эксмо	Санкт-Петербург
3	5	Росмэн	Казань
4	7	Фламинго	Санкт-Петербург
5	8	Олма-Пресс	Новосибирск
6	9	Азбука	Москва
7	10	Амфора	Екатеринбург

Рис. 8: Финальный результат для издательств

### 4.3 Управление информацией о книгах

```

1  INSERT INTO book (name_book, author, publisher_code, edition, year_publication,
2   circulation) VALUES
3   ('Колобок', 1, 1, 1, 2012, 5000),
4   ('Маша и медведь', 1, 3, 1, 2002, 5200),

```

```

4 ('Мастер и Маргарита', 2, 2, 1, 1966, 10000),
5 ('Война и мир', 3, 3, 1, 1869, 8000),
6 ('Преступление и наказание', 4, 1, 2, 1866, 7000),
7 ('Евгений Онегин', 1, 2, 3, 1833, 12000),
8 ('Мертвые души', 6, 3, 1, 1842, 6000),
9 ('Отцы и дети', 7, 1, 2, 1862, 5500),
10 ('Лолита', 8, 2, 1, 1955, 3000),
11 ('Анна Каренина', 3, 3, 2, 1877, 9000)
12 RETURNING id;
13
14 UPDATE book SET name_book='Маша и три медведя' WHERE id=2;
15
16 DELETE FROM book WHERE id IN (9,10);

```

Листинг 11: CRUD операции для таблицы book

	123 ↗ id	AZ name_book	123 author	123 publisher_soc	123 edition	123 year_publicat	123 circulation
1	1	Колобок		1	1	2,012	5,000
2	2	Маша и медведь		1	3	2,002	5,200
3	3	Мастер и Маргарит:		2	2	1,966	10,000
4	4	Война и мир		3	3	1,869	8,000
5	5	Преступление и нак		4	1	1,866	7,000
6	6	Евгений Онегин		1	2	1,833	12,000
7	7	Мертвые души		6	3	1,842	6,000
8	8	Отцы и дети		7	1	1,862	5,500
9	9	Лолита		8	2	1,955	3,000
10	10	Анна Каренина		3	3	1,877	9,000

Рис. 9: Результат операций с книгами - часть 1

	123 ↗ id	AZ name_book	123 author	123 publisher_soc	123 edition	123 year_publicat	123 circulation
1	1	Колобок		1	1	2,012	5,000
2	3	Мастер и Маргарит:		2	2	1,966	10,000
3	4	Война и мир		3	3	1,869	8,000
4	5	Преступление и нак		4	1	1,866	7,000
5	6	Евгений Онегин		1	2	1,833	12,000
6	7	Мертвые души		6	3	1,842	6,000
7	8	Отцы и дети		7	1	1,862	5,500
8	9	Лолита		8	2	1,955	3,000
9	10	Анна Каренина		3	3	1,877	9,000
	2	Маша и три медвед:		1	3	2,002	5,200

Рис. 10: Результат операций с книгами - часть 2

	123 ↗ id	AZ name_book	123 author	123 publisher_soc	123 edition	123 year_publicat	123 circulation
1	3	Мастер и Маргарит:		2	2	1,966	10,000
2	4	Война и мир		3	3	1,869	8,000
3	5	Преступление и нак		4	1	1,866	7,000
4	6	Евгений Онегин		1	2	1,833	12,000
5	7	Мертвые души		6	3	1,842	6,000
6	8	Отцы и дети		7	1	1,862	5,500
7	9	Лолита		8	2	1,955	3,000
8	10	Анна Каренина		3	3	1,877	9,000

Рис. 11: Результат операций с книгами - часть 3

## 4.4 Управление читателями

```

1  INSERT INTO reader (first_name, last_name, date_of_birth, gender,
2    date_of_registration) VALUES
3  ('Даниил', 'Молчанов', '2006-10-25', 'male', CURRENT_DATE),
4  ('Иван', 'Иванов', '1950-10-24', 'male', CURRENT_DATE),
5  ('Алексей', 'Перов', '1980-09-21', 'male', CURRENT_DATE),
6  ('Мария', 'Иванова', '2000-05-15', 'female', '2024-01-10'),
7  ('Ольга', 'Петрова', '1995-12-01', 'female', CURRENT_DATE - INTERVAL '5 days'),
8  ('Сергей', 'Сидоров', '1975-03-18', 'male', CURRENT_DATE - INTERVAL '10 days'),
9  ('Екатерина', 'Кузнецова', '1988-07-30', 'female', CURRENT_DATE - INTERVAL '3 days'),
10 ('Дмитрий', 'Васильев', '1992-11-11', 'male', '2024-01-15'),
11 ('Анастасия', 'Попова', '1999-04-22', 'female', CURRENT_DATE),
12 ('Михаил', 'Соколов', '1965-08-14', 'male', CURRENT_DATE - INTERVAL '7 days')
13
14 RETURNING id;
15
16 UPDATE reader SET date_of_registration='2024-01-20' WHERE id=10;
17
18 DELETE FROM reader WHERE id IN (1,2,3);

```

Листинг 12: CRUD операции для таблицы reader

	l23 id	AZ first_name	AZ last_name	Ø date_of_birth	AZ gender	Ø date_of_registration
1	1	Даниил	Молчанов	2006-10-25	male	2026-01-12
2	2	Иван	Иванов	1950-10-24	male	2026-01-12
3	3	Алексей	Перов	1980-09-21	male	2026-01-12
4	4	Мария	Иванова	2000-05-15	female	2024-01-10
5	5	Ольга	Петрова	1995-12-01	female	2026-01-07
6	6	Сергей	Сидоров	1975-03-18	male	2026-01-02
7	7	Екатерина	Кузнецова	1988-07-30	female	2026-01-09
8	8	Дмитрий	Васильев	1992-11-11	male	2024-01-15
9	9	Анастасия	Попова	1999-04-22	female	2026-01-12
10	10	Михаил	Соколов	1965-08-14	male	2026-01-05

Рис. 12: Результат операций с читателями - часть 1

	l23 id	AZ first_name	AZ last_name	Ø date_of_birth	AZ gender	Ø date_of_registration
1	1	Даниил	Молчанов	2006-10-25	male	2026-01-12
2	2	Иван	Иванов	1950-10-24	male	2026-01-12
3	3	Алексей	Перов	1980-09-21	male	2026-01-12
4	4	Мария	Иванова	2000-05-15	female	2024-01-10
5	5	Ольга	Петрова	1995-12-01	female	2026-01-07
6	6	Сергей	Сидоров	1975-03-18	male	2026-01-02
7	7	Екатерина	Кузнецова	1988-07-30	female	2026-01-09
8	8	Дмитрий	Васильев	1992-11-11	male	2024-01-15
9	9	Анастасия	Попова	1999-04-22	female	2026-01-12
10	10	Михаил	Соколов	1965-08-14	male	2024-01-20

Рис. 13: Результат операций с читателями - часть 2

	123 id	A-Z first_name	A-Z last_name	① date_of_birth	A-Z gender	① date_of_registration
1	4	Мария	Иванова	2000-05-15	female	2024-01-10
2	5	Ольга	Петрова	1995-12-01	female	2026-01-07
3	6	Сергей	Сидоров	1975-03-18	male	2026-01-02
4	7	Екатерина	Кузнецова	1988-07-30	female	2026-01-09
5	8	Дмитрий	Васильев	1992-11-11	male	2024-01-15
6	9	Анастасия	Попова	1999-04-22	female	2026-01-12
7	10	Михаил	Соколов	1965-08-14	male	2024-01-20

Рис. 14: Результат операций с читателями - часть 3

## 4.5 Управление экземплярами книг

```

1  INSERT INTO book_instance (book_information, state, status, location) VALUES
2  (1, 'good', 'in stock', 'Стеллаж 1'),
3  (2, 'excellent', 'reserved', 'Стеллаж 2'),
4  (3, 'good', 'issued', 'Стеллаж 3'),
5  (4, 'good', 'in stock', 'Стеллаж 4'),
6  (5, 'satisfactory', 'in stock', 'Стеллаж 5'),
7  (6, 'excellent', 'in stock', 'Стеллаж 1'),
8  (7, 'good', 'reserved', 'Стеллаж 2'),
9  (8, 'satisfactory', 'issued', 'Стеллаж 3'),
10 (9, 'good', 'in stock', 'Стеллаж 4'),
11 (10, 'excellent', 'in stock', 'Стеллаж 5')
12 RETURNING id;
13
14 UPDATE book_instance SET status='issued' WHERE id=9;
15
16 DELETE FROM book_instance WHERE id IN (4);

```

Листинг 13: CRUD операции для таблицы book\_instance

	123 id	123 book_information	A-Z state	A-Z status	A-Z location
1	1		1 good	in stock	Стеллаж 1
2	2		2 excellent	reserved	Стеллаж 2
3	3		3 good	issued	Стеллаж 3
4	4		4 good	in stock	Стеллаж 4
5	5		5 satisfactory	in stock	Стеллаж 5
6	6		6 excellent	in stock	Стеллаж 1
7	7		7 good	reserved	Стеллаж 2
8	8		8 satisfactory	issued	Стеллаж 3
9	9		9 good	in stock	Стеллаж 4
10	10		10 excellent	in stock	Стеллаж 5

Рис. 15: Результат операций с экземплярами книг - часть 1

	123 id	123 book_information	A-Z state	A-Z status	A-Z location
1	1		1 good	in stock	Стеллаж 1
2		2	2 excellent	reserved	Стеллаж 2
3		3	3 good	issued	Стеллаж 3
4		4	4 good	in stock	Стеллаж 4
5		5	5 satisfactory	in stock	Стеллаж 5
6		6	6 excellent	in stock	Стеллаж 1
7		7	7 good	reserved	Стеллаж 2
8		8	8 satisfactory	issued	Стеллаж 3
9		10	10 excellent	in stock	Стеллаж 5
10		9	9 good	issued	Стеллаж 4

Рис. 16: Результат операций с экземплярами книг - часть 2

	123 id	123 book_information	A-Z state	A-Z status	A-Z location
1	1		1 good	in stock	Стеллаж 1
2		2	2 excellent	reserved	Стеллаж 2
3		3	3 good	issued	Стеллаж 3
4		5	5 satisfactory	in stock	Стеллаж 5
5		6	6 excellent	in stock	Стеллаж 1
6		7	7 good	reserved	Стеллаж 2
7		8	8 satisfactory	issued	Стеллаж 3
8		10	10 excellent	in stock	Стеллаж 5
9		9	9 good	issued	Стеллаж 4

Рис. 17: Результат операций с экземплярами книг - часть 3

## 4.6 Сценарий выдачи книги

```

1 insert into public.issuance (reader_id, book_id, date_time, date_return_expected,
2   date_return_fact)
3 values(2,2,CURRENT_TIMESTAMP(0),CURRENT_DATE+ interval '14 days',null)
4 returning reader_id, book_id;
5
6 update public.book_instance
7 set status='issued'
where id=2;
```

Листинг 14: Процедура выдачи книги

	123 reader_id	123 book_id	⌚ date_time	⌚ date_return_expected	⌚ date_return_fact
1		2	2 6-01-12 04:29:20.000	2026-01-26	[NULL]

Рис. 18: Результат выдачи книги

	123 id	123 book_information	AZ state	AZ status	AZ location
1	1		1 good	in stock	Стеллаж 1
2		4	4 good	in stock	Стеллаж 4
3		5	5 satisfactory	in stock	Стеллаж 5
4		6	6 excellent	in stock	Стеллаж 1
5		7	7 good	reserved	Стеллаж 2
6		8	8 satisfactory	issued	Стеллаж 3
7		9	9 good	in stock	Стеллаж 4
8		10	10 excellent	in stock	Стеллаж 5
9		2	2 excellent	issued	Стеллаж 2
10		3	3 good	issued	Стеллаж 3

Рис. 19: Результат обновления статуса книги

## 4.7 Сценарий возврата книги

```

1 update public.book_instance
2 set status='in stock'
3 where id=2;
4
5 update public.issuance
6 set date_return_fact=CURRENT_DATE
7 where book_id=2 and reader_id=2;
```

Листинг 15: Процедура возврата книги

	123 reader_id	123 book_id	⌚ date_time	⌚ date_return_expected	⌚ date_return_fact
1		2	2 6-01-12 04:33:33.000	2026-01-26	2026-01-12

Рис. 20: Результат возврата книги - обновление статуса

	123 id	123 book_information	AZ state	AZ status	AZ location
1	1		1 good	in stock	Стеллаж 1
2		3	3 good	issued	Стеллаж 3
3		4	4 good	in stock	Стеллаж 4
4		5	5 satisfactory	in stock	Стеллаж 5
5		6	6 excellent	in stock	Стеллаж 1
6		7	7 good	reserved	Стеллаж 2
7		8	8 satisfactory	issued	Стеллаж 3
8		9	9 good	in stock	Стеллаж 4
9		10	10 excellent	in stock	Стеллаж 5
10		2	2 excellent	in stock	Стеллаж 2

Рис. 21: Результат возврата книги - обновление даты возврата

## 4.8 Представление выданных книг

```
1 insert into public.issuance (reader_id, book_id, date_time, date_return_expected,
2   date_return_fact)
3 values
4   (1,2,CURRENT_TIMESTAMP(0) - interval '3 days', CURRENT_DATE + interval '11 days',
5     null),
6   (2,3,CURRENT_TIMESTAMP(0) - interval '8 days', CURRENT_DATE + interval '2 days',
7     null);
8
8 UPDATE public.book_instance
9 set status = 'issued'
10 WHERE id IN (2,3);
```

Листинг 16: Представление issued\_books

	123 reader_id	123 book_id	date_time	date_return_expected	date_return_fact
1		1	2023-01-09 04:46:37.000	2026-01-23	[NULL]
2		2	2023-01-04 04:46:37.000	2026-01-14	[NULL]

Рис. 22: Результат добавления новых выдач

	123 id	123 book_information	A-Z state	A-Z status	A-Z location
1	1		1 good	in stock	Стеллаж 1
2	4		4 good	in stock	Стеллаж 4
3	5		5 satisfactory	in stock	Стеллаж 5
4	6		6 excellent	in stock	Стеллаж 1
5	7		7 good	reserved	Стеллаж 2
6	8		8 satisfactory	issued	Стеллаж 3
7	9		9 good	in stock	Стеллаж 4
8	10		10 excellent	in stock	Стеллаж 5
9	2		2 excellent	issued	Стеллаж 2
10	3		3 good	issued	Стеллаж 3

Рис. 23: Результат обновления статуса экземпляров

## 4.9 Представление просроченных выдач

```
1 CREATE OR REPLACE VIEW public.issued_books_view AS
2 SELECT
3   r.last_name AS reader_last_name,
4   r.first_name AS reader_first_name,
5   a.last_name AS author_last_name,
6   a.first_name AS author_first_name,
```

```

7   b.name_book AS book_title,
8   bi.state AS book_condition,
9   i.date_time AS issue_date
10  FROM public.issuance i
11  JOIN public.reader r ON i.reader_id = r.id
12  JOIN public.book_instance bi ON i.book_id = bi.id
13  JOIN public.book b ON bi.book_information = b.id
14  JOIN public.author a ON b.author = a.id
15  WHERE i.date_return_fact IS NULL;
16
17  SELECT * FROM public.issued_books_view;

```

Листинг 17: Представление overdue\_books

	AZ reader_last_name	AZ reader_first_name	AZ author_last_name	AZ author_first_name	AZ book_title	AZ book_condition	issue_date	Panes
1	Молчанов	Даниил	Пушкин	Александр	Маша и медведь	excellent	2026-01-09 04:46:3:	
2	Иванов	Иван	Булгаков	Михаил	Мастер и Маргарита	good	2026-01-04 04:46:3:	

Рис. 24: Результат представления issued\_books\_view

## 4.10 Модификация выдачи с проверкой просрочек

```

1 CREATE OR REPLACE VIEW public.overdue_books_view AS
2 SELECT
3     r.last_name AS reader_last_name,
4     r.first_name AS reader_first_name,
5     a.last_name AS author_last_name,
6     a.first_name AS author_first_name,
7     b.name_book AS book_title,
8     (CURRENT_DATE - i.date_return_expected) as overdue_days
9 FROM public.issuance i
10 JOIN public.reader r ON i.reader_id = r.id
11 JOIN public.book_instance bi ON i.book_id = bi.id
12 JOIN public.book b ON bi.book_information = b.id
13 JOIN public.author a ON b.author = a.id
14 WHERE i.date_return_fact IS null;
15
16 SELECT * from public.overdue_books_view;

```

Листинг 18: Модифицированная процедура выдачи

	AZ reader_last_name	AZ reader_first_name	AZ author_last_name	AZ author_first_name	AZ book_title	I23 overdue_days	
1	Молчанов	Даниил	Пушкин	Александр	Маша и медведь	-11	
2	Иванов	Иван	Булгаков	Михаил	Мастер и Маргарита	-2	

Рис. 25: Результат представления overdue\_books\_view

## 4.11 Сценарий бронирования книги

```
1 CREATE OR REPLACE FUNCTION reserve_book(
2     p_reader_id INTEGER ,
3     p_book_id INTEGER
4 ) RETURNS BOOLEAN AS $$  
5 DECLARE  
6     v_available_instance_id INTEGER;  
7     v_booking_count INTEGER;  
8     v_reservation_end_date DATE;  
9 BEGIN  
10  
11     SELECT COUNT(*) INTO v_booking_count  
12     FROM public.booking  
13     WHERE number_reader_ticket = p_reader_id  
14     AND data_time + INTERVAL '3 days' > CURRENT_TIMESTAMP;  
15  
16     IF v_booking_count >= 3 THEN  
17         RAISE EXCEPTION 'Превышено максимальное количество активных бронирований';  
18     END IF;  
19  
20     SELECT bi.id INTO v_available_instance_id  
21     FROM public.book_instance bi  
22     WHERE bi.book_information = p_book_id  
23     AND bi.status = 'in stock'  
24     AND bi.state >= 'satisfactory'  
25     LIMIT 1;  
26  
27     IF v_available_instance_id IS NULL THEN  
28         RAISE EXCEPTION 'Нет доступных экземпляров для бронирования';  
29     END IF;  
30  
31     v_reservation_end_date := CURRENT_DATE + INTERVAL '3 days';  
32  
33     INSERT INTO public.booking (  
34         number_reader_ticket ,  
35         book_information ,  
36         data_time ,  
37         min_state  
38     ) VALUES (  
39         p_reader_id ,  
40         p_book_id ,  
41         CURRENT_TIMESTAMP ,  
42         'satisfactory'  
43     );  
44  
45     UPDATE public.book_instance
```

```

46    SET status = 'reserved'
47    WHERE id = v_available_instance_id;
48
49    RETURN TRUE;
50
51 $$ LANGUAGE plpgsql;
52
53 WITH available_books AS (
54     SELECT bi.id AS instance_id, b.id AS book_id, b.name_book
55     FROM book_instance bi
56     JOIN book b ON bi.book_information = b.id
57     WHERE bi.status = 'in stock'
58     LIMIT 3
59 )
60 SELECT * FROM available_books;

```

Листинг 19: Процедура бронирования книги

	123 instance_id	123 book_id	A-Z name_book
1	1	1	Колобок
2		4	Война и мир
3		5	Преступление и наказание

Рис. 26: Результат работы функции reserve\_book

## 4.12 Сценарий отмены бронирования

```

1 CREATE OR REPLACE FUNCTION cancel_reservation(
2     p_reader_id INTEGER,
3     p_book_id INTEGER
4 ) RETURNS BOOLEAN AS $$$
5
6 DECLARE
7     v_booking_id INTEGER;
8     v_instance_id INTEGER;
9
10 BEGIN
11     SELECT b.id INTO v_booking_id
12     FROM public.booking b
13     WHERE b.number_reader_ticket = p_reader_id
14     AND b.book_information = p_book_id
15     AND b.data_time + INTERVAL '3 days' > CURRENT_TIMESTAMP
16     LIMIT 1;
17
18     IF v_booking_id IS NULL THEN

```

```

17      RAISE EXCEPTION 'Активное бронирование не найдено';
18  END IF;

19
20  SELECT bi.id INTO v_instance_id
21  FROM public.book_instance bi
22  WHERE bi.book_information = p_book_id
23  AND bi.status = 'reserved'
24  LIMIT 1;

25
26  DELETE FROM public.booking WHERE id = v_booking_id;
27

28  IF v_instance_id IS NOT NULL THEN
29      UPDATE public.book_instance
30      SET status = 'in stock'
31      WHERE id = v_instance_id;
32  END IF;

33
34  RETURN TRUE;
35
36 $$ LANGUAGE plpgsql;
37
38 BEGIN;
39 SELECT reserve_book(4, 5); -- Создать
40 SELECT cancel_reservation(4, 5); -- Отменить
41 SELECT * FROM booking WHERE number_reader_ticket = 4; -- Проверить (должно быть пусто)
42 SELECT * FROM book_instance WHERE book_information = 5;

```

Листинг 20: Процедура отмены бронирования

	123 ↗ id	123 ↗ number_reader_ticket	123 ↗ book_information	⌚ data_time	A-Z min_state
1	6	4	5	2026-01-12 05:06:58.238	satisfactory

Рис. 27: Результат работы функции cancel\_reservation - часть 1

	123 ↗ id	123 ↗ book_information	A-Z state	A-Z status	A-Z location
1	5	5	satisfactory	reserved	Стеллаж 5

Рис. 28: Результат работы функции cancel\_reservation - часть 2

	123 ↗ id	123 ↗ number_reader_ticket	123 ↗ book_information	⌚ data_time	A-Z min_state

Рис. 29: Результат работы функции cancel\_reservation - часть 3

## 4.13 Модификация выдачи с проверкой бронирований

```
1 CREATE OR REPLACE FUNCTION issue_book_safe(
2     p_reader_id INTEGER ,
3     p_instance_id INTEGER
4 ) RETURNS BOOLEAN AS $$  
5 DECLARE  
6     v_book_id INTEGER ;
7     v_overdue_count INTEGER ;
8     v_active_booking_count INTEGER ;
9     v_book_status availability;  
10 BEGIN  
11     SELECT book_information , status  
12         INTO v_book_id , v_book_status  
13     FROM public.book_instance  
14     WHERE id = p_instance_id;  
15  
16     IF v_book_id IS NULL THEN  
17         RAISE EXCEPTION 'Экземпляр книги не найден';  
18     END IF;  
19  
20     SELECT COUNT(*) INTO v_overdue_count  
21     FROM public.issuance i  
22     WHERE i.reader_id = p_reader_id  
23         AND i.date_return_fact IS NULL  
24         AND i.date_return_expected < CURRENT_DATE;  
25  
26     IF v_overdue_count > 0 THEN  
27         RAISE EXCEPTION 'У читателя есть просроченные книги';  
28     END IF;  
29  
30     SELECT COUNT(*) INTO v_active_booking_count  
31     FROM public.booking b  
32     WHERE b.book_information = v_book_id  
33         AND b.number_reader_ticket != p_reader_id  
34         AND b.data_time + INTERVAL '3 days' > CURRENT_TIMESTAMP;  
35  
36     IF v_active_booking_count > 0 THEN  
37         RAISE EXCEPTION 'Книга забронирована другим читателем';  
38     END IF;  
39  
40     IF v_book_status NOT IN ('in stock' , 'reserved') THEN  
41         RAISE EXCEPTION 'Книга недоступна для выдачи. Текущий статус: %' ,  
42             v_book_status;  
43     END IF;  
44  
        INSERT INTO public.issuance (
```

```

45     reader_id,
46     book_id,
47     date_time,
48     date_return_expected,
49     date_return_fact
50 ) VALUES (
51     p_reader_id,
52     p_instance_id,
53     CURRENT_TIMESTAMP,
54     CURRENT_DATE + INTERVAL '14 days',
55     NULL
56 );
57
58 UPDATE public.book_instance
59 SET status = 'issued'
60 WHERE id = p_instance_id;
61
62 DELETE FROM public.booking
63 WHERE number_reader_ticket = p_reader_id
64 AND book_information = v_book_id;
65
66 RETURN TRUE;
67
68 $$ LANGUAGE plpgsql;
69
70 SELECT issue_book_safe(4, 9);
71
72 SELECT 'Проверка выдачи:' as результат;
73
74 SELECT
75     r.first_name as читатель,
76     b.name_book as книга,
77     i.date_time as дата_выдачи,
78     bi.status as статус_экземпляра
79
80 FROM issuance i
81 JOIN reader r ON i.reader_id = r.id
82 JOIN book_instance bi ON i.book_id = bi.id
83 JOIN book b ON bi.book_information = b.id;

```

Листинг 21: Проверка бронирований при выдаче

	AZ читатель	AZ книга	AZ дата_выдачи	AZ статус_экземпляра
1	Мария	Анна Каренина	2026-01-12 05:48:23.374	issued

Рис. 30: Результат работы функции issue\_book\_safe

## 4.14 Функция поиска местоположений книги

```
1 CREATE OR REPLACE FUNCTION get_book_locations(p_book_id INTEGER)
2 RETURNS TABLE(
3     location VARCHAR,
4     book_state state,
5     instance_count BIGINT,
6     status availability
7 ) AS $$
8 BEGIN
9     RETURN QUERY
10    SELECT
11        bi.location,
12        bi.state,
13        COUNT(*) as instance_count,
14        bi.status
15    FROM public.book_instance bi
16    WHERE bi.book_information = p_book_id
17    GROUP BY bi.location, bi.state, bi.status
18    ORDER BY
19        CASE bi.state
20            WHEN 'excellent' THEN 1
21            WHEN 'good' THEN 2
22            WHEN 'satisfactory' THEN 3
23            WHEN 'dilapidated' THEN 4
24            WHEN 'lost' THEN 5
25            ELSE 6
26        END,
27        bi.location;
28    END;
29 $$ LANGUAGE plpgsql;
30
31 SELECT * FROM get_book_locations(8);
```

Листинг 22: Функция find\_book\_locations

	AZ location	AZ book_state	I23 instance_count	AZ status
1	Стеллаж 7	good	1	in stock

Рис. 31: Результат работы функции get\_book\_locations

## 4.15 Представление доступных экземпляров

```
1 CREATE OR REPLACE VIEW public.available_books_view AS
```

```

2 | SELECT
3 |     b.id as book_id,
4 |     b.name_book,
5 |     a.first_name as author_first_name,
6 |     a.last_name as author_last_name,
7 |     bi.state,
8 |     COUNT(*) as available_count
9 | FROM public.book b
10 | JOIN public.author a ON b.author = a.id
11 | JOIN public.book_instance bi ON bi.book_information = b.id
12 | WHERE bi.status = 'in stock'
13 | GROUP BY b.id, b.name_book, a.first_name, a.last_name, bi.state
14 | ORDER BY b.name_book,
15 | CASE bi.state
16 |     WHEN 'excellent' THEN 1
17 |     WHEN 'good' THEN 2
18 |     WHEN 'satisfactory' THEN 3
19 |     WHEN 'dilapidated' THEN 4
20 |     ELSE 5
21 | END;
22 |
23 | SELECT * FROM public.available_books_view;

```

Листинг 23: Представление available\_instances

	book_id	name_book	author_first_name	author_last_name	state	available_count
1	4	Война и мир	Лев	Толстой	good	1
2	6	Евгений Онегин	Александр	Пушкин	excellent	1
3	9	Лолита	Владимир	Набоков	excellent	1
4	3	Мастер и Маргарита	Михаил	Булгаков	excellent	1
5	3	Мастер и Маргарита	Михаил	Булгаков	good	2
6	7	Мертвые души	Николай	Гоголь	good	1
7	8	Отцы и дети	Иван	Тургенев	good	2
8	5	Преступление и наказание	Федор	Достоевский	good	1

Рис. 32: Результат представления available\_books\_view

## 4.16 Представление долгосрочных выдач

```

1 | CREATE OR REPLACE VIEW public.overdue_books_year_view AS
2 | SELECT
3 |     r.id as reader_id,
4 |     r.last_name AS reader_last_name,
5 |     r.first_name AS reader_first_name,
6 |     a.last_name AS author_last_name,
7 |     a.first_name AS author_first_name,
8 |     b.name_book AS book_title,
9 |     i.date_time AS issue_date,

```

```

10    i.date_return_expected,
11    (CURRENT_DATE - i.date_time::DATE) as days_issued
12  FROM public.issuance i
13  JOIN public.reader r ON i.reader_id = r.id
14  JOIN public.book_instance bi ON i.book_id = bi.id
15  JOIN public.book b ON bi.book_information = b.id
16  JOIN public.author a ON b.author = a.id
17 WHERE i.date_return_fact IS NULL
18 AND i.date_time < CURRENT_DATE - INTERVAL '1 year'
19 ORDER BY i.date_time;
20
21 SELECT * FROM public.overdue_books_year_view;

```

Листинг 24: Представление long\_term\_issuances

	reader_last_name	reader_first_name	author_last_name	author_first_name	book_title	issue_date	date_return_expected	days_issued
1	ва	Мария	Толстой	Лев	Анна Каренина	2025-01-11 06:06:25.369	2025-01-12	366

Рис. 33: Результат представления overdue\_books\_year\_view

## 4.17 Создание таблицы логов

```

1 CREATE TABLE public.logs (
2     id SERIAL PRIMARY KEY,
3     log_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
4     table_name VARCHAR(100) NOT NULL,
5     log_content TEXT NOT NULL
6 );

```

Листинг 25: Создание таблицы logs

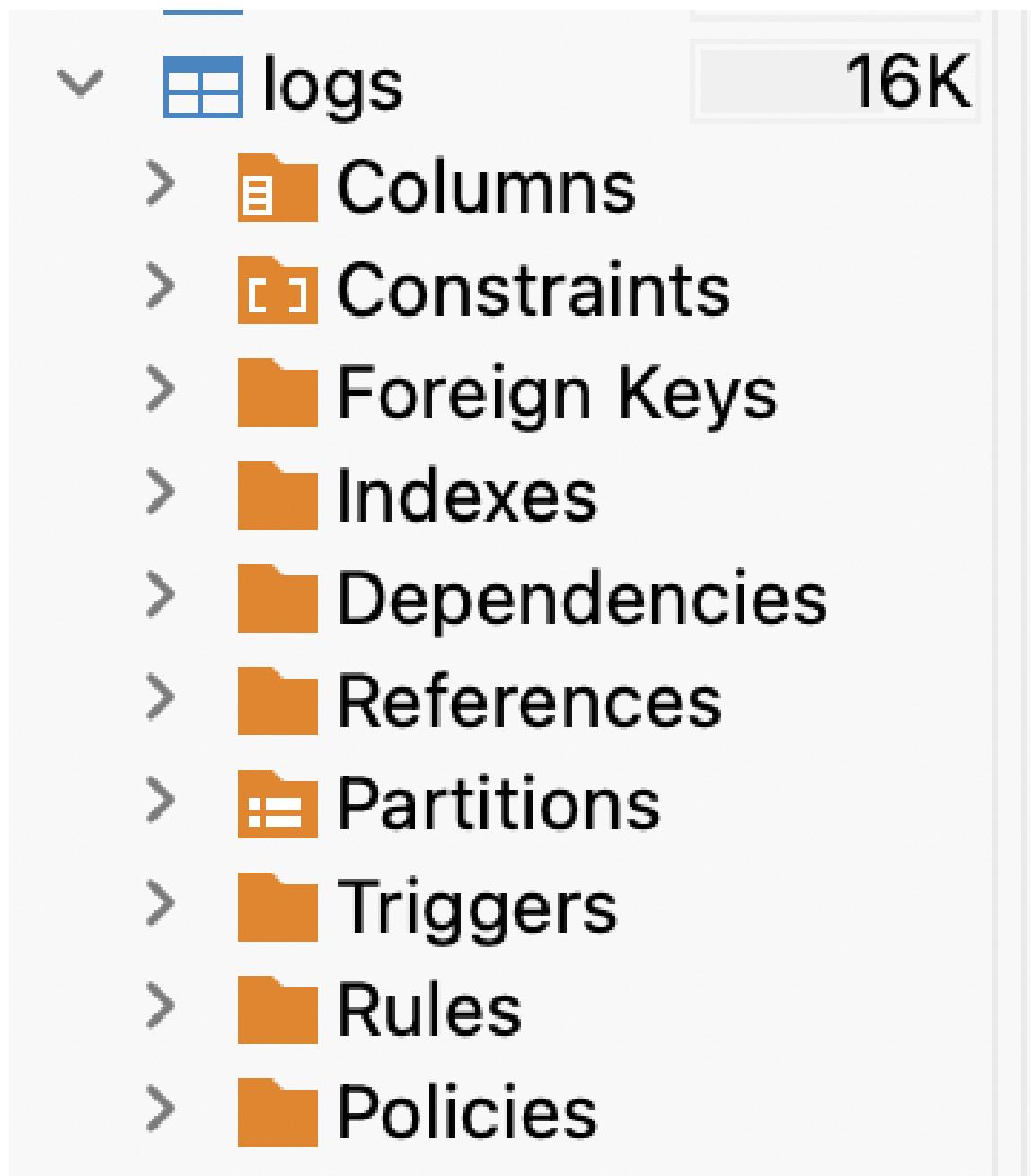


Рис. 34: Результат создания таблицы logs

#### 4.18 Реализация триггеров для логгирования

```
1 CREATE OR REPLACE FUNCTION log_changes()
2 RETURNS TRIGGER AS $$ 
3 DECLARE
4     v_operation TEXT;
5     v_content TEXT;
6 BEGIN
7     IF TG_OP = 'INSERT' THEN
8         v_operation := 'INSERT';
```

```

9      v_content := 'Добавлена запись: ' || row_to_json(NEW);
10     ELSIF TG_OP = 'UPDATE' THEN
11         v_operation := 'UPDATE';
12         v_content := 'Изменена запись. Старые данные: ' || row_to_json(OLD) ||
13                         ' Новые данные: ' || row_to_json(NEW);
14     ELSIF TG_OP = 'DELETE' THEN
15         v_operation := 'DELETE';
16         v_content := 'Удалена запись: ' || row_to_json(OLD);
17     END IF;
18
19     INSERT INTO public.logs (table_name, log_content)
20     VALUES (TG_TABLE_NAME, v_operation || ' - ' || v_content);
21
22     IF TG_OP = 'DELETE' THEN
23         RETURN OLD;
24     ELSE
25         RETURN NEW;
26     END IF;
27 END;
28 $$ LANGUAGE plpgsql;
29
30 CREATE TRIGGER trg_log_author
31     AFTER INSERT OR UPDATE OR DELETE ON public.author
32     FOR EACH ROW EXECUTE FUNCTION log_changes();
33
34 CREATE TRIGGER trg_log_book
35     AFTER INSERT OR UPDATE OR DELETE ON public.book
36     FOR EACH ROW EXECUTE FUNCTION log_changes();
37
38 CREATE TRIGGER trg_log_reader
39     AFTER INSERT OR UPDATE OR DELETE ON public.reader
40     FOR EACH ROW EXECUTE FUNCTION log_changes();
41
42 CREATE TRIGGER trg_log_book_instance
43     AFTER INSERT OR UPDATE OR DELETE ON public.book_instance
44     FOR EACH ROW EXECUTE FUNCTION log_changes();
45
46 CREATE TRIGGER trg_log_issuance
47     AFTER INSERT OR UPDATE OR DELETE ON public.issuance
48     FOR EACH ROW EXECUTE FUNCTION log_changes();
49
50 CREATE TRIGGER trg_log_booking
51     AFTER INSERT OR UPDATE OR DELETE ON public.booking
52     FOR EACH ROW EXECUTE FUNCTION log_changes();
53
54 -- Бронирование книги
55 SELECT reserve_book(4, 9);

```

```

56
57 -- Отмена бронирования
58 SELECT cancel_reservation(4, 1);
59
60 -- Выдача книги
61 SELECT issue_book(4, 1);
62
63 -- Получение информации о местоположениях книги
64 SELECT * FROM get_book_locations(8);
65
66 -- Просмотр доступных книг
67 SELECT * FROM available_books_view;
68
69 -- Просмотр долгосрочных просрочек
70 SELECT * FROM overdue_books_year_view;
71
72 -- Просмотр логов
73 SELECT * FROM logs ORDER BY log_date DESC;

```

Листинг 26: Триггеры для автоматического логгирования

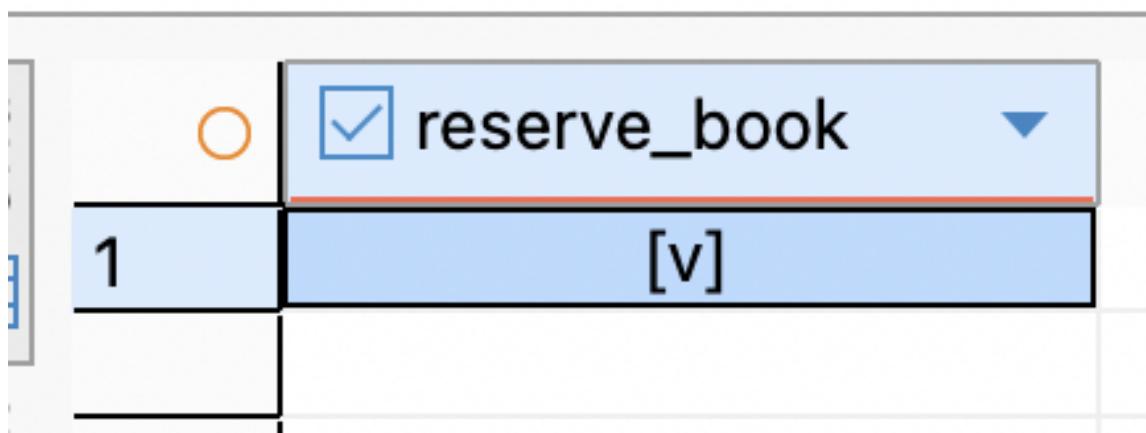


Рис. 35: Результат создания триггеров - часть 1

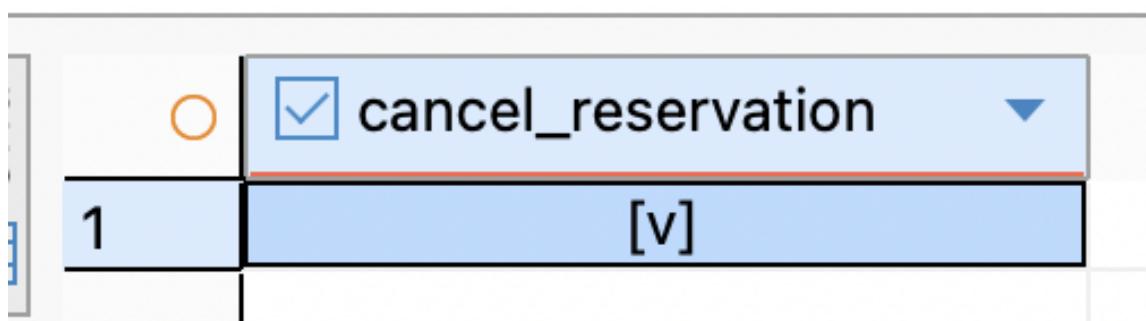


Рис. 36: Результат создания триггеров - часть 2

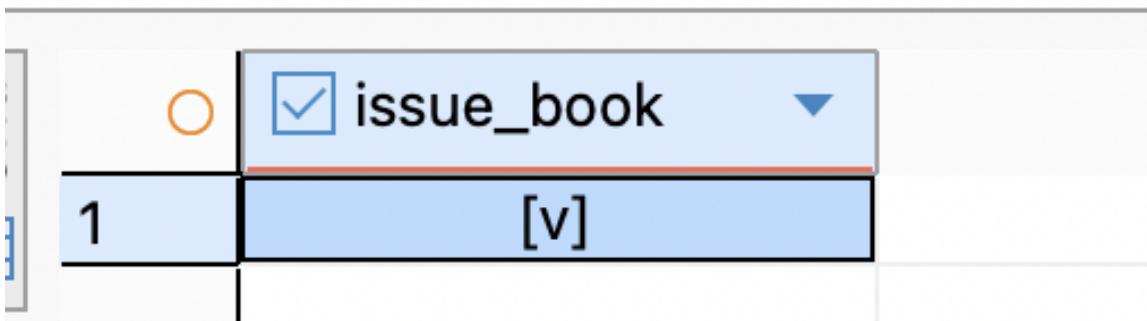


Рис. 37: Результат создания триггеров - часть 3

	A-Z location	A-Z book_state	123 instance_count	A-Z status
1	Стеллаж 3	satisfactory	1	issued

Рис. 38: Результат создания триггеров - часть 4

	123 book_id	AZ name_book	AZ author_first_name	AZ author_last_name	AZ state	123 available_count
1	10	Анна Каренина	Лев	Толстой	excellent	1
2	4	Война и мир	Лев	Толстой	good	1
3	6	Евгений Онегин	Александр	Пушкин	excellent	1
4	9	Лолита	Владимир	Набоков	good	1
5	5	Преступление и наказание	Федор	Достоевский	satisfactory	1

Рис. 39: Результат создания триггеров - часть 5

	I23 reader_id	AZ reader_last_name	AZ reader_first_name	AZ author_last_name	AZ author_first_name	AZ book_title	O issue_d
1	4	Иванова	Мария	Набоков	Владимир	Лолита	2025-01-11

Рис. 40: Результат создания триггеров - часть 6

	123 id	⌚ log_date	⤵ AZ table_name	⤵ AZ log_content
1	5	2026-01-12 06:45:30.267	issuance	INSERT - Добавлена запись: {"reader_id":4,"book_id":1,"date_time":"2026-01-12T06:45:30.267024","data_type":1}
2	6	2026-01-12 06:45:30.267	book_instance	UPDATE - Изменена запись. Старые данные: {"id":1,"book_information":1,"state":"good","status":"in store"}; Новые данные: {"id":1,"book_information":1,"state":"good","status":"in store"}
3	3	2026-01-12 06:45:21.049	booking	DELETE - Удалена запись: {"id":1,"number_reader_ticket":8,"book_information":5,"data_time":"2026-01-12T06:45:21.049000"}
4	4	2026-01-12 06:45:21.049	book_instance	UPDATE - Изменена запись. Старые данные: {"id":5,"book_information":5,"state":"satisfactory","status":1}; Новые данные: {"id":5,"book_information":5,"state":"satisfactory","status":1}
5	1	2026-01-12 06:44:55.645	booking	INSERT - Добавлена запись: {"id":1,"number_reader_ticket":8,"book_information":5,"data_time":"2026-01-12T06:44:55.645000"}
6	2	2026-01-12 06:44:55.645	book_instance	UPDATE - Изменена запись. Старые данные: {"id":5,"book_information":5,"state":"satisfactory","status":1}; Новые данные: {"id":5,"book_information":5,"state":"satisfactory","status":1}

Рис. 41: Результат создания триггеров - часть 7

## 5 Заключение

В результате выполнения курсовой работы была создана полнофункциональная система управления библиотекой, демонстрирующая практическое применение знаний

в области проектирования баз данных, SQL-программирования и разработки бизнес-логики. Система соответствует современным требованиям к информационным системам и может служить основой для дальнейшего развития и внедрения в реальные библиотечные процессы. Работа подтвердила важность правильного проектирования структуры базы данных и эффективного использования возможностей СУБД для создания надежных и производительных информационных систем.

## 6 Список использованных источников

1. Грофф Дж., Вайнберг П. SQL: полное руководство. – 3-е изд. – М.: Вильямс, 2019. – 960 с.
2. PostgreSQL: мощь открытых СУБД / Под ред. О. В. Самойленко. – СПб.: БХВ-Петербург, 2020. – 576 с.
3. Таненбаум Э., ван Стейн М. Распределенные системы. Принципы и парадигмы. – 2-е изд. – СПб.: Питер, 2021. – 1104 с.
4. Базы данных: проектирование, реализация, сопровождение. Теория и практика / К. Дж. Дейт. – М.: Вильямс, 2022. – 1440 с.
5. SQL и реляционная теория. Как грамотно писать код на SQL / К. Дж. Дейт. – М.: Символ-Плюс, 2018. – 480 с.