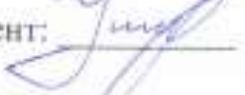



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

КУРСОВОЙ ПРОЕКТ

По курсу **Проектирование и администрирование баз данных**
по направлению 09.03.01 – Информатика и вычислительная техника
Образовательная программа (профиль) «Системная и программная
инженерия»

ТЕМА
«ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ
«БИБЛИОТЕКА» В СУБД POSTGRESQL»

Студент:  / Ушаков Матвей Викторович, 221-329/
ФИО, группа

Проверил:  к.т.н., доцент Евдошенко О.И.

Москва, 2023

Федеральное государственное автономное
образовательное учреждение высшего образования
«Московский политехнический университет»

Кафедра: инфокогнитивных технологий

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Дисциплина: Проектирование и администрирование баз данных

Тема: Проектирование и разработка базы данных «Библиотека» в СУБД PostgreSQL.

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

1. Необходимо выбрать и проанализировать предметную область.
2. Определить сущности (>5), атрибуты (>3), связи между сущностями. Привести отношения к 3 НФ.
3. Разработать базу данных средствами СУБД на основе спроектированной модели. Установить связи для поддержки ссылочной целостности.
4. Разработать SQL-запросы (DML) (не менее 20) различной степени сложности (агрегатные функции, группировка (GROUP BY, HAVING), подзапросы, CASE, оконные функции)
5. Разработать функции для расчетов (3 функции, используя язык SQL и plpgsql)
6. Разработать триггеры (5 триггеров)

Исполнитель: студент гр. 221-329 Ушаков М.В.

Руководитель: к.т.н., доцент кафедры ИКТ Евдошенко О.И.


ПОДПИСЬ

ПОДПИСЬ

СОДЕРЖАНИЕ

| | |
|---|----|
| Введение | 4 |
| Актуальность проблемы | 4 |
| Цель работы: | 4 |
| Задачи работы: | 4 |
| 1 Описание предметной области | 6 |
| 2 Проектирование и разработка данных | 7 |
| 2.1 Концептуальное проектирование базы данных | 7 |
| 2.2 Логическое проектирование базы данных | 9 |
| 2.3 Выбор и описание СУБД | 10 |
| 2.4 Физическое проектирование базы данных | 11 |
| 3 Использование базы данных | 24 |
| 3.1 Создание DataSet | 24 |
| 3.2 Добавление данных в базу | 28 |
| 3.3 Запросы к базе данных | 36 |
| 3.4 Тестирование функций и триггеров | 47 |
| Заключение | 58 |
| Список использованных источников | 59 |

ВВЕДЕНИЕ

Актуальность проблемы

Модернизация структуры базы данных становится неотъемлемой частью деятельности для всех современных компаний. Она обусловлена следующими факторами:

- 1) стремительно развивающиеся технологии выдвигают требования по обновлению и адаптации баз данных к новым возможностям и объемам информации;
- 2) изменения в бизнес-процессах подразумевают необходимость согласованной структуры баз данных для эффективного управления информационными потоками;
- 3) адаптация к изменяющемуся законодательству требует своевременной модернизации баз данных для соответствия обязательствам по хранению и обработке пользовательских данных.

Учет всех этих факторов позволяет компаниям поддерживать гибкость, эффективность и гарантировать защиту интересов клиентов, делая модернизацию баз данных важной составляющей успешной бизнес-стратегии.

Курсовая работа выполнена в рамках учебного предмета «Проектирование и администрирование баз данных» и нацелена на закрепление навыков по разработке с использованием СУБД PostgreSQL.

Цель работы:

Систематизация и закрепление полученных теоретических и практических умений по разработке баз данных с использованием СУБД PostgreSQL.

Задачи работы:

- 1) изучить литературу по проектированию и разработке базы данных в СУБД PostgreSQL;

- 2) получить практический опыт разработки базы данных в СУБД PostgreSQL.

1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Вы являетесь руководителем библиотеки. Ваша библиотека решила зарабатывать деньги, выдавая напрокат некоторые книги, имеющиеся в небольшом количестве экземпляров. Вашей задачей является отслеживание финансовых показателей работы. У каждой книги, выдаваемой в прокат, есть название, автор, жанр. В зависимости от ценности книги вы определили для каждой из них залоговую стоимость (сумма, вносимая клиентом при взятии книги напрокат) и стоимость проката (сумма, которую клиент платит при возврате книги, получая назад залог). В библиотеку обращаются читатели. Все читатели регистрируются в картотеке, которая содержит стандартные анкетные данные (фамилия, имя, отчество, адрес, телефон). Каждый читатель может обращаться в библиотеку несколько раз. Все обращения читателей фиксируются, при этом по каждому факту выдачи книги запоминаются дата выдачи и ожидаемая дата возврата.

Несложный анализ показал, что стоимость проката книги должна зависеть не только от самой книги, но и от срока ее проката. Кроме того, необходимо добавить систему штрафов за вред, нанесенный книге, и систему скидок для некоторых категорий читателей.

При изучении предметной области было выявлено 5 бизнес-процессов. Подробнее с ними можно ознакомиться в «Таблице 1 – Список основных бизнес-процессов».

Таблица 1 – Список основных бизнес-процессов

| Номер бизнес-процесса | Наименование бизнес процесса |
|-----------------------|--|
| 1 | Добавление новой книги в прокат |
| 2 | Изъятие книги из проката |
| 3 | Регистрация пользователя в картотеке |
| 4 | Выдача книги в прокат клиенту |
| 5 | Получение книги после проката от клиента |

2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ДАННЫХ

2.1 Концептуальное проектирование базы данных

При анализе предметной области были выявлены следующие сущности: «книга», «прокат», «клиент».

Сущность «книга» содержит следующие атрибуты:

- 1) автор;
- 2) жанр книги (может быть несколько);
- 3) название;
- 4) размер депозита, вносимого клиентом при взятии в прокат;
- 5) стоимость аренды книги за день проката;
- 6) комментарий о состоянии книги;
- 7) статус доступности для проката.

Сущность «прокат» содержит следующие атрибуты:

- 1) длительность проката;
- 2) стоимость проката;
- 3) штраф за нанесенный книге ущерб;
- 4) книга, взятая в прокат;
- 5) клиент, воспользовавшийся услугой.

Сущность «клиент» содержит следующие атрибуты:

- 1) ФИО клиента;
- 2) категория клиента (школьник, студент, пенсионер, стандарт);
- 3) размер скидки в зависимости от категории;
- 4) контакты клиента;
- 5) адрес проживания.

Связи между сущностями отображены в «Рисунке 1 – Концептуальная модель данных».

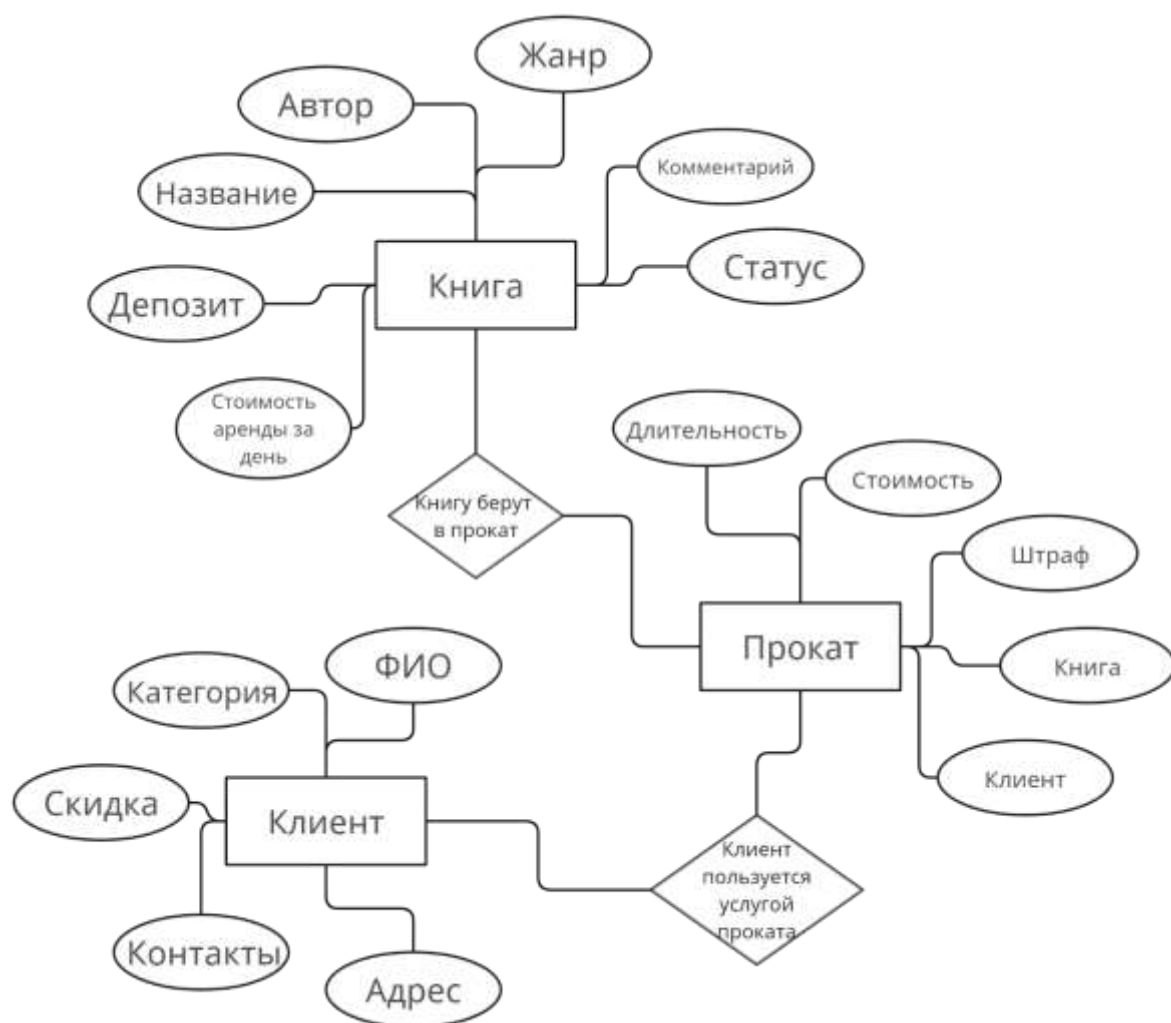


Рисунок 1 – Концептуальная модель данных

Структуры сущностей приведены в последующих таблицах: «Таблица 2 – Структура сущности книга», «Таблица 3 – Структура сущности прокат», «Таблица 4 – Структура сущности клиент».

Таблица 2 – Структура сущности книга

| Название параметра | Тип данных | Размер | Диапазон значений |
|-------------------------------|------------|--------|--|
| Название | Текстовый | 100 | |
| Автор | Текстовый | 100 | |
| Жанр | Текстовый | 100 | |
| Размер депозита | Числовой | | От 0 |
| Стоимость аренды за день | Числовой | | От 0 |
| Комментарий о состоянии книги | Текстовый | | |
| Статус доступности | Логический | 1 | TRUE – книга в наличии и доступна для проката. FALSE – книга уже взята в прокат |

Таблица 3 – Структура сущности прокат

| Название параметра | Тип данных | Размер | Диапазон значений |
|--------------------|------------|--------|-------------------|
| Длительность | Дата | 8 | ДД-ММ-ГГГГ |
| Стоимость | Числовой | | От 0 |
| Штраф | Числовой | | От 0 |
| Книга | Текстовый | | |
| Клиент | Текстовый | | |

Таблица 4 – Структура сущности клиент

| Название параметра | Тип данных | Размер | Диапазон значений |
|--------------------|------------|--------|---------------------------------------|
| ФИО | Текстовый | 100 | |
| Категория | Текстовый | 100 | |
| Размер скидки | Числовой | | От 0 до 1 |
| Контакты | Текстовый | | Подходит под маску +X (XXX) XXX-XX-XX |
| Адрес проживания | Текстовый | | |

2.2 Логическое проектирование базы данных

Для выявленных сущностей была создана логическая схема базы данных (Рисунок 2 – Логическая схема базы данных).

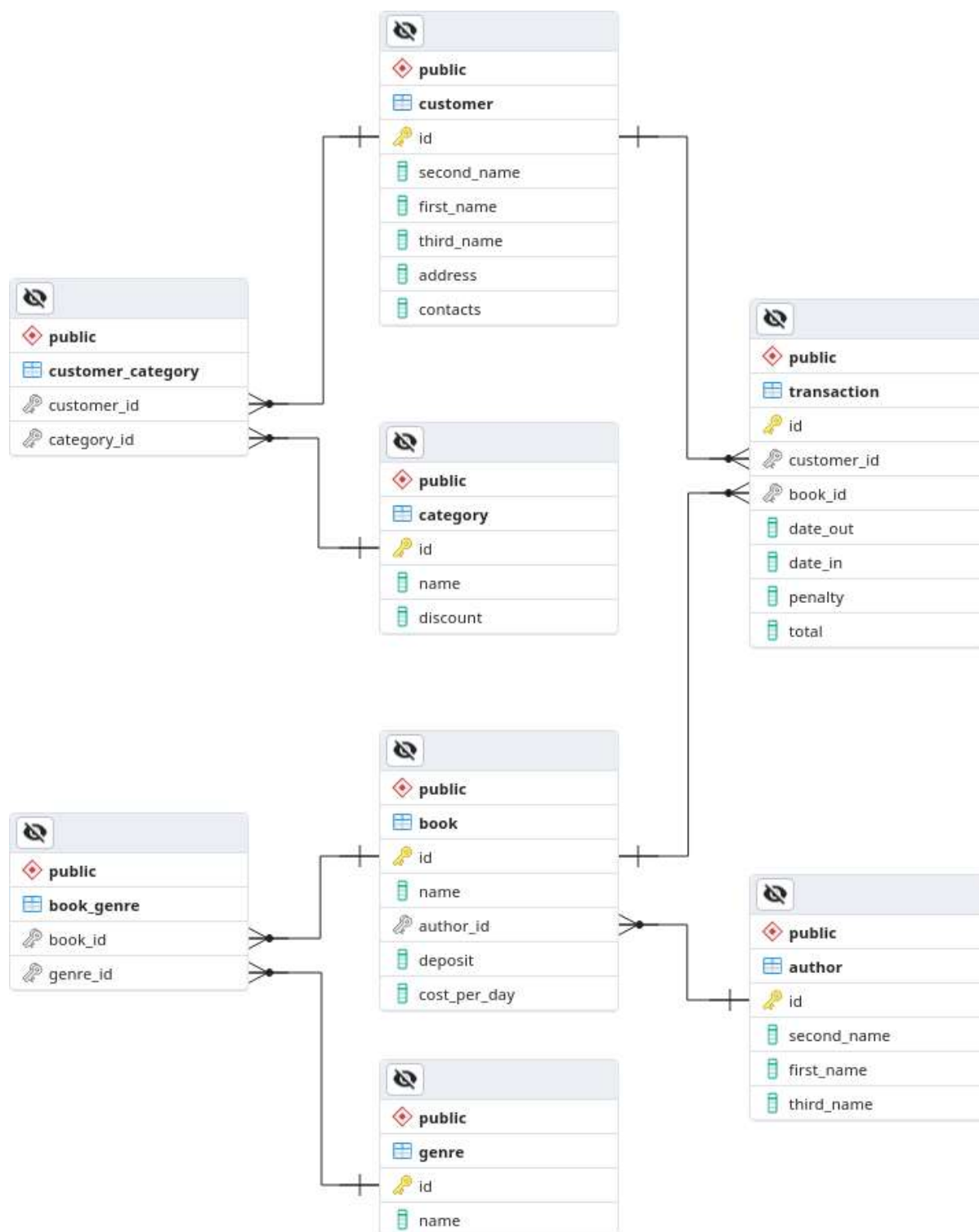


Рисунок 2 – Логическая модель данных

2.3 Выбор и описание СУБД

Для реализации курсовой работы по базам данных была выбрана СУБД PostgreSQL.

PostgreSQL является СУБД с открытым исходным кодом, что позволяет как можно глубже изучить внутреннее устройство СУБД. PostgreSQL также обладает широким набором функций, обеспечивающих гибкость при разработке баз данных, поддерживает различные типы данных. Надежность и стабильность PostgreSQL также являются важными преимуществами. СУБД активно поддерживается и разрабатывается сообществом разработчиков, что гарантирует безопасность и исправление ошибок. Встроенные механизмы резервного копирования, восстановления и системы журналирования обеспечивают сохранность данных и доступность базы данных. Кроме того, PostgreSQL обладает высокой производительностью и масштабируемостью. Он способен эффективно обрабатывать большие объемы данных и поддерживает параллельную обработку запросов, оптимизацию запросов и кэширование. Наконец, PostgreSQL имеет широкое сообщество разработчиков и пользователей, а также обширную документацию. Это облегчает изучение и работу с PostgreSQL, а также предоставляет возможность получить поддержку и советы от опытных пользователей. Все эти причины делают PostgreSQL отличным выбором для реализации курсовой работы.

2.4 Физическое проектирование базы данных

Используя функционал PostgreSQL была создана физическая схема базы данных (Рисунок 3 – Физическая модель данных).

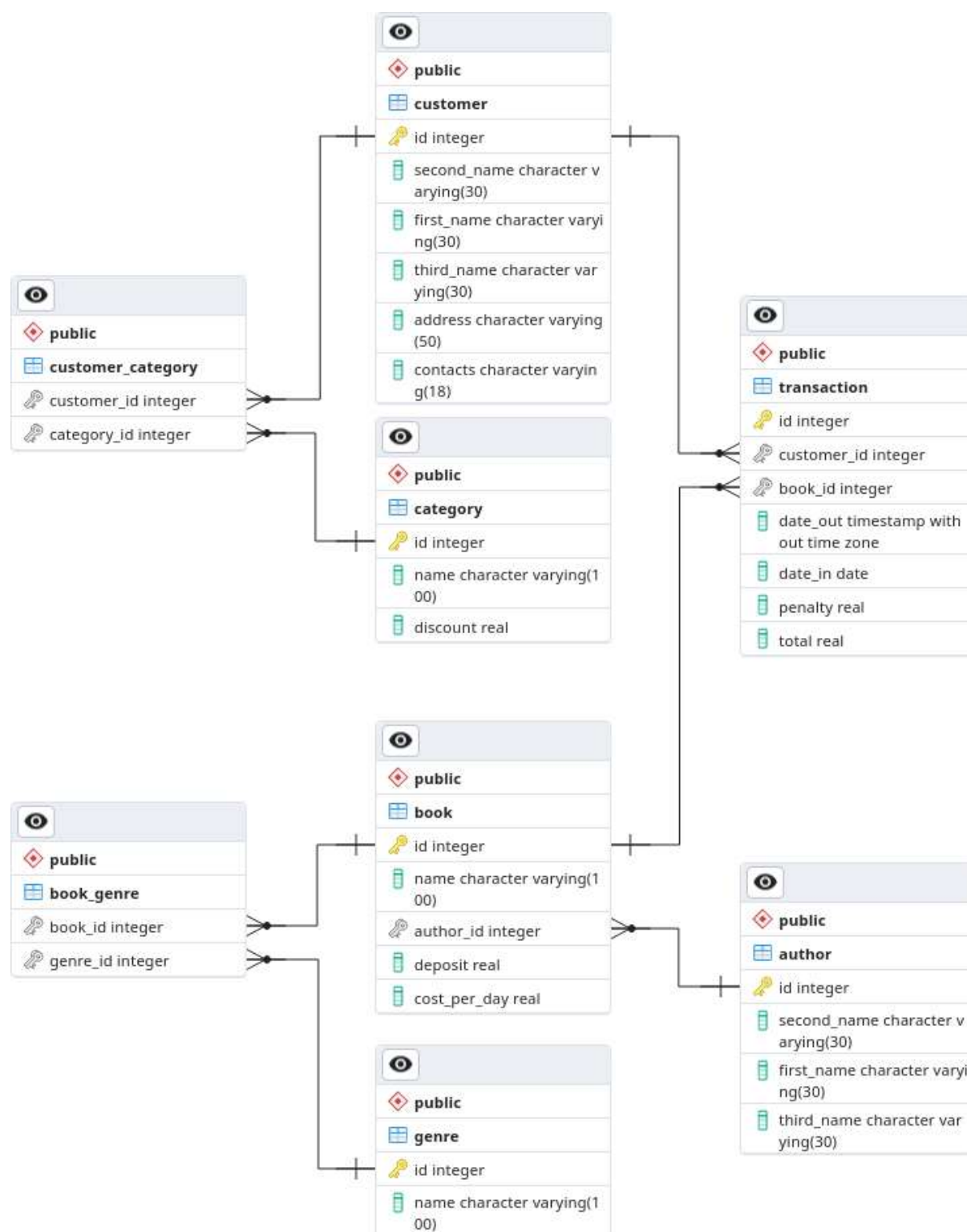


Рисунок 3 – Физическая модель данных

Далее модель была реализована при помощи PostgreSQL. Для создания таблиц и связей между ними были использованы SQL-запросы, которые указаны в «Листинге 1 – Инициализация базы данных».

Листинг 1 – Инициализация базы данных

```
-----  
-- Table author -----  
  
DROP TABLE IF EXISTS author CASCADE;  
CREATE TABLE IF NOT EXISTS author  
(  
    id                serial PRIMARY KEY,  
    second_name       varchar (30) NOT NULL,  
    first_name        varchar (30) NOT NULL,  
    third_name        varchar (30) DEFAULT (NULL)  
);  
  
-----  
-- Table book -----  
  
DROP TABLE IF EXISTS book CASCADE;  
CREATE TABLE IF NOT EXISTS book  
(  
    id                serial PRIMARY KEY,  
    name              varchar (100) NOT NULL,  
    author_id         integer REFERENCES author (id) NOT NULL,  
    deposit           real NOT NULL,  
    cost_per_day      real NOT NULL,  
    comment            text DEFAULT ('Новая книга'),  
    available         boolean DEFAULT (TRUE)  
);
```

```

-----
-- Table genre -----

DROP TABLE IF EXISTS genre CASCADE;
CREATE TABLE IF NOT EXISTS genre
(
    id          serial PRIMARY KEY,
    name        varchar(100) NOT NULL
);

-----

-- Table book_genre -----

DROP TABLE IF EXISTS book_genre CASCADE;
CREATE TABLE IF NOT EXISTS book_genre
(
    book_id      serial REFERENCES book (id) NOT NULL,
    genre_id     integer REFERENCES genre (id) NOT NULL
);

-----

-- Table customer -----

DROP TABLE IF EXISTS customer CASCADE;
CREATE TABLE IF NOT EXISTS customer
(
    id serial PRIMARY KEY,
    second_name  varchar(30) NOT NULL,
    first_name   varchar(30) NOT NULL,
    third_name   varchar(30) DEFAULT (''),
    address      varchar(50) NOT NULL,
    contacts     varchar(18) NOT NULL
);

```

```

-----
-- Table category -----

DROP TABLE IF EXISTS category CASCADE;
CREATE TABLE IF NOT EXISTS category
(
    id            serial PRIMARY KEY,
    name          varchar(100) NOT NULL,
    discount      real NOT NULL
);

-----
-- Table customer_category -----

DROP TABLE IF EXISTS customer_category CASCADE;
CREATE TABLE IF NOT EXISTS customer_category
(
    customer_id   integer REFERENCES customer (id) NOT NULL,
    category_id   integer REFERENCES category (id) NOT NULL
);

```

```

-----
-- Table transaction -----

DROP TABLE IF EXISTS transaction CASCADE;
CREATE TABLE IF NOT EXISTS transaction
(
    id                serial PRIMARY KEY,
    customer_id       integer REFERENCES customer (id) NOT NULL,
    book_id           integer REFERENCES book (id) NOT NULL,
    date_out          date DEFAULT (CURRENT_DATE),
    date_in           date DEFAULT (NULL),
    penalty           real DEFAULT (0),
    total             real DEFAULT (NULL)
);

```

Для работы с базой данных были разработаны следующие функции и триггеры:

1) функция open_transaction (customer_contacts character varying (18), book_name character varying(100)), которая используется для создания новой записи о выдаче книги в прокат. Функция автоматически выставляет время открытия транзакции на текущую дату. Код функции и ее инициализация указаны в «Листинге 2 – Функция open_transaction»;

Листинг 2 – Функция open_transaction

```

CREATE OR REPLACE PROCEDURE open_transaction
(
    customer_contacts    character varying(18),
    book_name            character varying(100)
)
AS $$
BEGIN
    INSERT INTO transaction
    (customer_id, book_id)
VALUES

```



```

        ((SELECT id FROM customer WHERE contacts =
contacts_formatter(customer_contacts))),
        (SELECT id FROM book WHERE name = book_name));
END;
$$ LANGUAGE PLPGSQL;

```

2) функция `close_transaction (customer_contacts character varying(18), book_name character varying(100))` и `close_transaction (customer_contacts character varying(18), book_name character varying(100), customer_penalty real, new_comment text)`, которая используется для закрытия транзакции в ситуации, когда клиент вернул книгу из проката. Функция автоматически подсчитывает дату закрытия транзакции и заносит эти данные в базу данных. Использована перегрузка функции, чтобы предусмотреть варианты возвращения книги в изначальном состоянии, или при порче книги. Код функции и ее инициализация указаны в «Листинге 3 – Функция `close_transaction`»;

Листинг 3 – Функция `close_transaction`

```

-- Если нет штрафа за состояние книги
CREATE OR REPLACE PROCEDURE close_transaction
(
    customer_contacts character varying(18),
    book_name character varying(100)
)
AS $$
BEGIN
    UPDATE transaction SET date_in = CURRENT_DATE
    WHERE date_in IS NULL AND
        customer_id = (SELECT id FROM customer
                        WHERE contacts =
contacts_formatter(customer_contacts)) AND
        book_id = (SELECT id FROM book WHERE name = book_name);
END;
$$ LANGUAGE plpgsql;

```

```

-- если есть штраф за состояние книги
CREATE OR REPLACE PROCEDURE close_transaction
(
    customer_contacts character varying(18),
    book_name character varying(100),
    customer_penalty real,
    new_comment text
)
AS $$
BEGIN
    UPDATE transaction SET date_in = CURRENT_DATE, penalty =
customer_penalty
    WHERE date_in IS NULL AND
    customer_id = (SELECT id FROM customer
                    WHERE contacts =
contacts_formatter(customer_contacts)) AND
    book_id = (SELECT id FROM book WHERE name = book_name);
    UPDATE book SET comment = new_comment WHERE name =
book_name;
END;
$$ LANGUAGE plpgsql;

```

3) функция `contacts_formatter (contacts character varying(18))`, которая используется другими функциями и триггерами для приведения вводимого номера телефона к необходимому формату для хранения в базе данных. Функция преобразует `XXXXXXXXXXXX` в `+X (XXX) XXX-XX-XX`. Код функции и ее инициализация указаны в «Листинге 4 – Функция `contacts_formatter`»;

Листинг 4 – Функция `contacts_formatter`

```

CREATE OR REPLACE FUNCTION contacts_formatter(contacts character
varying(18))
RETURNS character varying(18)
AS $$
DECLARE
    formatted_contacts character varying(18);

```

```
BEGIN
```

```
    formatted_contacts = CONCAT(  
        '+', substr (contacts, 1, 1),  
        ' (' , substr (contacts, 2, 3), ') ',  
        substr (contacts, 5, 3), '-',  
        substr (contacts, 8, 2), '-',  
        substr( contacts, 10, 2));  
    RETURN formatted_contacts;  
END;  
$$ LANGUAGE plpgsql;
```

4) функция `interval_counter (date_o date, date_i date)`, которая используется другими функциями и триггерами для подсчета длительности проката книги в днях. Код функции и ее инициализация указаны в «Листинге 5 – Функция `interval_counter`»;

Листинг 5 – Функция `interval_counter`

```
CREATE OR REPLACE FUNCTION interval_counter(date_o date, date_i  
date)  
RETURNS integer  
AS $$  
    DECLARE  
        days integer;  
    BEGIN  
        days = (EXTRACT(EPOCH FROM AGE(date_i,  
date_o))/60/60/24)::integer;  
        RETURN days;  
    END  
$$ LANGUAGE plpgsql;
```

5) триггер `auto_category`, который автоматически создает запись о категории нового клиента при внесении новой записи в таблицу `customer`. Код триггера и его инициализация указаны в «Листинге 6 – Триггер `auto_category`»;

Листинг 6 – Триггер auto_category

```
CREATE OR REPLACE FUNCTION auto_category()  
RETURNS TRIGGER  
AS $$  
    BEGIN  
        INSERT INTO customer_category  
        VALUES (NEW.id, 4);  
        RETURN NEW;  
    END;  
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER auto_category  
AFTER INSERT ON customer  
FOR EACH ROW EXECUTE FUNCTION auto_category();
```

б) триггер auto_contacts_formatter, который автоматически применяет функцию contacts_formatter(contacts character varying(18)) к значению атрибута contacts в новой записи в таблице customer. Код триггера и его инициализация указаны в «Листинге 7 – Триггер auto_contacts_formatter»;

Листинг 7 – Триггер auto_contacts_formatter

```
CREATE OR REPLACE FUNCTION auto_contacts_formatter()  
RETURNS TRIGGER  
AS $$  
    BEGIN  
        NEW.contacts = contacts_formatter(NEW.contacts);  
        RETURN NEW;  
    END;  
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER auto_contacts_formatter  
BEFORE INSERT ON customer  
FOR EACH ROW  
EXECUTE FUNCTION auto_contacts_formatter();
```

7) триггер `auto_unique_contacts_checker`, который проверяет при создании новой записи в таблицу `customer` уникальность данных, внесенных в атрибут `contacts`. Если такой номер телефона уже зарегистрирован, то в регистрации клиенту будет отказано. Код триггера и его инициализация указаны в «Листинге 8 – Триггер `auto_unique_contacts_checker`»;

Листинг 8 – Триггер `auto_unique_contacts_checker`

```
CREATE OR REPLACE FUNCTION unique_contacts_checker()
RETURNS TRIGGER
AS $$
BEGIN
    IF NEW.contacts IN (SELECT contacts FROM customer) THEN
        RAISE EXCEPTION 'Уже есть клиент с таким номером
телефона!';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER auto_unique_contacts_checker
BEFORE INSERT OR UPDATE ON customer
FOR EACH ROW
EXECUTE FUNCTION unique_contacts_checker();
```

8) триггер `book_status_changer`, который при создании или обновлении записи в таблице `transaction` меняет значения атрибута `available` на противоположный, чтобы обеспечить согласованность данных в базе данных автоматически. Код триггера и его инициализация указаны в «Листинге 9 – Триггер `book_status_changer`»;

Листинг 9 – Триггер `book_status_changer`

```
CREATE OR REPLACE FUNCTION book_status_changer()
RETURNS TRIGGER
AS $$
BEGIN
    IF NEW.date_in IS NOT NULL THEN
```

```

        UPDATE book SET available = TRUE WHERE id =
NEW.book_id;
    ELSE
        UPDATE book SET available = FALSE WHERE id =
NEW.book_id;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
CREATE OR REPLACE TRIGGER book_status_changer
BEFORE INSERT OR UPDATE ON transaction
FOR EACH ROW
EXECUTE FUNCTION book_status_changer();

```

9) триггер `total_counter`, который при обновлении записи в таблице `transaction` автоматически пересчитывает стоимость транзакции, записывает ее в базу данных и выводит сумму в сообщении для пользователя базы данных. Код триггера и его инициализация указаны в «Листинге 10 – Триггер `total_counter`»;

Листинг 10 – Триггер `total_counter`

```

CREATE OR REPLACE FUNCTION total_counter()
RETURNS TRIGGER
AS $$
    DECLARE
        discount_size real;
    BEGIN
        discount_size := 1 - (SELECT(discount/100)
                                FROM customer, customer_category, category
                                WHERE customer.id = NEW.customer_id AND
                                customer.id = customer_category.customer_id AND
                                category.id = customer_category.category_id);
        IF NEW.date_in IS NOT NULL THEN
            NEW.total = interval_counter(NEW.date_out, NEW.date_in)
            *(SELECT cost_per_day from book where id = NEW.book_id);

```

```

        NEW.total = FLOOR(NEW.total*discount_size) +
NEW.penalty;
        RAISE NOTICE 'Итоговая стоимость составит: % рублей',
NEW.total;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE OR REPLACE TRIGGER total_counter
BEFORE UPDATE ON transaction
FOR EACH ROW
EXECUTE FUNCTION total_counter();

```

Итого было создано 8 таблиц, 5 функций, 5 триггеров.

3 ИСПОЛЬЗОВАНИЕ БАЗЫ ДАННЫХ

3.1 Создание DataSet

Для дальнейшей работы с базой данных был разработан dataset, указанный в «Таблице 5 – Dataset таблицы author», «Таблице 6 – Dataset таблицы book», «Таблице 7 – Dataset таблицы book_genre», «Таблице 8 – Dataset таблицы category», «Таблице 9 – Dataset таблицы customer_category», «Таблице 10 – Dataset таблицы customer», «Таблице 11 – Dataset таблицы genre», «Таблице 12 – Dataset таблицы transaction».

Таблица 5 – Dataset таблицы author

| id (идентификатор автора в базе данных) | second_name (фамилия) | first_name (имя) | third_name (отчество) |
|--|--------------------------|---------------------|--------------------------|
| 1 | Толстой | Лев | Николаевич |
| 2 | Пушкин | Александр | Сергеевич |
| 3 | Достоевский | Фёдор | Михайлович |
| 4 | Чехов | Антон | Павлович |
| 5 | Гоголь | Николай | Васильевич |

Таблица 6 – Dataset таблицы book

| Id (идентификатор книги в базе данных) | name (название книги) | author_id (идентификатор автора в базе данных) | deposit (размер депозит а) | cost_per_day (стоимость за день аренды книги) | comment (комментарий о состоянии книги) |
|---|--------------------------|---|-------------------------------------|---|--|
| 1 | Война и мир | 1 | 1000 | 10 | Отличное состояние |
| 2 | Анна Каренина | 1 | 500 | 20 | Хорошее состояние |
| 3 | Детство | 1 | 1500 | 30 | Новая книга |
| 4 | Юность | 1 | 800 | 40 | Слегка изношенная |
| 5 | Отрочество | 1 | 1200 | 50 | Идеальное состояние |
| 6 | Медный всадник | 2 | 1000 | 10 | Идеальное состояние |
| 7 | Капитанская дочка | 2 | 800 | 20 | Отличное состояние |
| 8 | Дубровский | 2 | 1500 | 30 | Хорошее состояние |
| 9 | Сказка о царе Салтане | 2 | 1200 | 40 | Новая книга |
| 10 | Сказка о рыбаке и рыбке | 2 | 1000 | 50 | Слегка изношенная |

| Id (идентификатор книги в базе данных) | name (название книги) | author_id (идентификатор автора в базе данных) | deposit (размер депозит а) | cost_per_day (стоимость за день аренды книги) | comment (комментарий о состоянии книги) |
|--|--------------------------|--|-------------------------------------|---|--|
| 11 | Преступление и наказание | 3 | 1000 | 10 | Отличное состояние |
| 12 | Братья Карамазовы | 3 | 500 | 20 | Хорошее состояние |
| 13 | Бесы | 3 | 1500 | 30 | Новая книга |
| 14 | Бедные люди | 3 | 800 | 40 | Слегка изношенная |
| 15 | Двойник | 3 | 1200 | 50 | Идеальное состояние |
| 16 | Рассказы | 4 | 1000 | 10 | Отличное состояние |
| 17 | Человек в футляре | 4 | 800 | 20 | Хорошее состояние |
| 18 | Хамелеон | 4 | 1500 | 30 | Новая книга |
| 19 | Дама с собачкой | 4 | 1200 | 40 | Слегка изношенная |
| 20 | Палата №6 | 4 | 1000 | 50 | Идеальное состояние |
| 21 | Мёртвые души | 5 | 1000 | 10 | Отличное состояние |
| 22 | Тарас Бульба | 5 | 500 | 20 | Хорошее состояние |
| 23 | Ночь перед Рождеством | 5 | 1500 | 30 | Новая книга |
| 24 | Вий | 5 | 800 | 40 | Слегка изношенная |
| 25 | Нос | 5 | 1200 | 50 | Идеальное состояние |

Таблица 7 – Dataset таблицы book_genre

| book_id (идентификатор книги в базе данных) | genre_id (идентификатор жанра в базе данных) |
|--|---|
| 1 | 7 |
| 1 | 10 |
| 2 | 10 |
| 2 | 2 |
| 3 | 10 |
| 3 | 7 |
| 4 | 10 |
| 4 | 7 |
| 5 | 10 |
| 5 | 7 |
| 6 | 6 |
| 7 | 10 |

| book_id (идентификатор книги в базе данных) | genre_id (идентификатор жанра в базе данных) |
|--|---|
| 8 | 10 |
| 9 | 12 |
| 10 | 12 |
| 11 | 10 |
| 12 | 10 |
| 12 | 14 |
| 13 | 10 |
| 13 | 14 |
| 14 | 9 |
| 15 | 9 |
| 15 | 13 |
| 16 | 9 |
| 17 | 9 |
| 18 | 9 |
| 19 | 9 |
| 20 | 9 |
| 21 | 6 |
| 21 | 11 |
| 22 | 9 |
| 23 | 9 |
| 23 | 5 |
| 24 | 9 |
| 24 | 5 |
| 25 | 9 |
| 25 | 11 |

Таблица 8 - Dataset таблицы category

| id (идентификатор категории в базе данных) | name (название категории) | discount (размер скидки в %) |
|--|------------------------------|---------------------------------|
| 1 | Школьник | 15 |
| 2 | Студент | 10 |
| 3 | Пенсионер | 30 |
| 4 | Стандарт | 0 |

Таблица 9 – Dataset таблицы customer_category

| customer_id (идентификатор клиента в базе данных) | category_id (идентификатор категории в базе данных) |
|--|--|
| 1 | 3 |
| 2 | 4 |
| 3 | 4 |
| 4 | 4 |
| 5 | 4 |
| 6 | 3 |

| customer_id (идентификатор клиента в базе данных) | category_id (идентификатор категории в базе данных) |
|--|--|
| 7 | 2 |
| 8 | 2 |
| 9 | 1 |
| 10 | 4 |

Таблица 10 – Dataset таблицы customer

| id (идентификатор клиента в базе данных) | second_name (фамилия) | first_name (имя) | third_name (отчество) | address (адрес проживания) | contacts (контакты клиента) |
|--|--------------------------|---------------------|--------------------------|----------------------------------|-----------------------------------|
| 1 | Иванов | Иван | Иванович | ул. Ленина, 10 | +7 (912) 345-67-89 |
| 2 | Петров | Петр | Петрович | ул. Пушкина, 5 | +7 (923) 456-78-90 |
| 3 | Сидорова | Анна | Викторовна | пр. Гагарина, 20 | +7 (934) 567-89-01 |
| 4 | Смирнов | Алексей | Игоревич | ул. Московская, 15 | +7 (945) 678-90-12 |
| 5 | Козлова | Мария | Сергеевна | пр. Ленинский, 30 | +7 (956) 789-01-23 |
| 6 | Николаев | Дмитрий | Анатольевич | ул. Советская, 25 | +7 (967) 890-12-34 |
| 7 | Васильева | Екатерина | Александровна | пр. Победы, 12 | +7 (978) 901-23-45 |
| 8 | Морозов | Игорь | Владимирович | ул. Зеленая, 8 | +7 (989) 012-34-56 |
| 9 | Федоров | Артем | Олегович | ул. Пролетарская, 18 | +7 (990) 123-45-67 |
| 10 | Кузнецова | Анастасия | Павловна | пр. Свердлова, 7 | +7 (901) 234-56-78 |

Таблица 11 – Dataset таблицы genre

| id (идентификатор жанра в базе данных) | name (название жанра) |
|---|-----------------------------|
| 1 | Детектив |
| 2 | Драма |
| 3 | Комедия |
| 4 | Лирика |
| 5 | Мистика |
| 6 | Поэма |

| id (идентификатор жанра в базе данных) | name (название жанра) |
|---|-----------------------------|
| 7 | Проза |
| 8 | Пьеса |
| 9 | Рассказ |
| 10 | Роман |
| 11 | Сатира |
| 12 | Сказка |
| 13 | Фантастика |
| 14 | Философия |
| 15 | Фэнтези |

Таблица 12 – Dataset таблицы customer

| id (идентификатор транзакции в базе данных) | customer_id (идентификатор клиента в базе данных) | book_id (идентификатор книги в базе данных) | date_out (дата выдачи книги в прокат) | date_in (дата возвращения книги из проката) | penalty (штраф) | total (итоговая стоимость проката) |
|---|---|---|---|---|--------------------|---|
| 1 | 1 | 3 | 2023-01-10 | 2023-02-01 | 0 | 461 |
| 2 | 5 | 3 | 2023-02-15 | 2023-03-02 | 0 | 450 |
| 3 | 9 | 3 | 2023-04-01 | 2023-04-07 | 0 | 153 |
| 4 | 2 | 6 | 2023-01-11 | 2023-02-08 | 100 | 280 |
| 5 | 2 | 8 | 2023-01-16 | 2023-02-08 | 0 | 690 |
| 6 | 3 | 9 | 2023-01-18 | 2023-03-14 | 0 | 2280 |
| 7 | 3 | 25 | 2023-02-21 | 2023-02-26 | 200 | 450 |
| 8 | 4 | 1 | 2023-04-02 | 2023-05-08 | 0 | 280 |
| 9 | 10 | 1 | 2023-01-11 | 2023-02-08 | 0 | 280 |
| 10 | 6 | 20 | 2023-03-16 | 2023-03-24 | 0 | 280 |
| 11 | 7 | 21 | 2023-03-03 | 2023-03-23 | 0 | 180 |
| 12 | 5 | 7 | 2023-04-15 | 2023-05-01 | 0 | 320 |
| 13 | 6 | 14 | 2023-02-03 | 2023-02-22 | 50 | 582 |
| 14 | 7 | 13 | 2023-03-30 | 2023-04-01 | 600 | 672 |

3.2 Добавление данных в базу

Разработанный dataset был добавлен в таблицу при помощи SQL-запросов, указанных в «Листинге 11 – внесение данных в систему»

Листинг 11 – Внесение данных в систему

```
-----  
-- Dataset author -----
```

```
INSERT INTO author (second_name, first_name, third_name)  
VALUES
```

```

('Толстой',      'Лев',      'Николаевич'),
('Пушкин',      'Александр', 'Сергеевич'),
('Достоевский', 'Фёдор',      'Михайлович'),
('Чехов',      'Антон',      'Павлович'),
('Гоголь',      'Николай',    'Васильевич');

-----
-- Dataset book -----

INSERT INTO book (name, author_id, deposit, cost_per_day,
comment)
VALUES
-- Автор 1: Толстой
    ('Война и мир',      1, 1000, 10, 'Отличное
состояние'),
    ('Анна Каренина',    1, 500, 20, 'Хорошее
состояние'),
    ('Детство',          1, 1500, 30, 'Новая книга'),
    ('Юность',           1, 800, 40, 'Слегка
изношенная'),
    ('Отрочество',       1, 1200, 50, 'Идеальное
состояние'),

-- Автор 2: Пушкин
    ('Медный всадник',   2, 1000, 10, 'Идеальное
состояние'),
    ('Капитанская дочка', 2, 800, 20, 'Отличное
состояние'),
    ('Дубровский',       2, 1500, 30, 'Хорошее
состояние'),
    ('Сказка о царе Салтане', 2, 1200, 40, 'Новая книга'),
    ('Сказка о рыбаке и рыбке', 2, 1000, 50, 'Слегка
изношенная'),

```

-- Автор 3: Достоевский

('Преступление и наказание', 3, 1000, 10, 'Отличное
состояние'),
('Братья Карамазовы', 3, 500, 20, 'Хорошее
состояние'),
('Бесы', 3, 1500, 30, 'Новая книга'),
('Бедные люди', 3, 800, 40, 'Слегка
изношенная'),
('Двойник', 3, 1200, 50, 'Идеальное
состояние'),

-- Автор 4: Чехов

('Рассказы', 4, 1000, 10, 'Отличное
состояние'),
('Человек в футляре', 4, 800, 20, 'Хорошее
состояние'),
('Хамелеон', 4, 1500, 30, 'Новая книга'),
('Дама с собачкой', 4, 1200, 40, 'Слегка
изношенная'),
('Палата №6', 4, 1000, 50, 'Идеальное
состояние'),

-- Автор 5: Гоголь

('Мёртвые души', 5, 1000, 10, 'Отличное
состояние'),
('Тарас Бульба', 5, 500, 20, 'Хорошее
состояние'),
('Ночь перед Рождеством', 5, 1500, 30, 'Новая книга'),
('Вий', 5, 800, 40, 'Слегка
изношенная'),
('Нос', 5, 1200, 50, 'Идеальное
состояние');

```

-----
-- Dataset genre -----

INSERT INTO genre (name)
VALUES
    ('Детектив'),
    ('Драма'),
    ('Комедия'),
    ('Лирика'),
    ('Мистика'),
    ('Поэма'),
    ('Проза'),
    ('Пьеса'),
    ('Рассказ'),
    ('Роман'),
    ('Сатира'),
    ('Сказка'),
    ('Фантастика'),
    ('Философия'),
    ('Фэнтези');

-----

-- Dataset book_genre -----

INSERT INTO book_genre (book_id, genre_id) VALUES
-- Автор 1: Толстой
    -- Книга 1: Война и мир
    (1, 10),    -- Роман
    (1, 7),     -- Проза
    -- Книга 2: Анна Каренина
    (2, 10),    -- Роман
    (2, 2),     -- Драма
    -- Книга 3: Детство
    (3, 10),    -- Роман

```

(3, 7), -- Проза
-- Книга 4: Юность
(4, 10), -- Роман
(4, 7), -- Проза
-- Книга 5: Отрочество
(5, 10), -- Роман
(5, 7), -- Проза

-- Автор 2: Пушкин
-- Книга 6: Мёртвые души
(6, 6), -- Поэма
-- Книга 7: Капитанская дочка
(7, 10), -- Роман
-- Книга 8: Дубровский
(8, 10), -- Роман
-- Книга 9: Сказка о царе Салтане
(9, 12), -- Сказка
-- Книга 10: Сказка о рыбаке и рыбке
(10, 12), -- Сказка

-- Автор 3: Достоевский
-- Книга 11: Преступление и наказание
(11, 10), -- Роман
-- Книга 12: Братья Карамазовы
(12, 10), -- Роман
(12, 14), -- Философия
-- Книга 13: Бесы
(13, 10), -- Роман
(13, 14), -- Философия
-- Книга 14: Бедные люди
(14, 9), -- Рассказ
-- Книга 15: Двойник
(15, 9), -- Рассказ
(15, 13), -- Фантастика


```

-- Автор 4: Чехов
  -- Книга 16: Рассказы
  (16, 9),    -- Рассказ
  -- Книга 17: Человек в футляре
  (17, 9),    -- Рассказ
  -- Книга 18: Хамелеон
  (18, 9),    -- Рассказ
  -- Книга 19: Дама с собачкой
  (19, 9),    -- Рассказ
  -- Книга 20: Палата №6
  (20, 9),    -- Рассказ

-- Автор 5: Гоголь
  -- Книга 21: Мёртвые души
  (21, 6),    -- Поэма
  (21, 11),   -- Сатира
  -- Книга 22: Тарас Бульба
  (22, 9),    -- Рассказ
  -- Книга 23: Ночь перед Рождеством
  (23, 9),    -- Рассказ
  (23, 5),    -- Мистика
  -- Книга 24: Вий
  (24, 9),    -- Рассказ
  (24, 5),    -- Мистика
  -- Книга 25: Нос
  (25, 9),    -- Рассказ
  (25, 11);   -- Сатира

-----
-- Dataset category -----

INSERT INTO category (name, discount) VALUES
  ('Школьник', 15),

```

```

('Студент', 10),
('Пенсионер', 30),
('Стандарт', 0);

-----
-- Dataset customer -----

INSERT INTO customer (second_name, first_name, third_name,
address, contacts) VALUES
    ('Иванов',      'Иван',      'Иванович',      'ул. Ленина,
10',      '+7 (912) 345-67-89'),
    ('Петров',      'Петр',      'Петрович',      'ул. Пушкина,
5',      '+7 (923) 456-78-90'),
    ('Сидорова',    'Анна',      'Викторовна',    'пр. Гагарина,
20',      '+7 (934) 567-89-01'),
    ('Смирнов',     'Алексей',    'Игоревич',      'ул. Московская,
15',      '+7 (945) 678-90-12'),
    ('Козлова',     'Мария',      'Сергеевна',     'пр. Ленинский,
30',      '+7 (956) 789-01-23'),
    ('Николаев',    'Дмитрий',    'Анатольевич',   'ул. Советская,
25',      '+7 (967) 890-12-34'),
    ('Васильева',   'Екатерина',  'Александровна', 'пр. Победы,
12',      '+7 (978) 901-23-45'),
    ('Морозов',     'Игорь',      'Владимирович',  'ул. Зеленая,
8',      '+7 (989) 012-34-56'),
    ('Федоров',     'Артем',      'Олегович',      'ул.
Пролетарская, 18', '+7 (990) 123-45-67'),
    ('Кузнецова',   'Анастасия',  'Павловна',      'пр. Свердлова,
7',      '+7 (901) 234-56-78');

-----

```

-- Dataset customer_category -----

INSERT INTO customer_category

VALUES

(1, 3),
(2, 4),
(3, 4),
(4, 4),
(5, 4),
(6, 3),
(7, 2),
(8, 2),
(9, 1),
(10, 4);

-- Dataset transaction -----

INSERT INTO TRANSACTION

(customer_id, book_id, date_out, date_in, penalty, total)

VALUES

-- Книги забирали и вернули из проката

(1, 3, '2023-01-10', '2023-02-01', 0, 461),
(5, 3, '2023-02-15', '2023-03-02', 0, 450),
(9, 3, '2023-04-01', '2023-04-07', 0, 153),
(2, 6, '2023-01-11', '2023-02-08', 100, 280),
(2, 8, '2023-01-16', '2023-02-08', 0, 690),
(3, 9, '2023-01-18', '2023-03-14', 0, 2280),
(3, 25, '2023-02-21', '2023-02-26', 200, 450),
(4, 1, '2023-04-02', '2023-05-08', 300, 360),
(10, 1, '2023-01-11', '2023-02-08', 0, 280),
(6, 20, '2023-03-16', '2023-03-24', 0, 280),
(7, 21, '2023-03-03', '2023-03-23', 0, 180),
(5, 7, '2023-04-15', '2023-05-01', 0, 320),

```
(6, 14, '2023-02-03', '2023-02-22', 50, 582),
(7, 13, '2023-03-30', '2023-04-01', 600, 672);
```

```
INSERT INTO TRANSACTION
(customer_id, book_id, date_out)
VALUES
-- Книги еще не вернули из проката
(8, 11, '2023-04-08'),
(9, 23, '2023-05-11'),
(10, 2, '2023-02-06');
```

```
UPDATE book SET available = FALSE
WHERE id = 11 or id = 23 or id = 2;
```

3.3 Запросы к базе данных

Для работы с базой данных были разработаны следующие запросы:

1) Вывод всех названий книг. Код запроса указан в «Листинге 12 – Запрос 1»;

Листинг 12 – Запрос 1

```
SELECT name FROM book;
```

Вывод запроса указан на «Рисунке 4 – Вывод запроса 1».

| Data Output | | Messages | Notifications |
|-------------|---------------------------|----------|---------------|
| | name character varying | | |
| 1 | Война и мир | | |
| 2 | Детство | | |
| 3 | Юность | | |
| 4 | Отрочество | | |
| 5 | Медный всадник | | |
| 6 | Капитанская дочка | | |
| 7 | Дубровский | | |
| 8 | Сказка о царе Салтане | | |
| 9 | Сказка о рыбаке и рыбке | | |
| 10 | Братья Карамазовы | | |

Рисунок 4 – Вывод запроса 1

2) Вывод размера скидки для каждого клиента. Код запроса указан в «Листинге 13 – Запрос 2»;

Листинг 13 – Запрос 2

```
SELECT second_name, first_name, third_name, discount
FROM customer, customer_category, category
WHERE customer_id = customer.id AND
category_id = category.id;
```

Вывод запроса указан на «Рисунке 5 – Вывод запроса 2».

| | second_name character varying | first_name character varying | third_name character varying | discount real |
|----|----------------------------------|---------------------------------|---------------------------------|------------------|
| 1 | Иванов | Иван | Иванович | 30 |
| 2 | Петров | Петр | Петрович | 0 |
| 3 | Сидорова | Анна | Викторовна | 0 |
| 4 | Смирнов | Алексей | Игоревич | 0 |
| 5 | Козлова | Мария | Сергеевна | 0 |
| 6 | Николаев | Дмитрий | Анатолевич | 30 |
| 7 | Васильева | Екатерина | Александровна | 10 |
| 8 | Морозов | Игорь | Владимирович | 10 |
| 9 | Федоров | Артем | Олегович | 15 |
| 10 | Кузнецова | Анастасия | Павловна | 0 |

Рисунок 5 – Вывод запроса 2

3) Вывод контактов клиента, названия книги для каждой транзакции в базе данных. Код запроса указан в «Листинге 14 – Запрос 3»;

Листинг 14 – Запрос 3

```
SELECT contacts, book.name, date_out, date_in, penalty, total
FROM transaction, customer, book
WHERE customer_id = customer.id AND
book_id = book.id;
```

Вывод запроса указан на «Рисунке 6 – Вывод запроса 3».

| | contacts character varying | name character varying | date_out date | date_in date | penalty real | total real |
|----|-------------------------------|---------------------------|------------------|-----------------|-----------------|---------------|
| 1 | +7 (912) 345-67-89 | Детство | 2023-01-10 | 2023-02-01 | 0 | 461 |
| 2 | +7 (956) 789-01-23 | Детство | 2023-02-15 | 2023-03-02 | 0 | 450 |
| 3 | +7 (990) 123-45-67 | Детство | 2023-04-01 | 2023-04-07 | 0 | 153 |
| 4 | +7 (923) 456-78-90 | Медный всадник | 2023-01-11 | 2023-02-08 | 100 | 280 |
| 5 | +7 (923) 456-78-90 | Дубровский | 2023-01-16 | 2023-02-08 | 0 | 690 |
| 6 | +7 (934) 567-89-01 | Сказка о царе Салтане | 2023-01-18 | 2023-03-14 | 0 | 2280 |
| 7 | +7 (934) 567-89-01 | Нос | 2023-02-21 | 2023-02-26 | 200 | 450 |
| 8 | +7 (945) 678-90-12 | Война и мир | 2023-04-02 | 2023-05-08 | 300 | 360 |
| 9 | +7 (901) 234-56-78 | Война и мир | 2023-01-11 | 2023-02-08 | 0 | 280 |
| 10 | +7 (967) 890-12-34 | Палата №6 | 2023-03-16 | 2023-03-24 | 0 | 280 |

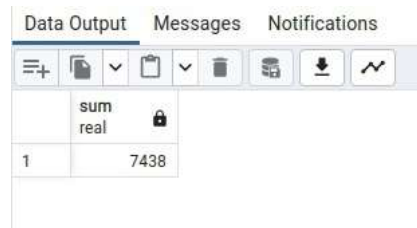
Рисунок 6 – Вывод запроса 3

4) Вывод суммы всех транзакций. Код запроса указан в «Листинге 15 – Запрос 4»;

Листинг 15 – Запрос 4

```
SELECT SUM(total) FROM transaction
WHERE total IS NOT NULL;
```

Вывод запроса указан на «Рисунке 7 – Вывод запроса 4».



| | sum | real |
|---|-----|------|
| 1 | | 7438 |

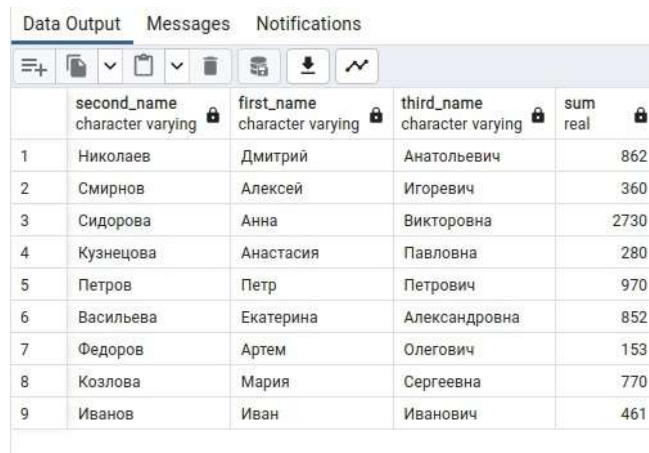
Рисунок 7 – Вывод запроса 4

5) Вывод суммы транзакций каждого клиента. Код запроса указан в «Листинге 16 – Запрос 5»;

Листинг 16 – Запрос 5

```
SELECT second_name, first_name, third_name, SUM(total)
FROM transaction, customer
WHERE total IS NOT NULL AND
customer_id = customer.id
GROUP BY second_name, first_name, third_name;
```

Вывод запроса указан на «Рисунке 8 – Вывод запроса 5»



| | second_name character varying | first_name character varying | third_name character varying | sum real |
|---|----------------------------------|---------------------------------|---------------------------------|-------------|
| 1 | Николаев | Дмитрий | Анатолевич | 862 |
| 2 | Смирнов | Алексей | Игоревич | 360 |
| 3 | Сидорова | Анна | Викторовна | 2730 |
| 4 | Кузнецова | Анастасия | Павловна | 280 |
| 5 | Петров | Петр | Петрович | 970 |
| 6 | Васильева | Екатерина | Александровна | 852 |
| 7 | Федоров | Артем | Олегович | 153 |
| 8 | Козлова | Мария | Сергеевна | 770 |
| 9 | Иванов | Иван | Иванович | 461 |

Рисунок 8 – Вывод запроса 5

6) Вывод суммы всех транзакций без учета скидки. Код запроса указан в «Листинге 17 – Запрос 6»;

Листинг 17 – Запрос 6

```
SELECT id, customer_id, book_id, date_out, date_in,
penalty, total, SUM(interval_counter(date_out, date_in) *
(select cost_per_day from book where book_id = book.id))
```

```

OVER (PARTITION BY id) AS "total without discount"
FROM
transaction;

```

Вывод запроса указан на «Рисунке 9 – Вывод запроса 6».

| | id [PK] integer | customer_id integer | book_id integer | date_out date | date_in date | penalty real | total real | total without discount double precision |
|----|--------------------|------------------------|--------------------|------------------|-----------------|-----------------|---------------|--|
| 1 | 1 | 1 | 3 | 2023-01-10 | 2023-02-01 | 0 | 461 | 660 |
| 2 | 2 | 5 | 3 | 2023-02-15 | 2023-03-02 | 0 | 450 | 450 |
| 3 | 3 | 9 | 3 | 2023-04-01 | 2023-04-07 | 0 | 153 | 180 |
| 4 | 4 | 2 | 6 | 2023-01-11 | 2023-02-08 | 100 | 280 | 280 |
| 5 | 5 | 2 | 8 | 2023-01-16 | 2023-02-08 | 0 | 690 | 690 |
| 6 | 6 | 3 | 9 | 2023-01-18 | 2023-03-14 | 0 | 2280 | 2280 |
| 7 | 7 | 3 | 25 | 2023-02-21 | 2023-02-26 | 200 | 450 | 250 |
| 8 | 8 | 4 | 1 | 2023-04-02 | 2023-05-08 | 300 | 360 | 360 |
| 9 | 9 | 10 | 1 | 2023-01-11 | 2023-02-08 | 0 | 280 | 280 |
| 10 | 10 | 6 | 20 | 2023-03-16 | 2023-03-24 | 0 | 290 | 400 |
| 11 | 11 | 7 | 21 | 2023-03-03 | 2023-03-23 | 0 | 180 | 200 |
| 12 | 12 | 5 | 7 | 2023-04-15 | 2023-05-01 | 0 | 320 | 320 |
| 13 | 13 | 6 | 14 | 2023-02-03 | 2023-02-22 | 50 | 582 | 760 |
| 14 | 14 | 7 | 13 | 2023-03-30 | 2023-04-01 | 600 | 672 | 60 |
| 15 | 15 | 8 | 11 | 2023-04-08 | [null] | 0 | [null] | [null] |
| 16 | 16 | 9 | 23 | 2023-05-11 | [null] | 0 | [null] | [null] |
| 17 | 17 | 10 | 2 | 2023-02-06 | [null] | 0 | [null] | [null] |

Рисунок 9 – Вывод запроса 6

7) Вывод средней длительности проката книги. Код запроса указан в «Листинге 9 – Запрос 7»;

Листинг 18 – Запрос 7

```

SELECT AVG(interval_counter(date_out, date_in))
FROM transaction WHERE date_in IS NOT NULL;

```

Вывод запроса указан на «Рисунке 10 – Вывод запроса 7».

| | avg numeric |
|---|---------------------|
| 1 | 20.3571428571428571 |

Рисунок 10 – Вывод запроса 7

8) Вывод количества клиентов для каждой категории. Код запроса указан в «Листинге 19 – Запрос 8»;

Листинг 19 – Запрос 8

```

SELECT name, COUNT(*)
FROM category, customer_category
WHERE category.id = category_id
GROUP BY name, category_id;

```

Вывод запроса указан на «Рисунке 11 – Вывод запроса 8».

| Data Output | | Messages | Notifications |
|-------------|---------------------------|-----------------|---------------|
| | name character varying | count bigint | |
| 1 | Студент | 2 | |
| 2 | Пенсионер | 2 | |
| 3 | Школьник | 1 | |
| 4 | Стандарт | 5 | |

Рисунок 11 – Вывод запроса 8

9) Вывод сводки о клиентах. Если у клиента более одного проката, то выводится «Частный клиент», иначе – «Обычный клиент». Код запроса указан в «Листинге 20 – Запрос 9»;

Листинг 20 – Запрос 9

```
SELECT second_name, first_name, third_name,
CASE
    WHEN COUNT(*) > 1 THEN 'Частый клиент'
    ELSE 'Обычный клиент'
END AS "information"
FROM customer, transaction
WHERE customer_id = customer.id AND
date_in IS NOT NULL
GROUP BY second_name, first_name, third_name;
```

Вывод запроса указан на «Рисунке 12 – Вывод запроса 9».

| Data Output | | Messages | Notifications | |
|--|----------------------------------|---------------------------------|---------------------------------|---------------------|
| <div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div></div> | | | | |
| | second_name character varying | first_name character varying | third_name character varying | information text |
| 1 | Николаев | Дмитрий | Анатольевич | Частый клиент |
| 2 | Смирнов | Алексей | Игоревич | Обычный клиент |
| 3 | Сидорова | Анна | Викторовна | Частый клиент |
| 4 | Кузнецова | Анастасия | Павловна | Обычный клиент |
| 5 | Петров | Петр | Петрович | Частый клиент |
| 6 | Васильева | Екатерина | Александровна | Частый клиент |
| 7 | Федоров | Артем | Олегович | Обычный клиент |
| 8 | Козлова | Мария | Сергеевна | Частый клиент |
| 9 | Иванов | Иван | Иванович | Обычный клиент |

Рисунок 12 – Вывод запроса 9

10) Вывод всех книг и их авторов. Код запроса указан в «Листинге 21 – Запрос 10»;

Листинг 21 – Запрос 10

```
SELECT second_name, first_name, third_name, name
FROM author
JOIN (SELECT author_id, name FROM book) b
ON author.id = b.author_id;
```

Вывод запроса указан на «Рисунке 13 – Вывод запроса 10».

| | second_name character varying | first_name character varying | third_name character varying | name character varying |
|----|----------------------------------|---------------------------------|---------------------------------|---------------------------|
| 1 | Толстой | Лев | Николаевич | Война и мир |
| 2 | Толстой | Лев | Николаевич | Детство |
| 3 | Толстой | Лев | Николаевич | Юность |
| 4 | Толстой | Лев | Николаевич | Отрочество |
| 5 | Пушкин | Александр | Сергеевич | Медный всадник |
| 6 | Пушкин | Александр | Сергеевич | Капитанская дочка |
| 7 | Пушкин | Александр | Сергеевич | Дубровский |
| 8 | Пушкин | Александр | Сергеевич | Сказка о царе Салтане |
| 9 | Пушкин | Александр | Сергеевич | Сказка о рыбаке и рыбке |
| 10 | Достоевский | Фёдор | Михайлович | Братья Карамазовы |
| 11 | Достоевский | Фёдор | Михайлович | Бесы |

Рисунок 13 – Вывод запроса 10

11) Вывод последних 5 транзакций. Код запроса указан в «Листинге 22 – Запрос 11»;

Листинг 22 – Запрос 11

```
SELECT * FROM transaction
ORDER BY date_out DESC
LIMIT 5;
```

Вывод запроса указан на «Рисунке 14 – Вывод запроса 11».

| | id [PK] integer | customer_id integer | book_id integer | date_out date | date_in date | penalty real | total real |
|---|--------------------|------------------------|--------------------|------------------|-----------------|-----------------|---------------|
| 1 | 16 | 9 | 23 | 2023-05-11 | [null] | 0 | [null] |
| 2 | 12 | 5 | 7 | 2023-04-15 | 2023-05-01 | 0 | 320 |
| 3 | 15 | 8 | 11 | 2023-04-08 | [null] | 0 | [null] |
| 4 | 8 | 4 | 1 | 2023-04-02 | 2023-05-08 | 300 | 360 |
| 5 | 3 | 9 | 3 | 2023-04-01 | 2023-04-07 | 0 | 153 |

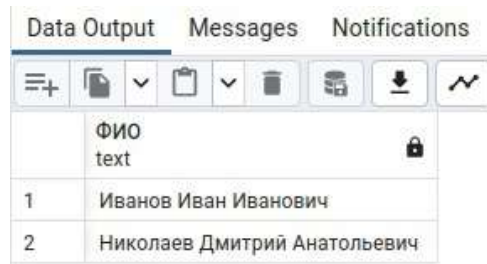
Рисунок 14 – Вывод запроса 11

12) Вывод ФИО каждого пенсионера. Код запроса указан в «Листинге 23 – Запрос 12»;

Листинг 23 – Запрос 12

```
SELECT second_name || ' ' || first_name || ' ' || third_name AS  
"ФИО"  
FROM customer, category, customer_category  
WHERE customer_id = customer.id AND  
category.id = category_id AND  
category.name = 'Пенсионер';
```

Вывод запроса указан на «Рисунке 15 – Вывод запроса 12».



| | ФИО | text |
|---|------------------------------|------|
| 1 | Иванов Иван Иванович | |
| 2 | Николаев Дмитрий Анатольевич | |

Рисунок 15 – Вывод запроса 12

13) Вывод информации для каждого жанра: сколько книг такого жанра, какие авторы у книг этого жанра. Код запроса указан в «Листинге 24 – Запрос 13»;

Листинг 24 – Запрос 13

```
SELECT g.name AS genre_name,  
COUNT(*) AS book_count,  
STRING_AGG(DISTINCT a.first_name || ' ' || a.second_name,  
, ' ') AS authors  
FROM transaction t  
JOIN customer c ON t.customer_id = c.id  
JOIN book b ON t.book_id = b.id  
JOIN book_genre bg ON b.id = bg.book_id  
JOIN genre g ON bg.genre_id = g.id  
JOIN author a ON b.author_id = a.id  
WHERE c.id = customer_id  
GROUP BY g.name  
ORDER BY book_count DESC;
```

Вывод запроса указан на «Рисунке 16 – Вывод запроса 13».

| Data Output | | | Messages | Notifications |
|-------------|-------------------|------------|--|---------------|
| | | | Save results to file | |
| | | | F8 | |
| | genre_name | book_count | authors | |
| | character varying | | text | |
| 1 | Роман | | Александр Пушкин, Лев Толстой, Фёдор Достоевский | |
| 2 | Проза | 5 | Лев Толстой | |
| 3 | Рассказ | 4 | Антон Чехов, Николай Гоголь, Фёдор Достоевский | |
| 4 | Сатира | 2 | Николай Гоголь | |
| 5 | Поэма | 2 | Александр Пушкин, Николай Гоголь | |
| 6 | Философия | 1 | Фёдор Достоевский | |
| 7 | Мистика | 1 | Николай Гоголь | |
| 8 | Сказка | 1 | Александр Пушкин | |
| 9 | Драма | 1 | Лев Толстой | |

Рисунок 16 – Вывод запроса 13

14) Вывод любимого жанра клиента. Код запроса указан в «Листинге 25 – Запрос 14»;

Листинг 25 – Запрос 14

```
SELECT g.name AS favorite_genre
FROM genre g
JOIN book_genre bg ON g.id = bg.genre_id
JOIN book b ON bg.book_id = b.id
JOIN transaction t ON b.id = t.book_id
WHERE t.customer_id = 1
GROUP BY g.name
ORDER BY COUNT(*) DESC
LIMIT 1;
```

Вывод запроса указан на «Рисунке 17 – Вывод запроса 14».

| Data Output | | | Messages | Notifications |
|-------------|-------------------|--|----------|---------------|
| | | | | |
| | favorite_genre | | | |
| | character varying | | | |
| 1 | Проза | | | |

Рисунок 17 – Вывод запроса 14

15) Вывод книг, которые ни разу не брали в прокат. Код запроса указан в «Листинге 26 – Запрос 15»;

Листинг 26 – Запрос 15

```
SELECT name
FROM book
WHERE
id NOT IN (SELECT book_id FROM transaction)
```

Вывод запроса указан на «Рисунке 18 – Вывод запроса 15».

| Data Output | | Messages | Notifications |
|-------------|---------------------------|----------|---------------|
| | | | |
| | name character varying | | |
| 1 | Юность | | |
| 2 | Отрочество | | |
| 3 | Сказка о рыбаке и рыбке | | |
| 4 | Братья Карамазовы | | |
| 5 | Двойник | | |
| 6 | Рассказы | | |
| 7 | Человек в футляре | | |
| 8 | Хамелеон | | |
| 9 | Дама с собачкой | | |
| 10 | Тарас Бульба | | |
| 11 | Вий | | |

Рисунок 18 – Вывод запроса 15

16) Вывод информации для каждой книги: сколько раз ее брали в прокат. Код запроса указан в «Листинге 27 – Запрос 16»;

Листинг 27 – Запрос 16

```
SELECT b.name AS book_name, COUNT(*) AS rent_count
FROM book b
JOIN transaction t ON b.id = t.book_id
GROUP BY b.name
ORDER BY rent_count DESC;
```

Вывод запроса указан на «Рисунке 19 – Вывод запроса 16».

| Data Output | | Messages | Notifications |
|-------------|--------------------------------|----------|----------------------|
| | | | |
| | book_name character varying | | rent_count bigint |
| 1 | Детство | | 3 |
| 2 | Война и мир | | 2 |
| 3 | Медный всадник | | 1 |
| 4 | Капитанская дочка | | 1 |
| 5 | Нос | | 1 |
| 6 | Мёртвые души | | 1 |
| 7 | Бесы | | 1 |
| 8 | Палата №6 | | 1 |
| 9 | Преступление и наказание | | 1 |
| 10 | Бедные люди | | 1 |
| 11 | Дубровский | | 1 |
| 12 | Ночь перед Рождеством | | 1 |
| 13 | Анна Каренина | | 1 |
| 14 | Сказка о царе Салтане | | 1 |

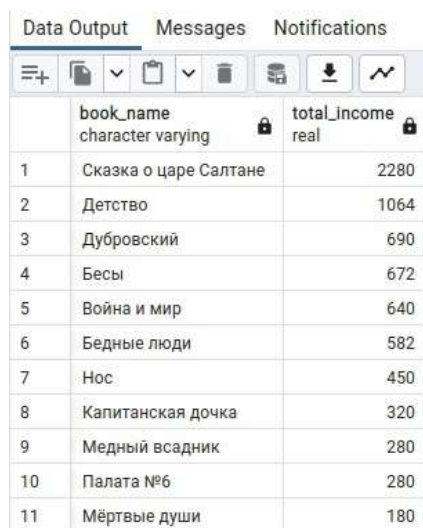
Рисунок 19 – Вывод запроса 16

17) Вывод суммы транзакций для каждой книги. Код запроса указан в «Листинге 28 – Запрос 17»;

Листинг 28 – Запрос 17

```
SELECT b.name AS book_name, SUM(t.total) AS total_income
FROM book b
JOIN transaction t ON b.id = t.book_id
WHERE t.date_in IS NOT NULL
GROUP BY b.name
ORDER BY total_income DESC;
```

Вывод запроса указан на «Рисунке 20 – Вывод запроса 17».



| | book_name character varying | total_income real |
|----|--------------------------------|----------------------|
| 1 | Сказка о царе Салтане | 2280 |
| 2 | Детство | 1064 |
| 3 | Дубровский | 690 |
| 4 | Бесы | 672 |
| 5 | Война и мир | 640 |
| 6 | Бедные люди | 582 |
| 7 | Нос | 450 |
| 8 | Капитанская дочка | 320 |
| 9 | Медный всадник | 280 |
| 10 | Палата №6 | 280 |
| 11 | Мёртвые души | 180 |

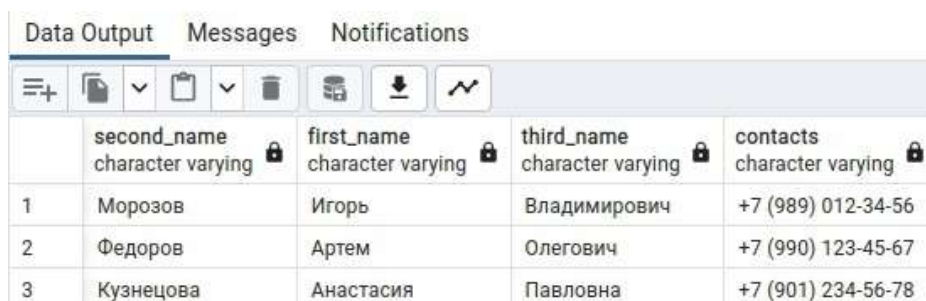
Рисунок 20 – Вывод запроса 17

18) Вывод списка клиентов, кто сейчас арендует книгу. Код запроса указан в «Листинге 29 – Запрос 18»;

Листинг 29 – Запрос 18

```
SELECT second_name, first_name, third_name, contacts
FROM customer
JOIN transaction t ON customer.id = t.customer_id
WHERE date_in IS NULL;
```

Вывод запроса указан на «Рисунке 21 – Вывод запроса 18».



| | second_name character varying | first_name character varying | third_name character varying | contacts character varying |
|---|----------------------------------|---------------------------------|---------------------------------|-------------------------------|
| 1 | Морозов | Игорь | Владимирович | +7 (989) 012-34-56 |
| 2 | Федоров | Артем | Олегович | +7 (990) 123-45-67 |
| 3 | Кузнецова | Анастасия | Павловна | +7 (901) 234-56-78 |

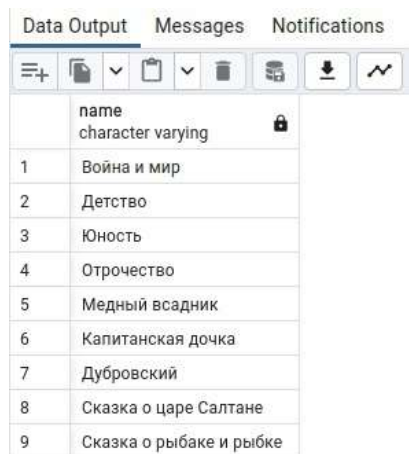
Рисунок 21 – Вывод запроса 18

19) Вывод всех доступных для аренды книг. Код запроса указан в «Листинге 30 – Запрос 19»;

Листинг 30 – Запрос 19

```
SELECT name  
FROM book  
WHERE available = TRUE;
```

Вывод запроса указан на «Рисунке 22 – Вывод запроса 19».



| | name character varying |
|---|---------------------------|
| 1 | Война и мир |
| 2 | Детство |
| 3 | Юность |
| 4 | Отрочество |
| 5 | Медный всадник |
| 6 | Капитанская дочка |
| 7 | Дубровский |
| 8 | Сказка о царе Салтане |
| 9 | Сказка о рыбаке и рыбке |

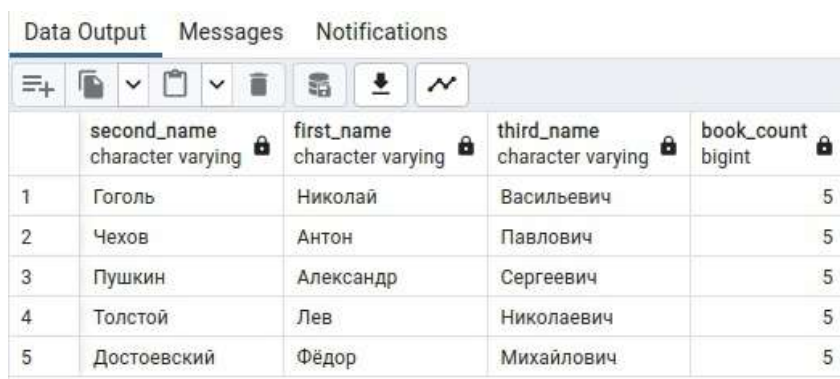
Рисунок 22 – Вывод запроса 19

20) Вывод количества книг для каждого автора. Код запроса указан в «Листинге 31 – Запрос 20»;

Листинг 31 – Запрос 20

```
SELECT second_name, first_name, third_name, COUNT(b.id) AS  
book_count  
FROM author  
JOIN book b ON author.id = b.author_id  
GROUP BY author.id, first_name, second_name;
```

Вывод запроса указан на «Рисунке 23 – Вывод запроса 20».



| | second_name character varying | first_name character varying | third_name character varying | book_count bigint |
|---|----------------------------------|---------------------------------|---------------------------------|----------------------|
| 1 | Гоголь | Николай | Васильевич | 5 |
| 2 | Чехов | Антон | Павлович | 5 |
| 3 | Пушкин | Александр | Сергеевич | 5 |
| 4 | Толстой | Лев | Николаевич | 5 |
| 5 | Достоевский | Фёдор | Михайлович | 5 |

Рисунок 23 – Вывод запроса 20

3.4 Тестирование функций и триггеров

Было проведено тестирование функций на новых наборах данных, был описан ожидаемый результат и фактический результат и сделан вывод о работоспособности функции.

1) Функция open_transaction (customer_contacts character varying (18), book_name character varying(100)).

Входные данные: «79123456789», «Тарас Бульба»

Ожидаемый результат: появление в таблице transaction новой записи с текущей датой (на момент проведения тестов 29.05.2023) в поле date_out, значения 22 в поле book_id, значения 1 в поле customer_id.

Фактический результат представлен на «Рисунке 24 – Тестирование функции open_transaction».

```
1 call open_transaction('79123456789','Тарас Бульба');
2 select * from transaction
```

| | id [PK] integer | customer_id integer | book_id integer | date_out date | date_in date | penalty real | total real |
|----|--------------------|------------------------|--------------------|------------------|-----------------|-----------------|---------------|
| 1 | 1 | 1 | 3 | 2023-01-10 | 2023-02-01 | 0 | 461 |
| 2 | 2 | 5 | 3 | 2023-02-15 | 2023-03-02 | 0 | 450 |
| 3 | 3 | 9 | 3 | 2023-04-01 | 2023-04-07 | 0 | 153 |
| 4 | 4 | 2 | 6 | 2023-01-11 | 2023-02-08 | 100 | 280 |
| 5 | 5 | 2 | 8 | 2023-01-16 | 2023-02-08 | 0 | 690 |
| 6 | 6 | 3 | 9 | 2023-01-18 | 2023-03-14 | 0 | 2280 |
| 7 | 7 | 3 | 25 | 2023-02-21 | 2023-02-26 | 200 | 450 |
| 8 | 8 | 4 | 1 | 2023-04-02 | 2023-05-08 | 300 | 360 |
| 9 | 9 | 10 | 1 | 2023-01-11 | 2023-02-08 | 0 | 280 |
| 10 | 10 | 6 | 20 | 2023-03-16 | 2023-03-24 | 0 | 280 |
| 11 | 11 | 7 | 21 | 2023-03-03 | 2023-03-23 | 0 | 180 |
| 12 | 12 | 5 | 7 | 2023-04-15 | 2023-05-01 | 0 | 320 |
| 13 | 13 | 6 | 14 | 2023-02-03 | 2023-02-22 | 50 | 582 |
| 14 | 14 | 7 | 13 | 2023-03-30 | 2023-04-01 | 600 | 672 |
| 15 | 17 | 10 | 2 | 2023-02-06 | [null] | 0 | [null] |
| 16 | 15 | 8 | 11 | 2023-04-08 | 2023-05-28 | 0 | 450 |
| 17 | 16 | 9 | 23 | 2023-05-11 | 2023-05-28 | 500 | 933 |
| 18 | 20 | 1 | 22 | 2023-05-29 | [null] | 0 | [null] |

Рисунок 24 – Тестирование функции open_transaction

Вывод о работе функции: функция работает исправно;

2) функция `close_transaction(customer_contacts character varying(18), book_name character varying(100))` и `close_transaction (customer_contacts character varying(18), book_name character varying(100), customer_penalty real, new_comment text))`.

Входные данные: воспользуемся уже открытыми транзакциями. Данные для возврата без штрафа: «79012345678», «Анна Каренина».

Ожидаемый результат: обновление записи с `id = 15`, должно появиться значение в столбце `date_in = 29.05.2023` – текущая дата на момент тестирования.

Результат тестирования приведен на «Рисунке 25 – Тестирование `close_transaction 1`».

```

1 call close_transaction('79012345678','Анна Каренина');
2 select * from transaction

```

| | id [PK] integer | customer_id integer | book_id integer | date_out date | date_in date | penalty real | total real |
|----|--------------------|------------------------|--------------------|------------------|-----------------|-----------------|---------------|
| 1 | 1 | 1 | 3 | 2023-01-10 | 2023-02-01 | 0 | 461 |
| 2 | 2 | 5 | 3 | 2023-02-15 | 2023-03-02 | 0 | 450 |
| 3 | 3 | 9 | 3 | 2023-04-01 | 2023-04-07 | 0 | 153 |
| 4 | 4 | 2 | 6 | 2023-01-11 | 2023-02-08 | 100 | 280 |
| 5 | 5 | 2 | 8 | 2023-01-16 | 2023-02-08 | 0 | 690 |
| 6 | 6 | 3 | 9 | 2023-01-18 | 2023-03-14 | 0 | 2280 |
| 7 | 7 | 3 | 25 | 2023-02-21 | 2023-02-26 | 200 | 450 |
| 8 | 8 | 4 | 1 | 2023-04-02 | 2023-05-08 | 300 | 360 |
| 9 | 9 | 10 | 1 | 2023-01-11 | 2023-02-08 | 0 | 280 |
| 10 | 10 | 6 | 20 | 2023-03-16 | 2023-03-24 | 0 | 280 |
| 11 | 11 | 7 | 21 | 2023-03-03 | 2023-03-23 | 0 | 180 |
| 12 | 12 | 5 | 7 | 2023-04-15 | 2023-05-01 | 0 | 320 |
| 13 | 13 | 6 | 14 | 2023-02-03 | 2023-02-22 | 50 | 582 |
| 14 | 14 | 7 | 13 | 2023-03-30 | 2023-04-01 | 600 | 672 |
| 15 | 15 | 8 | 11 | 2023-04-08 | 2023-05-28 | 0 | 450 |
| 16 | 16 | 9 | 23 | 2023-05-11 | 2023-05-28 | 500 | 933 |
| 17 | 20 | 1 | 22 | 2023-05-29 | [null] | 0 | [null] |
| 18 | 17 | 10 | 2 | 2023-02-06 | 2023-05-29 | 0 | 2260 |

Рисунок 25 – Тестирование `close_transaction 1`

Вывод о работе функции: функция работает исправно.

Данные для возврата книги со штрафом: «79123456789», «Тарас Бульба», 400, «Порвана обложка».

Ожидаемый результат: date_in = 2023-05-29, penalty = 400 в таблице transaction, comment = «Порвана обложка» в таблице book.

Фактический результат приведен в «Рисунке 26 – Тестирование close_transaction 2», «Рисунке 27 – Тестирование close_transaction 3».

```

1 call close_transaction('79123456789','Тарас Бульба', 400, 'Порвана обложка');
2 select * from transaction

```

| | id [PK] integer | customer_id integer | book_id integer | date_out date | date_in date | penalty real | total real |
|----|--------------------|------------------------|--------------------|------------------|-----------------|-----------------|---------------|
| 1 | 1 | 1 | 3 | 2023-01-10 | 2023-02-01 | 0 | 461 |
| 2 | 2 | 5 | 3 | 2023-02-15 | 2023-03-02 | 0 | 450 |
| 3 | 3 | 9 | 3 | 2023-04-01 | 2023-04-07 | 0 | 153 |
| 4 | 4 | 2 | 6 | 2023-01-11 | 2023-02-08 | 100 | 280 |
| 5 | 5 | 2 | 8 | 2023-01-16 | 2023-02-08 | 0 | 690 |
| 6 | 6 | 3 | 9 | 2023-01-18 | 2023-03-14 | 0 | 2280 |
| 7 | 7 | 3 | 25 | 2023-02-21 | 2023-02-26 | 200 | 450 |
| 8 | 8 | 4 | 1 | 2023-04-02 | 2023-05-08 | 300 | 360 |
| 9 | 9 | 10 | 1 | 2023-01-11 | 2023-02-08 | 0 | 280 |
| 10 | 10 | 6 | 20 | 2023-03-16 | 2023-03-24 | 0 | 280 |
| 11 | 11 | 7 | 21 | 2023-03-03 | 2023-03-23 | 0 | 180 |
| 12 | 12 | 5 | 7 | 2023-04-15 | 2023-05-01 | 0 | 320 |
| 13 | 13 | 6 | 14 | 2023-02-03 | 2023-02-22 | 50 | 582 |
| 14 | 14 | 7 | 13 | 2023-03-30 | 2023-04-01 | 600 | 672 |
| 15 | 15 | 8 | 11 | 2023-04-08 | 2023-05-28 | 0 | 450 |
| 16 | 16 | 9 | 23 | 2023-05-11 | 2023-05-28 | 500 | 933 |
| 17 | 17 | 10 | 2 | 2023-02-06 | 2023-05-29 | 0 | 2260 |
| 18 | 20 | 1 | 22 | 2023-05-29 | 2023-05-29 | 400 | 400 |

Рисунок 26 – Тестирование close_transaction 2

1

call close_transaction('79123456789','Тарас Бульба', 400, 'Порвана обложка');

2

select * from book

Data Output

Messages

Notifications

≡

+

📄

▼

📋

▼

🗑️

📤

📶

📶

| | id [PK] integer | name character varying | author_id integer | deposit real | cost_per_day real | comment text | available boolean |
|----|--------------------|---------------------------|----------------------|-----------------|----------------------|---------------------|----------------------|
| 1 | 1 | Война и мир | 1 | 1000 | 10 | Отличное состояние | true |
| 2 | 3 | Детство | 1 | 1500 | 30 | Новая книга | true |
| 3 | 4 | Юность | 1 | 800 | 40 | Слегка изношенная | true |
| 4 | 5 | Отрочество | 1 | 1200 | 50 | Идеальное состояние | true |
| 5 | 6 | Медный всадник | 2 | 1000 | 10 | Идеальное состояние | true |
| 6 | 7 | Капитанская дочка | 2 | 800 | 20 | Отличное состояние | true |
| 7 | 8 | Дубровский | 2 | 1500 | 30 | Хорошее состояние | true |
| 8 | 9 | Сказка о царе Салтане | 2 | 1200 | 40 | Новая книга | true |
| 9 | 10 | Сказка о рыбаке и рыбке | 2 | 1000 | 50 | Слегка изношенная | true |
| 10 | 12 | Братья Карамазовы | 3 | 500 | 20 | Хорошее состояние | true |
| 11 | 13 | Бесы | 3 | 1500 | 30 | Новая книга | true |
| 12 | 14 | Бедные люди | 3 | 800 | 40 | Слегка изношенная | true |
| 13 | 15 | Двойник | 3 | 1200 | 50 | Идеальное состояние | true |
| 14 | 16 | Рассказы | 4 | 1000 | 10 | Отличное состояние | true |
| 15 | 17 | Человек в футляре | 4 | 800 | 20 | Хорошее состояние | true |
| 16 | 18 | Хамелеон | 4 | 1500 | 30 | Новая книга | true |
| 17 | 19 | Дама с собачкой | 4 | 1200 | 40 | Слегка изношенная | true |
| 18 | 20 | Палата №6 | 4 | 1000 | 50 | Идеальное состояние | true |
| 19 | 21 | Мёртвые души | 5 | 1000 | 10 | Отличное состояние | true |
| 20 | 24 | Вий | 5 | 800 | 40 | Слегка изношенная | true |
| 21 | 25 | Нос | 5 | 1200 | 50 | Идеальное состояние | true |
| 22 | 11 | Преступление и наказание | 3 | 1000 | 10 | Отличное состояние | true |
| 23 | 23 | Ночь перед Рождеством | 5 | 1500 | 30 | Порвана обложка | true |
| 24 | 2 | Анна Каренина | 1 | 500 | 20 | Порвана обложка | true |
| 25 | 22 | Тарас Бульба | 5 | 500 | 20 | Порвана обложка | true |

Рисунок 27 – Тестирование close_transaction 3

Вывод о работе функции: функция работает исправно;

3) Функция contacts_formatter (contacts character varying(18)).

Входные данные: «71230001122»


Ожидаемый результат: выходные данные: +7 (123) 000-11-22

Фактический результат приведен в «Рисунке 28 – тестирование contacts_formatter»


1 select * from contacts_formatter('71230001122')

Data OutputMessagesNotifications


≡+





▼




▼









contacts_formatter
character varying 🔒

1+7 (123) 000-11-22

Рисунок 28 – тестирование contacts_formatter

Вывод о работе функции: функция работает исправно;

4) функция interval_counter (date_o date, date_i date).

Входные данные: «2023-01-01», «2023-01-03»

Ожидаемый результат: 2

Фактический результат приведен в «Рисунке 29 – Тестирование interval_counter 1»

1

select * from interval_counter('2023-01-01','2023-01-03')

Data Output

Messages

Notifications

≡+



▼



▼









interval_counter

integer



1

2

Рисунок 29 – Тестирование interval_counter 1

Входные данные: «2023-01-01», «2023-02-03»

Ожидаемый результат: 32

Фактический результат приведен в «Рисунке 30 – Тестирование interval_counter 2»



Рисунок 30 – Тестирование interval_counter 2

Вывод о работе функции: функция работает исправно;

5) триггер auto_category.

Для проверки работы триггера необходимо создать нового пользователя.

Входные данные: создание нового пользователя с полями «Ушаков», «Матвей», «Викторович», «ул. Прянишникова, 2А», «79990001234».

Ожидаемый результат: создание новой записи в таблице customer_category под id = 11.

Фактический результат приведен в «Рисунке 31 – Тестирование auto_category».

| | |
|---|---|
| 1 | INSERT INTO CUSTOMER |
| 2 | (second_name, first_name, third_name, address, contacts) |
| 3 | VALUES |
| 4 | ('Ушаков', 'Матвей', 'Викторович', 'ул. Прянишникова, 2А', '799 000-12-34') |
| 5 | select * from customer_category |

| Data Output | Messages | Notifications |
|--|------------------------|------------------------|
| <div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🔄</div> <div>⬇️</div> <div>📈</div> </div> | | |
| | customer_id integer | category_id integer |
| 1 | 1 | 3 |
| 2 | 2 | 4 |
| 3 | 3 | 4 |
| 4 | 4 | 4 |
| 5 | 5 | 4 |
| 6 | 6 | 3 |
| 7 | 7 | 2 |
| 8 | 8 | 2 |
| 9 | 9 | 1 |
| 10 | 10 | 4 |
| 11 | 11 | 4 |

Рисунок 31 – Тестирование auto_category

Вывод о работе триггера: триггер работает корректно;

б) триггер auto_contacts_formatter.

При проверке предыдущей функции был создан новый пользователь, при добавлении уже сработала функция. Проверим запись в таблице customer.

Ожидаемый результат: +7 (999) 000-12-34 в поле contacts у нового клиента с фамилией Ушаков.

Фактический результат приведен в «Рисунке 32 – Тестирование auto_contacts_formatter».

| | |
|---|--|
| 1 | INSERT INTO CUSTOMER |
| 2 | (second_name, first_name, third_name, address, contacts) |
| 3 | VALUES |
| 4 | ('Ушаков', 'Матвей', 'Викторович', 'ул. Прянишникова, 2А', '79990001234'); |
| 5 | select * from customer |

| Data Output Messages Notifications | | | | | | |
|------------------------------------|--------------|-------------------|-------------------|-------------------|----------------------|--------------------|
| | id | second_name | first_name | third_name | address | contacts |
| | [PK] integer | character varying | character varying | character varying | character varying | character varying |
| 1 | 1 | Иванов | Иван | Иванович | ул. Ленина, 10 | +7 (912) 345-67-89 |
| 2 | 2 | Петров | Петр | Петрович | ул. Пушкина, 5 | +7 (923) 456-78-90 |
| 3 | 3 | Сидорова | Анна | Викторовна | пр. Гагарина, 20 | +7 (934) 567-89-01 |
| 4 | 4 | Смирнов | Алексей | Игоревич | ул. Московская, 15 | +7 (945) 678-90-12 |
| 5 | 5 | Козлова | Мария | Сергеевна | пр. Ленинский, 30 | +7 (956) 789-01-23 |
| 6 | 6 | Николаев | Дмитрий | Анатольевич | ул. Советская, 25 | +7 (967) 890-12-34 |
| 7 | 7 | Васильева | Екатерина | Александровна | пр. Победы, 12 | +7 (978) 901-23-45 |
| 8 | 8 | Морозов | Игорь | Владимирович | ул. Зеленая, 8 | +7 (989) 012-34-56 |
| 9 | 9 | Федоров | Артем | Олегович | ул. Пролетарская, 18 | +7 (990) 123-45-67 |
| 10 | 10 | Кузнецова | Анастасия | Павловна | пр. Свердлова, 7 | +7 (901) 234-56-78 |
| 11 | 11 | Ушаков | Матвей | Викторович | ул. Прянишникова, 2А | +7 (999) 000-12-34 |

Рисунок 32 – Тестирование auto_contacts_formatter

Вывод о работе триггера: триггер успешно выполнил свою работу;

7) триггер auto_unique_contacts_checker.

Входные данные: создание нового пользователя с полями «Ушаков», «Матвей», «Викторович», «ул. Прянишникова, 2А», «79990001234», повторяющие поля существующего пользователя.

Ожидаемый результат: ошибка создания нового пользователя.

Фактический результат приведен в «Рисунке 33 – Тестирование auto_unique_contacts_checker».

| | |
|---|--|
| 1 | INSERT INTO CUSTOMER |
| 2 | (second_name, first_name, third_name, address, contacts) |
| 3 | VALUES |
| 4 | ('Ушаков', 'Матвей', 'Викторович', 'ул. Прянишникова, 2А', '79990001234'); |

| Data Output Messages Notifications | | |
|--|--|--|
| ERROR: Уже есть клиент с таким номером телефона! | | |
| CONTEXT: PL/pgSQL function unique_contacts_checker() line 4 at RAISE | | |
| SQL state: P0001 | | |

Рисунок 33 – Тестирование auto_unique_contacts_checker

Вывод о работе триггера: триггер работает;

8) триггер book_status_changer.

Входные данные: создание новой транзакции с книгой «Мёртвые души».

Ожидаемый результат: смена значения в поле available на FALSE у записи книги «Мёртвые души».

Фактический результат приведен в «Рисунке 34 – Тестирование book_status_changer 1»

```
1 call open_transaction('79990001234','Мёртвые души');
2 select * from book
```

| | Id [PK] integer | name character varying | author_id integer | deposit real | cost_per_day real | comment text | available boolean |
|----|--------------------|---------------------------|----------------------|-----------------|----------------------|---------------------|----------------------|
| 1 | 1 | Война и мир | 1 | 1000 | 10 | Отличное состояние | true |
| 2 | 3 | Детство | 1 | 1500 | 30 | Новая книга | true |
| 3 | 4 | Юность | 1 | 800 | 40 | Слегка изношенная | true |
| 4 | 5 | Отрочество | 1 | 1200 | 50 | Идеальное состояние | true |
| 5 | 6 | Медный всадник | 2 | 1000 | 10 | Идеальное состояние | true |
| 6 | 7 | Капитанская дочка | 2 | 800 | 20 | Отличное состояние | true |
| 7 | 8 | Дубровский | 2 | 1500 | 30 | Хорошее состояние | true |
| 8 | 9 | Сказка о царе Салтане | 2 | 1200 | 40 | Новая книга | true |
| 9 | 10 | Сказка о рыбаке и рыбке | 2 | 1000 | 50 | Слегка изношенная | true |
| 10 | 12 | Братья Карамазовы | 3 | 500 | 20 | Хорошее состояние | true |
| 11 | 13 | Бесы | 3 | 1500 | 30 | Новая книга | true |
| 12 | 14 | Бедные люди | 3 | 800 | 40 | Слегка изношенная | true |
| 13 | 15 | Двойник | 3 | 1200 | 50 | Идеальное состояние | true |
| 14 | 16 | Рассказы | 4 | 1000 | 10 | Отличное состояние | true |
| 15 | 17 | Человек в футляре | 4 | 800 | 20 | Хорошее состояние | true |
| 16 | 18 | Хамелеон | 4 | 1500 | 30 | Новая книга | true |
| 17 | 19 | Дама с собачкой | 4 | 1200 | 40 | Слегка изношенная | true |
| 18 | 20 | Палата №6 | 4 | 1000 | 50 | Идеальное состояние | true |
| 19 | 24 | Вий | 5 | 800 | 40 | Слегка изношенная | true |
| 20 | 25 | Нос | 5 | 1200 | 50 | Идеальное состояние | true |
| 21 | 11 | Преступление и наказание | 3 | 1000 | 10 | Отличное состояние | true |
| 22 | 23 | Ночь перед Рождеством | 5 | 1500 | 30 | Порвана обложка | true |
| 23 | 2 | Анна Каренина | 1 | 500 | 20 | Порвана обложка | true |
| 24 | 22 | Тарас Бульба | 5 | 500 | 20 | Порвана обложка | true |
| 25 | 21 | Мёртвые души | 5 | 1000 | 10 | Отличное состояние | false |

Рисунок 34 – Тестирование book_status_changer 1

Входные данные: закрытие транзакции с книгой «Мёртвые души».

Ожидаемый результат: смена значения в поле available на FALSE у записи книги «Мёртвые души».

Фактический результат приведен в «Рисунке 35 – Тестирование book_status_changer 2»

1

call close_transaction('79990001234', 'Мёртвые души');

2

select * from book

Data Output

Messages

Notifications

id

[PK] integer

name

character varying

author_id

integer

deposit

real

cost_per_day

real

comment

text

available

boolean

1

1

Война и мир

1

1000

10

Отличное состояние

true

2

3

Детство

1

1500

30

Новая книга

true

3

4

Юность

1

800

40

Слегка изношенная

true

4

5

Отрочество

1

1200

50

Идеальное состояние

true

5

6

Медный всадник

2

1000

10

Идеальное состояние

true

6

7

Капитанская дочка

2

800

20

Отличное состояние

true

7

8

Дубровский

2

1500

30

Хорошее состояние

true

8

9

Сказка о царе Салтане

2

1200

40

Новая книга

true

9

10

Сказка о рыбаке и рыбке

2

1000

50

Слегка изношенная

true

10

12

Братья Карамазовы

3

500

20

Хорошее состояние

true

11

13

Бесы

3

1500

30

Новая книга

true

12

14

Бедные люди

3

800

40

Слегка изношенная

true

13

15

Двойник

3

1200

50

Идеальное состояние

true

14

16

Рассказы

4

1000

10

Отличное состояние

true

15

17

Человек в футляре

4

800

20

Хорошее состояние

true

16

18

Хамелеон

4

1500

30

Новая книга

true

17

19

Дама с собачкой

4

1200

40

Слегка изношенная

true

18

20

Палата №6

4

1000

50

Идеальное состояние

true

19

24

Вий

5

800

40

Слегка изношенная

true

20

25

Нос

5

1200

50

Идеальное состояние

true

21

11

Преступление и наказание

3

1000

10

Отличное состояние

true

22

23

Ночь перед Рождеством

5

1500

30

Порвана обложка

true

23

2

Анна Каренина

1

500

20

Порвана обложка

true

24

22

Тарас Бульба

5

500

20

Порвана обложка

true

25

21

Мёртвые души

5

1000

10

Отличное состояние

true

Рисунок 35 – Тестирование book_status_changer 2

Вывод о работе триггера: триггер работает корректно;

9) триггер total_counter.

Для проверки работы триггера создадим новую транзакцию, при закрытии транзакции должно появиться сообщение со стоимостью проката.

Входные данные: вызов функции open_transaction('79990001234', 'Мёртвые души'), close_transaction('79990001234', 'Мёртвые души', 1000, 'Тестирование').

Ожидаемый результат: вывод сообщения «Итоговая стоимость составит: 1000 рублей».

Фактический результат приведен в «Рисунке 36 – Тестирование total_counter».


```
1 call close_transaction('79990001234','Мёртвые души',1000,'Тестирование');
```

| Data Output | Messages | Notifications |
|--|----------|---------------|
| NOTICE: Итоговая стоимость составит: 1000 рублей | | |
| CALL | | |
| Query returned successfully in 29 msec. | | |

Рисунок 36 – Тестирование total_counter

Вывод о работе триггера: триггер работает корректно.

ЗАКЛЮЧЕНИЕ

В ходе данной курсовой работы была создана база данных с использованием системы управления базами данных PostgreSQL. Этот проект позволил мне приобрести ценные навыки в проектировании и разработке баз данных, а также в освоении SQL-запросов и работе с конкретной СУБД.

В процессе работы я изучил основные принципы моделирования данных, определил структуру базы данных, создал таблицы, определил связи между ними и заполнил их данными. Я также изучил различные возможности PostgreSQL, такие как создание и использование функций, триггеров и процедур.

Создание этой базы данных помогло мне лучше понять важность эффективной организации и хранения данных для успешного функционирования информационной системы. Я осознал значимость грамотного проектирования базы данных для обеспечения эффективности и масштабируемости системы, а также для обеспечения целостности и безопасности данных.

В процессе работы с PostgreSQL я также столкнулся с некоторыми трудностями и проблемами, но благодаря глубокому изучению документации, я смог успешно их преодолеть и получить ценный опыт.

В целом, данная курсовая работа позволила мне углубить свои знания и навыки в области баз данных и PostgreSQL в частности. Я приобрел опыт проектирования, разработки и оптимизации баз данных, что будет полезным для моей дальнейшей карьеры в области информационных технологий. PostgreSQL является мощной и гибкой СУБД, и я уверен, что полученные мною знания и навыки будут применяться мной в будущих проектах для создания надежных и эффективных баз данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.201-78. Режим доступа: <https://www.swrit.ru/doc/espd/19.201-78.pdf> (дата обращения: 18.04.2023).
2. Курс в ЛМС Московского политеха «Разработка технических текстов и документации». Режим доступа: <https://online.mospolytech.ru/course/view.php?id=1689> (дата обращения: 18.04.2023).
3. Официальная документация разработчика PostgreSQL. Режим доступа: <https://www.postgresql.org/> (дата обращения: 29.05.2023).
4. Статья «Изучаем PostgreSQL. Часть 1. Знакомимся с архитектурой». Режим доступа: <https://habr.com/ru/companies/otus/articles/706346/> (дата обращения: 29.05.2023).
5. Информационный портал «Habr». Режим доступа: <https://habr.com/ru/all/> (дата обращения: 29.05.2023).
6. ГОСТ 7.32-2017. Режим доступа: <https://files.stroyinf.ru/Data2/1/4293742/4293742537.pdf> (дата обращения: 29.05.2023).
7. «Оптимизация запросов PostgreSQL» — Г. Домбровская, Б. Новиков, А. Бейликова. Режим доступа: <https://dmkpress.com/files/PDF/978-5-97060-963-7.pdf> (дата обращения 29.05.2023).
8. «Основы технологий баз данных: учебное пособие» — Б. Новиков, Е. Горшкова, Н. Графеева. Режим доступа: <https://postgrespro.ru/education/books/dbtech> (дата обращения: 29.05.2023).
9. «PostgreSQL изнутри» — Е. Рогов. Режим доступа: <https://postgrespro.ru/education/books/internals> (дата обращения 29.05.2023).
10. «Postgres: первое знакомство» — П. Лузанов, Е. Рогов, И. Лёвшин. Режим доступа: <https://postgrespro.ru/education/books/introbook> (дата обращения 29.05.2023).

11. «PostgreSQL. Основы языка SQL» — Е. Моргунов. Режим доступа: <https://postgrespro.ru/education/books/sqlprimer> (дата обращения 29.05.2023).
12. Руководство по подготовке курсовых работ (проектов) и выпускных квалификационных работ. Режим доступа: http://www.skf-mtusi.ru/files/vkr/MU_DP_KP_MTUSI.pdf (дата обращения 29.05.2023)
13. Правила оформления курсовых и дипломных работ. Режим доступа: <https://studfile.net/preview/3009968/page:5/> (дата обращения 29.05.2023).
14. Правила оформления отчета к лабораторным и курсовым работам. Режим доступа: <https://publications.hse.ru/mirror/pubs/share/direct/227003831> (дата обращения 29.05.2023).
15. Статья «Моделирование данных: обзор». Режим доступа <https://habr.com/ru/articles/556790/> (дата обращения 29.05.2023).
16. Нотация Питера Чена. Режим доступа: https://studme.org/77222/informatika/notatsiya_pitera_chena (дата обращения 29.05.2023).
17. Презентация «Модель «сущность-связь». Режим доступа: <https://foreva.susu.ru/courses/db/lecture3.pdf> (дата обращения 29.05.2023)
18. Триггерные процедуры в PostgreSQL. Режим доступа: <https://postgrespro.ru/docs/postgresql/9.6/plpgsql-trigger> (дата обращения 29.05.2023)
19. Форум по PostgreSQL Wiki.postgresql. Режим доступа: https://wiki.postgresql.org/wiki/Main_Page/ru (дата обращения 29.05.2023).