 **ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ТОМСКИЙ ЭКОНОМИКО-ПРОМЫШЛЕННЫЙ КОЛЛЕДЖ»**

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ**

**«Онлайн библиотека комиксов»**

***ОТЧЁТ***

*СПЕЦИАЛЬНОСТЬ 09.02.07 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ*

Выполнил:

Студент группы 1912с

\_\_\_\_\_\_\_\_\_ / Малинин Е.М.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2024г.

Проверил:

Преподаватель ИТ

\_\_\_\_\_\_\_\_\_\_ / Летягина И.Е.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024г.

Томск 2024

**ОГЛАВЛЕНИЕ**

[ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#__RefHeading___Toc7804_850649020)

[АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#__RefHeading___Toc7806_850649020)

[ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ В DRAW.IO 15](#__RefHeading___Toc7808_850649020)

[СЛОВАРЬ ДАННЫХ 23](#__RefHeading___Toc7810_850649020)

[РАЗРАБОТКА БАЗЫ ДАННЫХ В PGADMIN4 30](#__RefHeading___Toc7812_850649020)

[Создание связей между таблицами 30](#__RefHeading___Toc7814_850649020)

[ЗАПОЛНЕНИЕ ТАБЛИЦ ДАННЫМИ 36](#__RefHeading___Toc7816_850649020)

[Таблица «acquiring» 36](#__RefHeading___Toc7818_850649020)

[Таблица «auths» 36](#__RefHeading___Toc7820_850649020)

[Таблица «avatars» 37](#__RefHeading___Toc7822_850649020)

[Таблица «bans» 38](#__RefHeading___Toc7824_850649020)

[Таблица «chapter\_review\_likes» 39](#__RefHeading___Toc7826_850649020)

[Таблица «chapter\_reviews» 39](#__RefHeading___Toc7828_850649020)

[Таблица «chapters» 40](#__RefHeading___Toc7830_850649020)

[Таблица «comic\_genres» 41](#__RefHeading___Toc7830_850649020_%2525D0)

[Таблица «comic\_names» 42](#__RefHeading___Toc7830_850649020_%2525D1)

[Таблица «comic\_ratings» 45](#__RefHeading___Toc7830_850649020_%2525D2)

[Таблица «comic\_reading\_statuses» 45](#__RefHeading___Toc7830_850649020_%2525D3)

[Таблица «comic\_reviews» 46](#__RefHeading___Toc7830_850649020_%2525D4)

[Таблица «comic\_types» 47](#__RefHeading___Toc7830_850649020_%2525D5)

[Таблица «comics» 47](#__RefHeading___Toc7830_850649020_%2525D6)

[Таблица «frames» 48](#__RefHeading___Toc7830_850649020_%2525D7)

[Таблица «genres» 49](#__RefHeading___Toc7830_850649020_%2525D8)

[Таблица «like\_chapters» 50](#__RefHeading___Toc7830_850649020_%2525D9)

[Таблица «reading\_statuses» 51](#__RefHeading___Toc7830_850649020_%2525Da)

[Таблица «release\_statuses» 51](#__RefHeading___Toc7830_850649020_%2525Db)

[Таблица «roles» 52](#__RefHeading___Toc7830_850649020_%2525Dc)

[Таблица «subscription\_prices» 52](#__RefHeading___Toc7830_850649020_%2525Dd)

[Таблица «subscriptions» 53](#__RefHeading___Toc7830_850649020_%2525De)

[Таблица «translate\_statuses» 54](#__RefHeading___Toc7830_850649020_%2525Df)

[Таблица «user\_subscriptions» 54](#__RefHeading___Toc7830_850649020_%2525Dg)

[Таблица «users» 55](#__RefHeading___Toc7830_850649020_%2525Dh)

[SQL-код создания базы данных 57](#__RefHeading___Toc7812_850649020_%2525D0)

[Заключение 101](#__RefHeading___Toc12130_1266828952)

[Источники 102](#__RefHeading___Toc12132_1266828952)

[Интернет-ресурсы: 102](#__RefHeading___Toc12134_1266828952)

[Литература: 102](#__RefHeading___Toc12136_1266828952)

# ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

На платформе предоставляется возможность пользователям читать комиксы (при наличии подписки) различных жанров, а также взаимодействовать с другими пользователями через отзывы и обзоры. Пользователи могут оценивать комиксы, ставить лайки отзывам и частям комиксов (главам), а также подписываться на доступ к чтению комиксов.

В базе данных должна храниться информация о:

* ПОЛЬЗОВАТЕЛЯХ: никнейм, email, хешированный пароль, роль и адрес на аватарку;
* БЛОКИРОВКАХ пользователей (банах): причина, дата начала действия, дата окончания действия;
* АВАТАРАХ: путь к изображению аватара пользователя;
* КОМИКСАХ: описание, год выпуска, возрастное ограничение, путь к постеру, тип комикса, создатели, статусы выпуска и перевода;
* ЖАНРАХ: наименование жанра;
* ГЛАВАХ: наименование главы, её порядковый номер, номер тома и дата добавления;
* ФРЕЙМАХ (кадрах в главе): порядковый номер в главе, а также путь до изображения;
* КОММЕНТАРИЯХ на комиксы и главы: текст отзыва, дата обновления, статус модерации;
* ЛАЙКАХ на главы и комментариях к главам;
* ОЦЕНКАХ к главам для составления рейтинга;
* СТАТУСАХ чтения пользователя;
* ПОДПИСКАХ пользователей для чтения глав.

При проектировании базы данных необходимо учитывать следующее:

* каждый пользователь имеет уникальный никнейм и email;
* у комиксов может быть несколько авторов, в том числе и изданий, либо не быть ни одного;
* у подписок должна сохраняться история цен;
* пользователь может поставить оценку на комикс только 1 раз;
* пользователь может хранить только 1 статус чтения для 1 комикса.

*Цель* разработки базы данных для платформы чтения комиксов с возможностью социального взаимодействия пользователей и анализа их активности - обеспечить эффективное хранение, обработку и предоставление информации о пользователях, их действиях и взаимодействиях на платформе.

На основе поставленной цели были выдвинуты следующие *задачи*:

1. Создание структуры базы данных для хранения информации о пользователях, комиксах, активностях и взаимодействиях на платформе.

2. Определение необходимых таблиц, полей и связей для эффективного хранения данных о пользователях, их профилях и действий на платформе.

3. Разработка механизмов для обеспечения безопасности и конфиденциальности данных пользователей на платформе.

5. Обеспечение возможности взаимодействия пользователей между собой через комментарии, оценки и обмен рекомендациями (лайки).

6. Разработка системы отчетов и аналитики для предоставления информации о активности пользователей и эффективности платформы.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

При анализе предметной области были выделены следующие сущности:

* *Пользователи*: хранят свой код, никнейм, email, пароль (в виде хеша), роль и аватар.

Пример данных:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код | Никнейм | email | пароль | Код роли | Код аватара |
| 1 | admin | admin@mail.ru | $2y$12$8g/Pco0bCqvVrk1MmoDYW.YDdwtg1dUCMJh57.GO1O3YOwtKuKKSe | 1 | 1 |
| 2 | chebyrek | chebyrek@mail.ru | $2y$12$p6Ntkzd.DJGHzGWqu6B0HOCh/TswDs7RYuHvl2ICDkM6Ixn5lH0Rm | 2 | 2 |
| 3 | odyvan | odyvan@mail.ru | $2y$12$faqn3bVSThe6y5j2xqWmbe58.CsKZ98EbsJl7H7MqL0tRsFYyDUMi | 2 | 3 |
| 4 | bannedUser | bannedUser@mail.ru | $2y$12$dnScfS6mz.3YEbjSr.v0ceYy8w0Qtcl1w4LISlwbRrXuqGrW1yE6. | 2 | 4 |

* *Блокировки*: хранят свой номер, дату начала действия, дата конца действия, причину блокировки.

Пример данных:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код | Дата начала действия | Дата конца действия | Причина | Код пользователя |
| 1 | 05-10-2024 | 06-10-2024 | Использование стороннего ПО | 4 |

* *Аватарки*: хранят свой номер карточки и путь до файла изображения (относительно каталога хранения).

Пример данных:

|  |  |
| --- | --- |
| Код | Путь до изображения |
| 1 | Default.jpg |
| 2 | 1.jpg |

* *Авторизации*: хранят токен авторизации и пользователя, которому она принадлежит.

Пример данных:

|  |  |
| --- | --- |
| Токен | Код пользователя |
| IG5icnRlbWJ2bS1ibHQ= | 1 |
| bWtucnRqZ25yZXZuIHI= | 2 |
| LG5iIGtrdHJl2a3RybA= | 3 |

* *Роли*: хранят коды роли пользователя.

Пример данных:

|  |  |
| --- | --- |
| Ключ | Код |
| 1 | admin |
| 2 | user |

* *Комиксы*: содержат описание, год выпуска, постер, возрастное ограничение, тип комикса, статус выпуска и перевода.

Пример данных:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ключ | Описание | Год выпуска | Постер | Возрастное ограничение | Код типа комикса | Код статуса выпуска | Код статуса перевода |
| 1 | В далеком будущем, населенном технологиями и роботами, молодой инженер ведет исследования в области искусственного интеллекта... | 2018 | https://remanga.org/media/titles/sweet\_home\_hwang\_youngchan/8526d40b663441e5189f6263cefca146.jpg | 16+ | 2 | 1 | 1 |
| 2 | В мире, раздираемом войной между ангелами и демонами, группа обычных людей оказывается втянутой в борьбу за выживание... | 2022 | https://example.com/media/titles/angel\_demon\_war/poster.jpg | 16+ | 1 | 1 | 1 |

* *Тип комикса*: содержит код и наименование типа.

Пример данных:

|  |  |
| --- | --- |
| Ключ | Наименование |
| 1 | Манга |
| 2 | Манхва |
| 3 | Маньхуа |
| 4 | Западный комикс |
| 5 | Рукомикс |
| 6 | Индонезийский комикс |
| 7 | Другое |

* *Статус выпуска*: содержит код и наименование статуса.

Пример данных:

|  |  |
| --- | --- |
| Ключ | Наименование |
| 1 | Анонсировано |
| 2 | Выпускается |
| 3 | Завершено |
| 4 | Неизвестно |

* *Статус перевода*: содержит код и наименование статуса.

Пример данных:

|  |  |
| --- | --- |
| Ключ | Наименование |
| 1 | Ожидает перевода |
| 2 | Переводится |
| 3 | Переведено |
| 4 | Приостановлено |

* *Жанр комикса*: содержит код комикса и жанра.

Пример данных:

|  |  |
| --- | --- |
| Код комикса | Код жанра |
| 1 | 1 |
| 1 | 2 |
| 1 | 5 |
| 4 | 4 |

* *Жанр*: содержит код и наименование жанра.

Пример данных:

|  |  |
| --- | --- |
| Код | Наименование |
| 1 | Экшен |
| 2 | Драма |
| 3 | Фэнтези |
| 4 | Романтика |

* *Роль создателя/издателя*: содержит код и наименование роли.

Пример данных:

|  |  |
| --- | --- |
| Код | Наименование |
| 1 | Издатель |
| 2 | Автор |
| 3 | Художник |

* *Издатель/создатель*: содержит код, наименование создателя и его роль.

Пример данных:

|  |  |  |
| --- | --- | --- |
| Код | Наименование | Код роли создателя |
| 1 | Webtoon | 1 |
| 2 | Naver | 1 |
| 3 | Carnby | 2 |

* *Создатель комикса*: содержит код комикса и издателя.

Пример данных:

|  |  |
| --- | --- |
| Код комикса | Код создателя |
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 2 | 1 |

* *Отзыв комикса*: содержит свой код, код комикса, код пользователя, текст отзыва, дату и время последнего обновления, статус модерации.

Пример данных:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код | Код комикса | Код пользователя | Текст отзыва | Дата и время обновления | Статус модерации |
| 1 | 1 | 1 | Сюжет необычный, но сами персонажи не раскрыты. На один раз прочитать сойдёт | 2024-06-20 00:00:00.000 | 1 |
| 2 | 2 | 2 | Отличная атмосфера и увлекательный сюжет. Персонажи живые, каждый со своей историей. Рекомендую всем! | 2024-06-20 00:00:00.000 | 0 |
| 3 | 3 | 3 | Необычные идеи и интересные повороты событий, но слишком много диалогов. Визуальная составляющая тоже важна | 2024-06-21 00:00:00.000 | 1 |
| 4 | 4 | 4 | Фантастический мир и глубокие персонажи. Читал на одном дыхании! | 2024-06-21 00:00:00.000 | 1 |

* *Статус чтения*: содержит свой код и наименование статуса.

Пример данных:

|  |  |
| --- | --- |
| Код | Наименование |
| 1 | Читаю |
| 2 | Отложено |
| 3 | Брошено |
| 4 | Прочитано |

* *Статус чтения комикса*: содержит код пользователя, комикса и статуса чтения.

Пример данных:

|  |  |  |
| --- | --- | --- |
| Код пользователя | Код комикса | Код статуса чтения |
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 2 |
| 4 | 2 | 1 |

* *Оценка комикса*: содержит код пользователя, комикса и значение оценки.

Пример данных:

|  |  |  |
| --- | --- | --- |
| Код пользователя | Код комикса | Оценка |
| 1 | 1 | 7 |
| 2 | 1 | 8 |
| 3 | 1 | 5 |
| 4 | 2 | 10 |

* *Глава комикса*: содержит свой код, наименование главы, её порядковый номер, номер тома, дату добавления.

Пример данных:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код | Наименование | Порядковый номер | Номер тома | Дата добавления |
| 1 | Начало | 1 | 1 | 2024-06-01 |
| 2 | Первый подвиг | 2 | 1 | 2024-06-05 |
| 3 | Встреча с противником | 3 | 1 | 2024-06-10 |
| 4 | Встреча с предателем | 1 | 1 | 2024-06-20 |

* *Фрейм*: содержит свой код, код главы, путь до изображения (относительно каталога хранения) и порядковый номер.

Пример данных:

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Код главы | Путь до изображения | Порядковый номер |
| 1 | 1 | knrthitenb.jpg | 1 |
| 2 | 1 | owekjfintb.jpg | 2 |
| 3 | 1 | ierhbndsmv.jpg | 3 |
| 4 | 2 | krtjgkerskmvd.jpg | 1 |

* *Комментарий главы*: содержит свой код, код главы, код пользователя, текст отзыва, дату и время последнего обновления, статус модерации.

Пример данных:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код | Код главы | Код пользователя | Текст отзыва | Дата и время обновления | Статус модерации |
| 1 | 1 | 1 | Ого, какой поворот сюжета! Никогда не думал, что она так поступит. | 2024-06-20 00:00:00.000 | 1 |
| 2 | 2 | 2 | Этот момент заставил меня задуматься над собственными поступками. Очень сильно. | 2024-06-20 00:00:00.000 | 0 |
| 3 | 3 | 3 | Необычные идеи и интересные повороты событий, но слишком много диалогов. Визуальная составляющая тоже важнаВот это я называю персонаж с характером! Его поступки всегда удивляют и вдохновляют. | 2024-06-21 00:00:00.000 | 1 |
| 4 | 4 | 4 | Этот момент заслуживает овации. Автору удалось создать настоящую эмоциональную бурю. | 2024-06-21 00:00:00.000 | 1 |

* *Лайк комментария главы*: содержит код комментария и пользователя.

Пример данных:

|  |  |
| --- | --- |
| Код комментария | Код пользователя |
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 4 | 1 |

# ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ В DRAW.IO

Цель данной работы – создание базы данных, отвечающей требованиям и предназначенной для хранения определенной информации. Для достижения этой цели были реализованы следующие этапы:

1. Исследовательский этап.

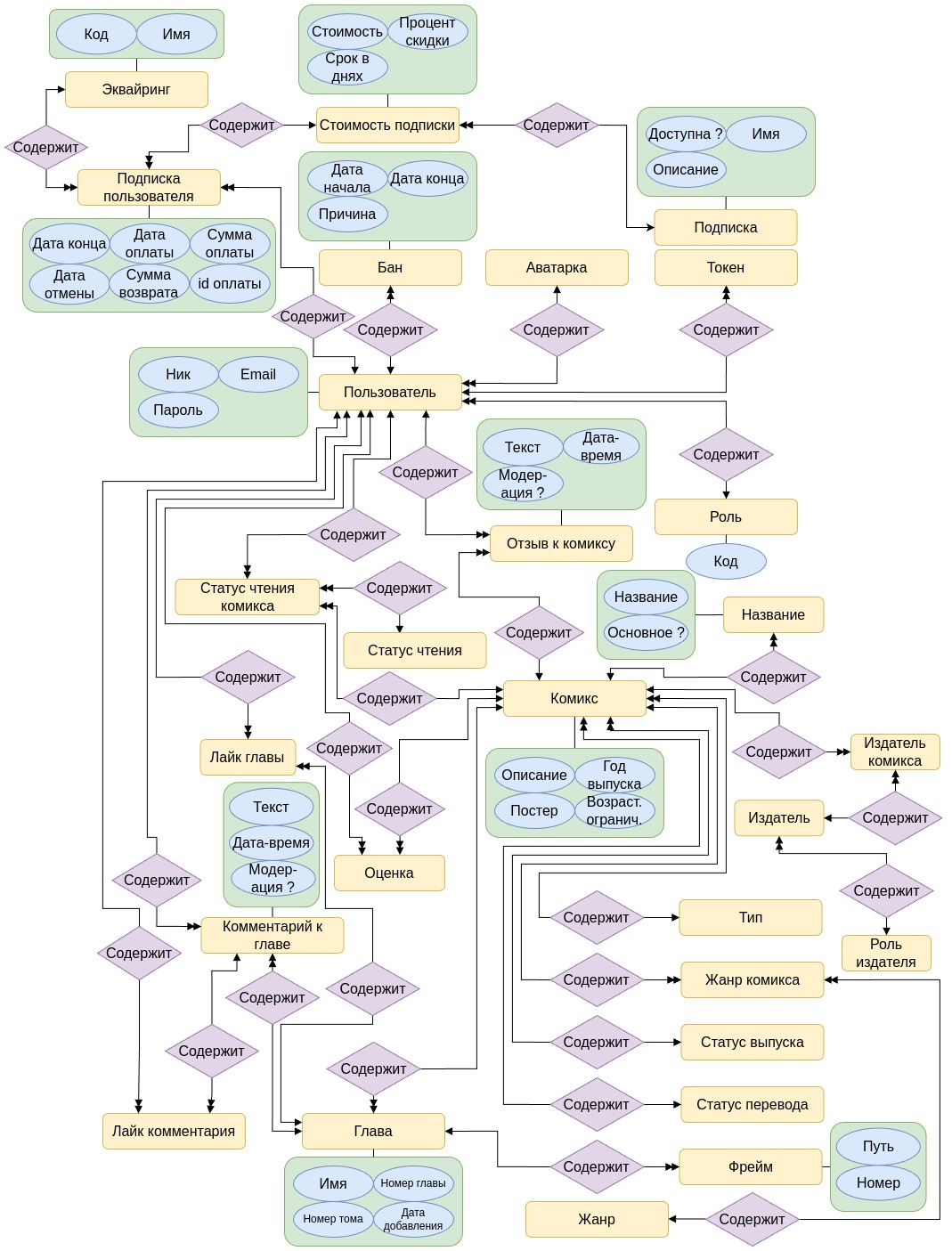
На данном этапе были проведены все необходимые исследования, собраны требования и определено, какую именно информацию необходимо хранить. После этого были выделены основные элементы, которые представляют информацию (сущности), и определены атрибуты для каждой из этих сущностей. В результате этого этапа мы получили концептуальную модель, которая представляет собой абстрактное описание структуры системы, свойств ее элементов и причинно-следственных связей, важных для достижения поставленной цели моделирования (см. рис. 1).

Рисунок 1 – Концептуальная модель базы данных

1. Трансформация концептуальной модели.

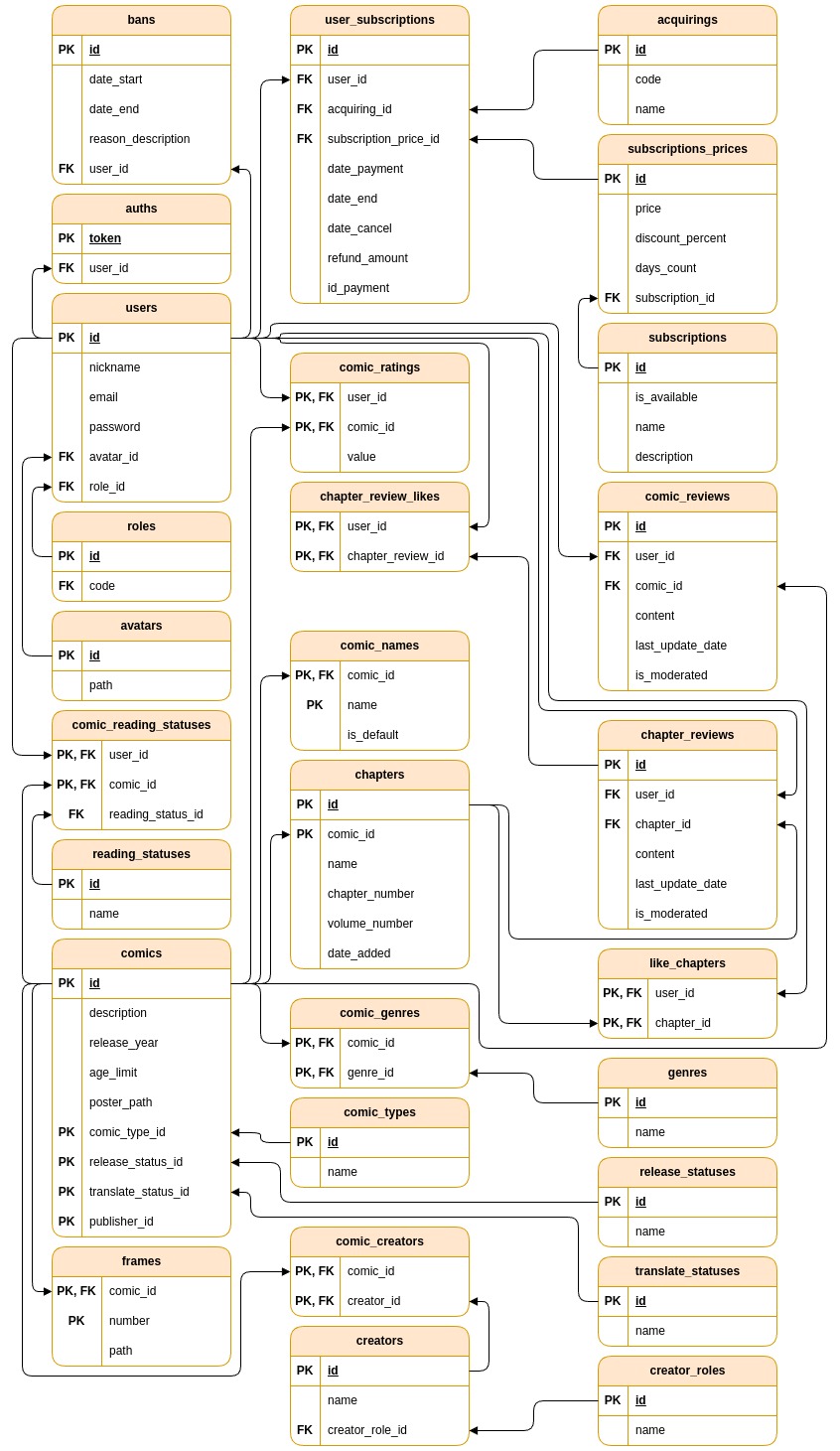
На втором этапе мы преобразовали концептуальную модель в реляционную модель данных. Были определены таблицы, их атрибуты и соответствующие типы данных для каждого атрибута. Также была проведена нормализация данных, то есть разделение их на отдельные таблицы и установление связей между ними (см. рис. 2).

Рисунок 2 – Логическая модель базы данных

1. На заключительном этапе было решено как данные будут храниться на физическом уровне, определено предполагаемое количество данных и выделены соответствующие ресурсы.

С помощью инструмента draw.io были заработаны схемы таблиц *физической модели* – это реализация логической модели данных, создаваемая администраторами и разработчиками баз данных, которая разрабатывается для определённой СУБД, технологии хранения и соединителей данных (см. рис. 3, 4, 5), а также были созданы физические таблицы и индексы в СУБД на основе созданной физической диаграммы базы данных, отображающей организацию данных на самом низком уровне.

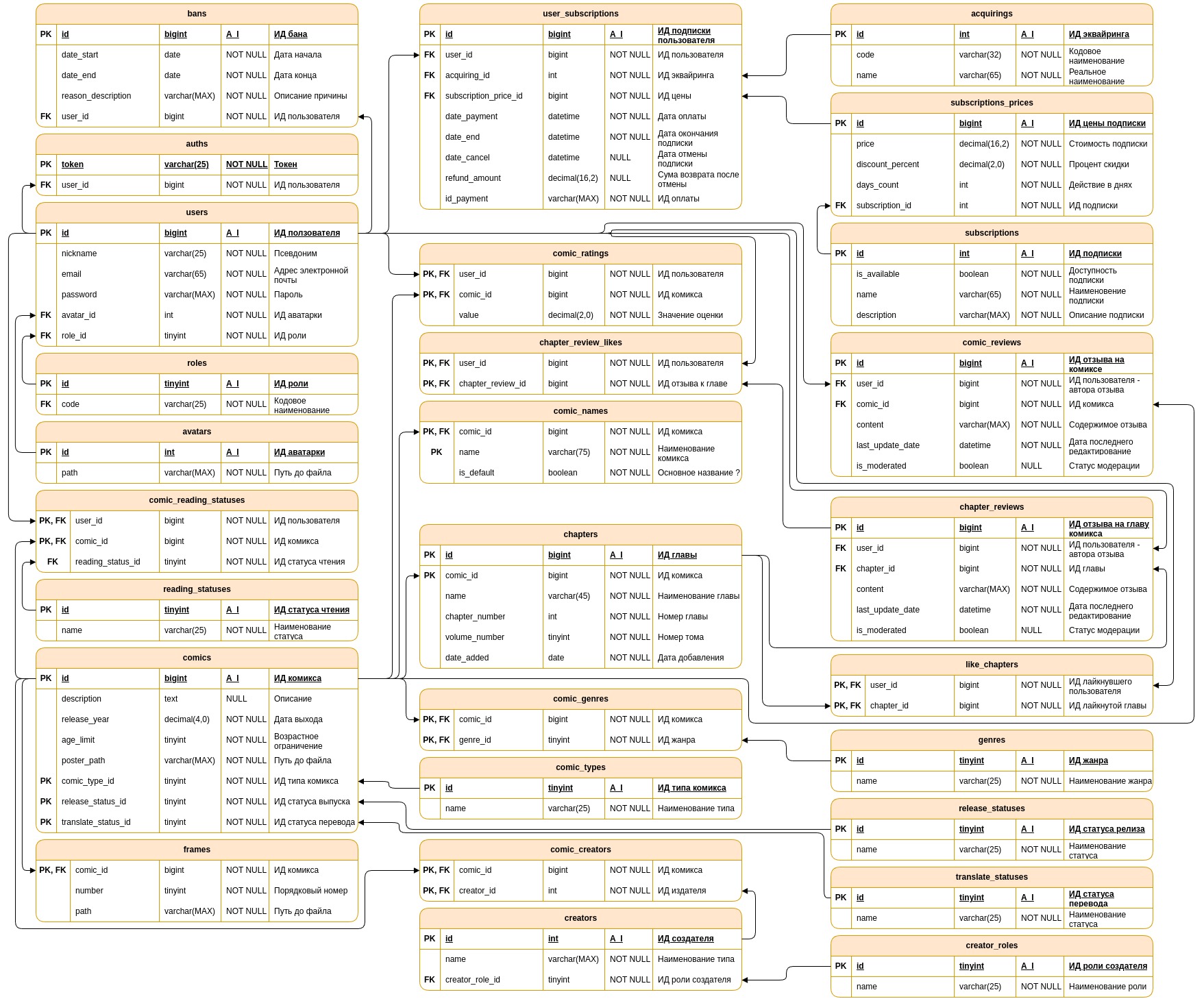


Рисунок 3 – Физическая модель базы данных

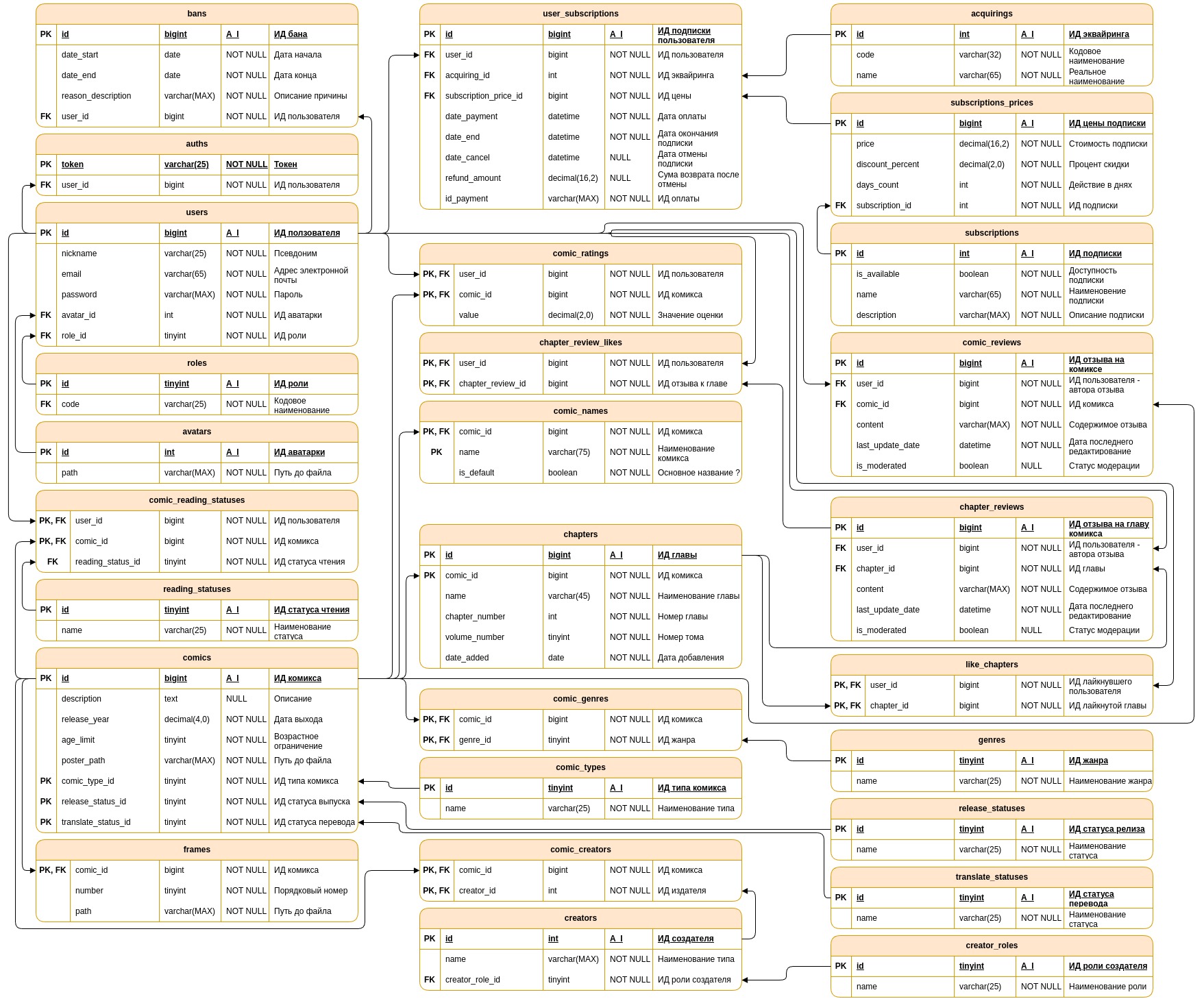


Рисунок 4 – Физическая модель базы данных

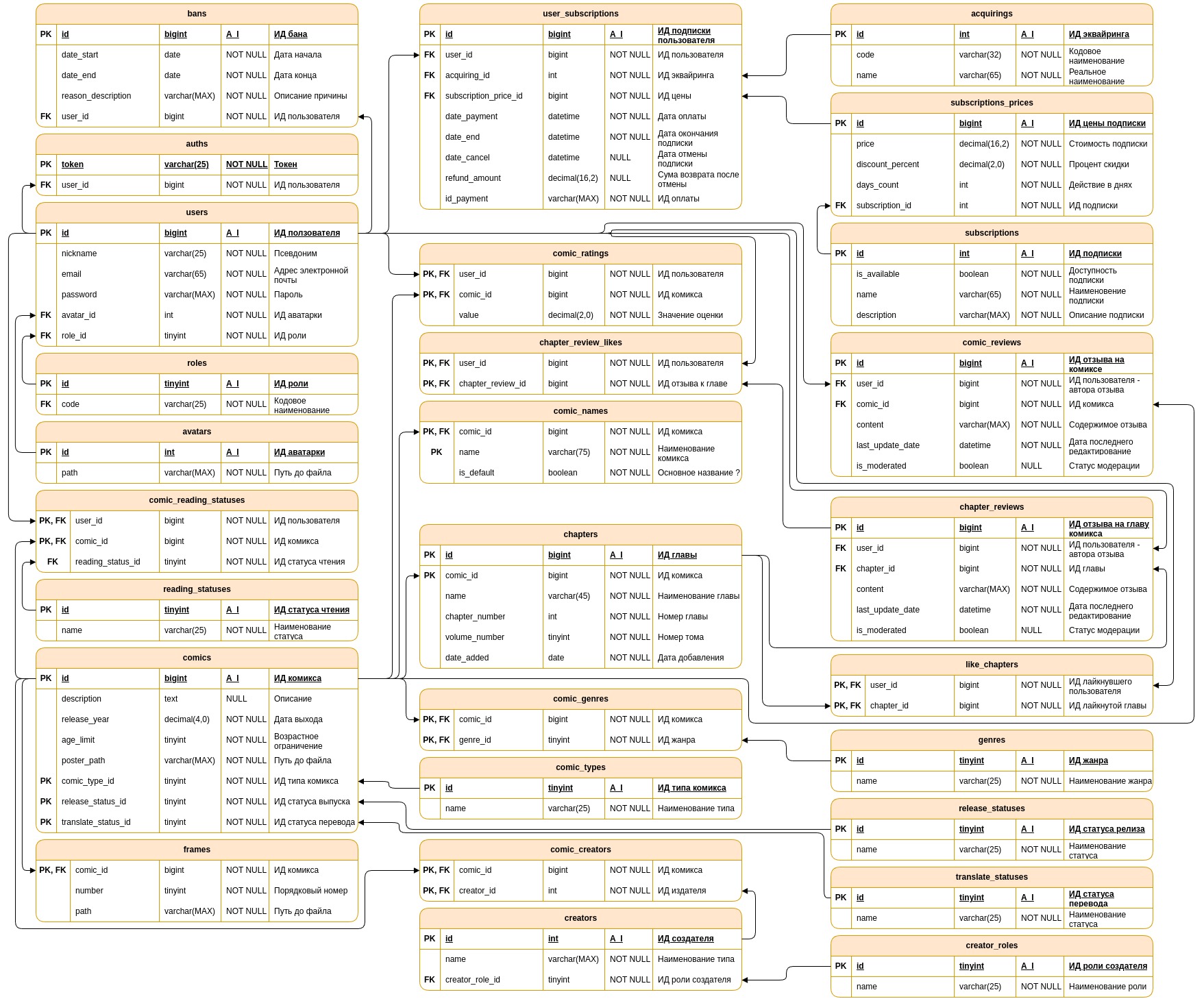


Рисунок 5 – Физическая модель базы данных

# СЛОВАРЬ ДАННЫХ

Словарь данных — это централизованная база данных, которая содержит метаинформацию о всех таблицах и их полях, а также описание данных, которые они хранят. Эта информация включает названия таблиц, имена полей, типы данных, ограничения уникальности, индексы и другие характеристики. Словарь данных предоставляет важную информацию, необходимую для администрирования, обслуживания и разработки баз данных, обеспечивая точность, согласованность и актуальность метаданных.

Словарь данных является неотъемлемой частью любой эффективной системы управления базами данных. Он позволяет организациям обеспечить стандартизацию, автоматизацию и контроль метаданных, что в свою очередь помогает снизить риски ошибок, улучшить производительность и обеспечить соответствие нормативным требованиям.

В данной таблице хранится информация о структуре и содержимом таблиц, что позволяет разработчикам и администраторам баз данных получить полное представление о данных, хранящихся в базе, и принимать обоснованные решения по ее оптимизации и управлению.

Таблица 1 – Словарь данных для БД «Онлайн библиотека комиксов»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| bans | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | bigint | Да | Первичный | ИД бана |
| date\_start | date | Да |  | Дата начала |
| date\_end | date | Да |  | Дата конца |
| reason\_description | varchar(MAX) | Да |  | Описание причины |
| user\_id | bigint | Да | Внешний | ИД пользователя |
| auths | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| token | varchar(25) | Да | Первичный | Токен |
| user\_id | bigint | Да | Внешний | ИД пользователя |
| users | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | bigint | Да | Первичный | ИД пользователя |
| nickname | varchar(25) | Да |  | Псевдоним |
| email | varchar(65) | Да |  | Адрес электронной почты |
| password | varchar(MAX) | Да |  | Пароль |
| avatar\_id | int | Да | Внешний | ИД аватара |
| role\_id | tinyint | Да | Внешний | ИД роли |
| roles | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | int | Да | Первичный | ИД роли |
| code | varchar(25) | Да |  | Кодовое наименование |
| avatars | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | int | Да | Первичный | ИД аватара |
| path | varchar(MAX) | Да |  | Путь до файла |
| comic\_reading\_statuses | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| user\_id | bigint | Да | Первичный | ИД пользователя |
| comic\_id | bigint | Да |  | ИД комикса |
| reading\_status\_id | tinyint | Да | Внешний | ИД статуса чтения |
| reading\_statuses | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | tinyint | Да | Первичный | ИД статуса чтения |
| name | varchar(25) | Да |  | Наименование статуса |
| comics | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | bigint | Да | Первичный | ИД комикса |
| description | text | Нет |  | Описание |
| release\_year | decimal(4,0) | Да |  | Дата выхода |
| age\_limit | tinyint | Да |  | Возрастное ограничение |
| poster\_path | varchar(MAX) | Да |  | Путь до файла |
| comic\_type\_id | tinyint | Да | Внешний | ИД типа комикса |
| release\_status\_id | tinyint | Да | Внешний | ИД статуса выпуска |
| translate\_status\_id | tinyint | Да | Внешний | ИД статуса перевода |
| frames | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| comic\_id | bigint | Да | Первичный, Внешний | ИД комикса |
| number | tinyint | Да |  | Порядковый номер |
| path | varchar(MAX) | Да |  | Путь до файла |
| user\_subscription | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | bigint | Да | Первичный | ИД подписки пользователя |
| user\_id | bigint | Да | Внешний | ИД пользователя |
| acquiring\_id | int | Да | Внешний | ИД эквайринга |
| subscription\_price\_id | bigint | Да | Внешний | ИД цены |
| date\_payment | datetime | Да |  | Дата оплаты |
| date\_end | datetime | Да |  | Дата окончания подписки |
| date\_cancel | datetime | Нет |  | Дата отмены подписки |
| refund\_amount | decimal(16,2) | Нет |  | Сумма возврата после отмены |
| id\_payment | varchar(MAX) | Да |  | ИД оплаты |
| comic\_ratings | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| user\_id | bigint | Да | Первичный, Внешний | ИД пользователя |
| comic\_id | bigint | Да | Первичный, Внешний | ИД комикса |
| value | decimal(2,0) | Да |  | Значение оценки |
| chapter\_review\_likes | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| user\_id | bigint | Да | Первичный, Внешний | ИД пользователя |
| chapter\_review\_id | bigint | Да | Первичный, Внешний | ИД отзыва к главе |
| comic\_names | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| comic\_id | bigint | Да | Первичный, Внешний | ИД комикса |
| name | varchar(75) | Да | Внешний | Наименование комикса |
| is\_default | boolean | Да |  | Основное название ? |
| chapters | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | bigint | Да | Первичный | ИД главы |
| comic\_id | bigint | Да | Внешний | ИД комикса |
| name | varchar(75) | Да |  | Наименование главы |
| chapter\_number | int | Да |  | Номер главы |
| volume\_number | tinyint | Да |  | Номер тома |
| date\_added | date | Да |  | Дата добавления |
| comic\_genres | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| comic\_id | bigint | Да | Первичный, Внешний | ИД комикса |
| genre\_id | tinyint | Да | Первичный, Внешний | ИД жанра |
| comic\_types | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | tinyint | Да | Первичный | ИД типа комикса |
| name | varchar(25) | Да |  | Наименование типа |
| comic\_creators | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| comic\_id | bigint | Да | Первичный, Внешний | ИД комикса |
| creator\_id | int | Да | Первичный, Внешний | ИД издателя |
| creators | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | int | Да | Первичный | ИД создателя |
| name | varchar(35) | Да |  | Наименование типа |
| creator\_role\_id | tinyint | Да | Внешний | ИД роли создателя |
| acquirings | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | int | Да | Первичный | ИД эквайринга |
| code | varchar(32) | Да |  | Кодовое наименование |
| name | varchar(65) | Да |  | Реальное наименование |
| subscription\_prices | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | bigint | Да | Первичный | ИД цены подписки |
| price | decimal(12,2) | Да |  | Стоимость подписки |
| discount\_percent | decimal(2,0) | Да |  | Процент скидки |
| days\_count | int | Да |  | Действие в днях |
| subscription\_id | int | Да | Внешний | ИД подписки |
| subscriptions | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | int | Да | Первичный | ИД подписки |
| is\_available | booleand | Да |  | Доступность подписки |
| name | varchar(65) | Да |  | Наименование подписки |
| description | varchar(MAX) | Да |  | Описание подписки |
| comic\_reviews | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | bigint | Да | Первичный | ИД отзыва на комиксе |
| user\_id | bigint | Да | Внешний | ИД пользователя — автора отзыва |
| comic\_id | bigint | Да | Внешний | ИД комикса |
| content | varchar(MAX) | Да |  | Содержимое отзыва |
| last\_update\_date | datetime | Да |  | Дата последнего редактирования |
| is\_moderated | boolean | Нет |  | Статус модерации |
| chapter\_reviews | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | bigint | Да | Первичный | ИД отзыва на главу комикса |
| user\_id | bigint | Да | Внешний | ИД пользователя — автора отзыва |
| chapter\_id | bigint | Да | Внешний | ИД главы |
| content | varchar(MAX) | Да |  | Содержимое отзыва |
| last\_update\_date | datetime | Да |  | Дата последнего редактирования |
| is\_moderated | boolean | Нет |  | Статус модерации |
| like\_chapters | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| user\_id | bigint | Да | Первичный, Внешний | ИД лайкнувшего пользователя |
| chapter\_id | bigint | Да | Первичный, Внешний | ИД лайкнутой главы |
| genres | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | tinyint | Да | Первичный | ИД жанра |
| name | varchar(25) | Да |  | Наименование жанра |
| release\_statuses | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | tinyint | Да | Первичный | ИД статуса релиза |
| name | varchar(25) | Да |  | Наименование статуса |
| translate\_statuses | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | tinyint | Да | Первичный | ИД статуса перевода |
| name | varchar(25) | Да |  | Наименование статуса |
| creator\_roles | | | | |
| Поле | Формат | Обязательное | Ключ | Примечание |
| id | tinyint | Да | Первичный | ИД роли создателя |
| name | varchar(25) | Да |  | Наименование роли |

# РАЗРАБОТКА БАЗЫ ДАННЫХ В PGADMIN4

После проектирования концептуальной, логической и физической моделей, была разработана база данных в pgAdmin4 для СУБД postgresql. В процессе разработки были описаны связи между таблицами, приведены примеры заполненных данными таблиц, а также проведена проверка ограничений на поля для обеспечения целостности и корректности данных (см. рис. 6.1).

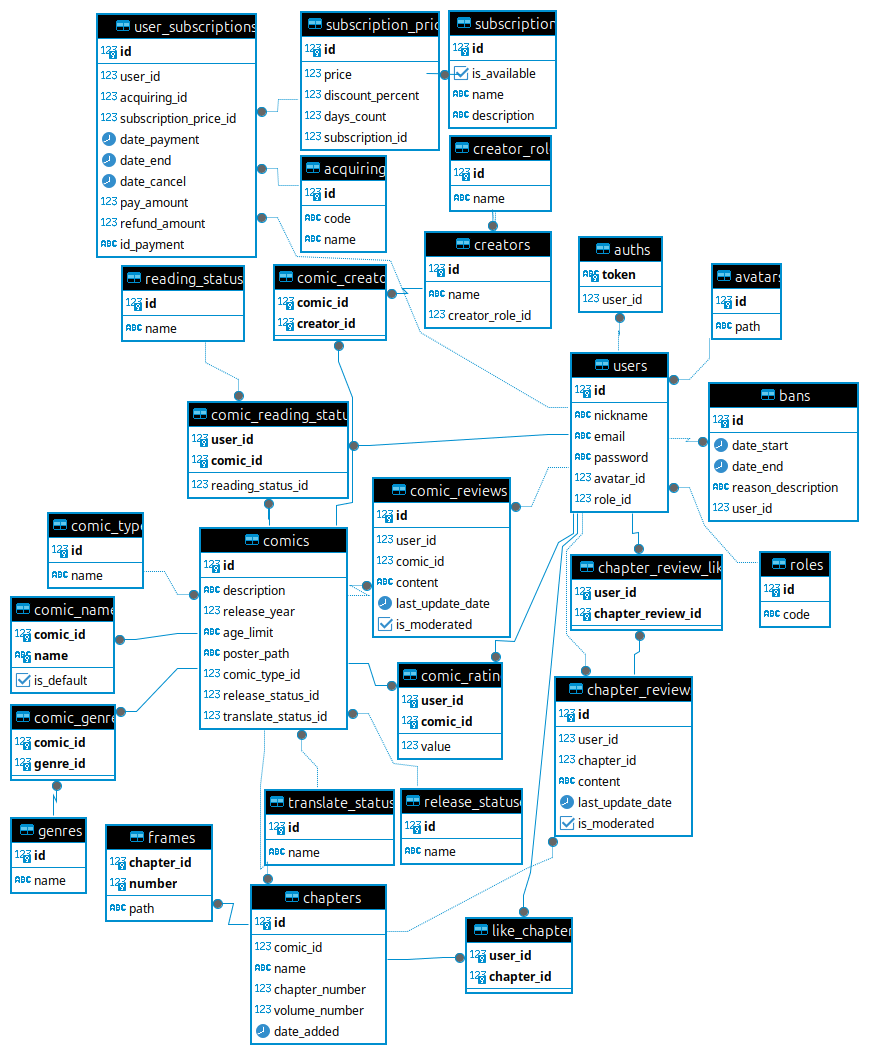


Рисунок 6.1 – Разработка базы данных в MS SQL

## Создание связей между таблицами

1. auths – > users (Аутентификации –> Пользователи)
   * При удалении: Нет действия (NO ACTION), так как аутентификационные данные могут оставаться в системе независимо от пользователя.
   * При обновлении: Cascade, так как если идентификатор пользователя изменяется, необходимо обновить его в связанных аутентификационных записях.
2. bans –> users (Блокировки –> Пользователи)
   * При удалении: Cascade, чтобы удалить информацию о бане при удалении пользователя.
   * При обновлении: Cascade, чтобы обновить информацию о бане при изменении пользователя.
3. chapter\_review\_likes –> users (Лайки комментариев к главам –> Пользователи)
   * При удалении: Cascade, чтобы удалить лайки при удалении пользователя.
   * При обновлении: Cascade, чтобы обновить информацию о пользователях, которые поставили лайки на комментарии глав.
4. chapter\_review\_likes –> chapter\_reviews (Лайки комментариев к главам –> Комментарии к главам)
   * При удалении: Cascade, чтобы удалить лайки при удалении обзора главы.
   * При обновлении: Cascade, чтобы обновить информацию о связанных обзорах главы при изменении.
5. chapter\_reviews –> users (Комментарии глав –> Пользователи)
   * При удалении: Cascade, чтобы удалить обзоры глав при удалении пользователя.
   * При обновлении: Cascade, чтобы обновить информацию о пользователях, оставивших обзоры глав.
6. chapters –> comics (Главы –> Комиксы)
   * При удалении: Cascade, чтобы удалить главы при удалении связанного комикса.
   * При обновлении: Cascade, чтобы обновить информацию о комиксе при изменении.
7. comic\_genres –> comics (Жанры комиксов –> Комиксы)
   * При удалении: Cascade, чтобы удалить связь с жанром при удалении связанного комикса.
   * При обновлении: Cascade, чтобы обновить информацию о комиксе при изменении.

8. comic\_genres –> genres (Жанры комиксов –> Жанры)

* + При удалении: Cascade, чтобы удалить жанр из связанных жанров комикса при удалении самого жанра.
  + При обновлении: Cascade, чтобы обновить информацию о жанре в связанных жанрах комикса при изменении.

9. comic\_names –> comics (Названия комиксов –> Комиксы)

* + При удалении: Cascade, чтобы удалить название при удалении связанного комикса.
  + При обновлении: Cascade, чтобы обновить информацию о комиксе при изменении.

10. comic\_names –> users (Названия комиксов –> Пользователи)

* + При удалении: Cascade, чтобы удалить название при удалении связанного пользователя.
  + При обновлении: Cascade, чтобы обновить информацию о пользователе при изменении.

11. comic\_ratings –> comics (Рейтинги комиксов –> Комиксы)

* + При удалении: Cascade, чтобы удалить рейтинг при удалении связанного комикса.
  + При обновлении: Cascade, чтобы обновить информацию о комиксе при изменении.

12. comic\_ratings –> users (Рейтинги комиксов –> Пользователи)

* + При удалении: Cascade, чтобы удалить рейтинг при удалении связанного пользователя.
  + При обновлении: Cascade, чтобы обновить информацию о пользователе при изменении.

13. comic\_reading\_statuses –> comics (Статусы чтения комиксов –> Комиксы)

* + При удалении: Cascade, чтобы удалить статус чтения при удалении связанного комикса.
  + При обновлении: Cascade, чтобы обновить информацию о комиксе при изменении.

14. comic\_reading\_statuses –> reading\_statuses (Статусы чтения комиксов –> Статусы чтения)

* + При удалении: Cascade, чтобы удалить статус чтения при удалении связанного статуса чтения.
  + При обновлении: Cascade, чтобы обновить информацию о статусе чтения при изменении.

15. comic\_reading\_statuses –> users (Статусы чтения комиксов –> Пользователи)

* + При удалении: Cascade, чтобы удалить статус чтения при удалении связанного пользователя.
  + При обновлении: Cascade, чтобы обновить информацию о пользователе при изменении.

16. comic\_reviews –> comics (Отзывы комиксы –> Комиксы)

* + При удалении: Cascade, чтобы удалить отзыв при удалении связанного комикса.
  + При обновлении: Cascade, чтобы обновить информацию о комиксе при изменении.

17. comic\_reviews –> users (Отзывы на комиксы –> Пользователи)

* + При удалении: Cascade, чтобы удалить отзыв при удалении связанного пользователя.
  + При обновлении: Cascade, чтобы обновить информацию о пользователе при изменении.

18. comics –> comic\_types (Комиксы –> Типы комиксов)

* + При удалении: No Action, так как тип комикса может оставаться в системе независимо от комиксов.
  + При обновлении: Cascade, чтобы обновить информацию о типе комикса при изменении.

19. comics –> release\_statuses (Комиксы –> Статусы выпуска)

* + При удалении: No Action, так как тип комикса может оставаться в системе независимо от комиксов.
  + При обновлении: Cascade, чтобы обновить информацию о типе статуса выпуска комикса при изменении.

20. comics –> translate\_statuses (Комиксы –> Статусы перевода)

* + При удалении: No Action, так как статус перевода может оставаться в системе независимо от комиксов.
  + При обновлении: Cascade, чтобы обновить информацию о статусе перевода при изменении.

21. frames –> chapters (Кадры –> Главы)

* + При удалении: Cascade, чтобы удалить кадры при удалении связанной главы.
  + При обновлении: Cascade, чтобы обновить информацию о главе при изменении.

22. like\_chapters –> users (Лайки на главы –> Пользователи)

* + При удалении: Cascade, чтобы удалить лайк при удалении связанного пользователя.
  + При обновлении: Cascade, чтобы обновить информацию о пользователе при изменении.

23. like\_chapters –> chapters (Лайки на главы –> Главы)

* + При удалении: Cascade, чтобы удалить лайк при удалении связанной главы.
  + При обновлении: Cascade, чтобы обновить информацию о главе при изменении.

24. roles –> users (Роли –> Пользователи)

* + При удалении: No Action, чтобы не позволить удалить роль, есил есть связанные пользователи. Это позволяет пользователям оставаться в системе и всегда иметь свою роль.
  + При обновлении: Cascade, чтобы обновить информацию о роли в связанных пользователях при изменении.

25. subscription\_prices –> subscriptions (Цены подписки –> Подписки)

* + При удалении: No Action, чтобы нельзя было удалить цену при удалении связанной подписки.
  + При обновлении: Cascade, чтобы обновить информацию о подписке при изменении.

# ЗАПОЛНЕНИЕ ТАБЛИЦ ДАННЫМИ

После окончания проектирования и разработки базы данных было выполнено заполнение таблиц данными. Это позволило проверить работоспособность ограничений, связей и структуры базы данных, а также убедиться в правильности реализации задуманных функциональностей.

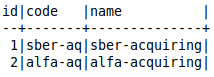
Заполнение производилось последовательно для каждой таблицы, в соответствии с её предназначением и зависимостями от других таблиц. Ниже описан процесс заполнения каждой из таблиц.

## Таблица «acquiring»

Для проверки созданных ограничений было произведено заполнение таблицы *«acquiring»* (см. рис. 6). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, что гарантирует уникальность каждой записи.
* Поле *«code»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«name»* также обязательно для заполнения и не может быть пустым.

Вывод: В таблице *«acquirings»*, предназначенной для хранения информации об эквайрингах, поле *«id»* обеспечивает уникальность каждой записи, а поля *«code»* и *«name»* обязательны для заполнения, что обеспечивает корректное идентифицирование и описание каждого эквайринга.

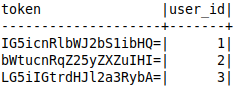
Рисунок 6 – Заполнение данными таблицы «acquiring»

## Таблица «auths»

Для проверки созданных ограничений было произведено заполнение таблицы *«auths»* (см. рис. 7). В результате были проверены следующие ограничения:

* Поле *«token»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице *«users»*.

Вывод: В таблице *«auths»*, предназначенной для хранения информации об аутентификации, поле *«token»* обеспечивает уникальность каждой записи, а связь с таблицей *«users»* через *«user\_id»* позволяет отслеживать связанных пользователей.

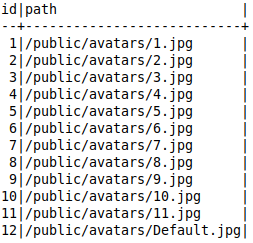
Рисунок 7 – Заполнение данными таблицы «auths»

## Таблица «avatars»

Для проверки созданных ограничений было произведено заполнение таблицы *«avatars»* (см. рис. 8). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, гарантирующим уникальность каждой записи.
* Поле *«path»* представляет собой обязательное поле, которое не может быть пустым.

Вывод: В таблице *«avatars»*, предназначенной для хранения путей к аватаркам пользователей, поле id обеспечивает уникальность каждой записи, а поле *«path»* гарантирует, что каждый аватар имеет корректный путь к файлу.

Рисунок 8 – Заполнение данными таблицы «avatars»

## Таблица «bans»

Для проверки созданных ограничений было произведено заполнение таблицы *«bans»* (см. рис. 9). В результате были проверены следующие ограничения:

* Поле id является первичным ключом, обеспечивающим уникальность каждой записи.
* Поля *«date\_start»* и *«date\_end»* являются обязательными для заполнения.
* Поле *«reason\_description»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице *«users»*.

Вывод: В таблице *«bans»*, предназначенной для хранения информации о блокировках пользователей, поля *«date\_start»* и *«date\_end»* определяют временной период блокировки, а поле *«reason\_description»* содержит причину блокировки. Связь с таблицей *«users»* через *«user\_id»* позволяет идентифицировать заблокированного пользователя.

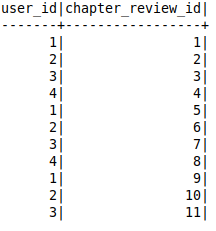
Рисунок 9 – Заполнение данными таблицы «bans»

## Таблица «chapter\_review\_likes»

Для проверки созданных ограничений было произведено заполнение таблицы *«chapter\_review\_likes»* (см. рис. 10). В результате были проверены следующие ограничения:

* Поля *«user\_id»* и *«chapter\_review\_id»* вместе составляют составной первичный ключ, гарантирующий уникальность комбинации.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице *«users»*.
* Поле *«chapter\_review\_id»* является обязательным для заполнения, ссылающимся на идентификатор отзыва в таблице *«chapter\_reviews»*.

Вывод: В таблице *«chapter\_review\_likes»*, предназначенной для хранения информации о лайках на отзывы о главах, составной первичный ключ из *«user\_id»* и *«chapter\_review\_id»* гарантирует уникальность каждой записи лайка. Связи с таблицами *«users»* и *«chapter\_reviews»* через соответствующие идентификаторы позволяют определить пользователя и отзыв, на который был поставлен лайк.

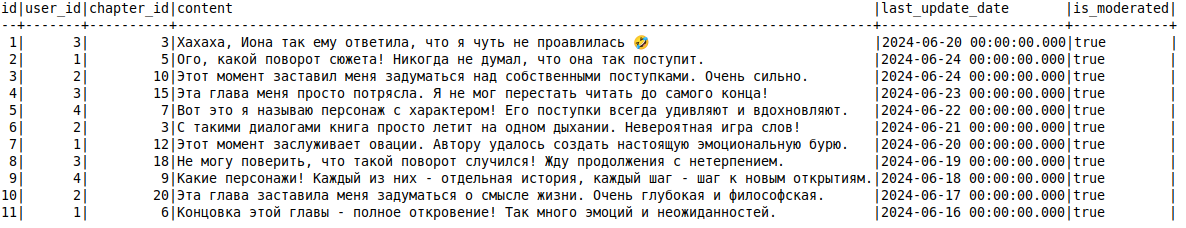
Рисунок 10 – Заполнение данными таблицы «chapter\_review\_likes»

## Таблица «chapter\_reviews»

Для проверки созданных ограничений было произведено заполнение таблицы *«chapter\_reviews»* (см. рис. 11). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице *«users»*.
* Поле *«chapter\_id»* является обязательным для заполнения, ссылающимся на идентификатор главы в таблице *«chapters»*.
* Поле *«content»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«last\_update\_date»* является обязательным для заполнения.

Вывод: В таблице *«chapter\_reviews»*, предназначенной для хранения отзывов пользователей о главах комиксов, поля *«user\_id»* и *«chapter\_id»* связывают отзывы с соответствующими пользователями и главами. Поле *«last\_update\_date»* обеспечивает отслеживание времени последнего обновления отзыва.

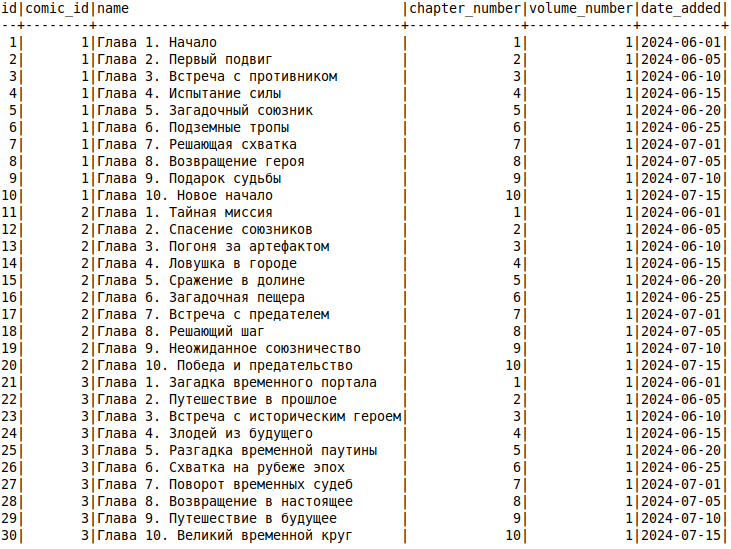
Рисунок 11 – Заполнение данными таблицы «chapter\_reviews»

## Таблица «chapters»

Для проверки созданных ограничений было произведено заполнение таблицы *«chapters»* (см. рис. 12). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«comic\_id»* является обязательным для заполнения, ссылающимся на идентификатор комикса в таблице *«comics»*.
* Поле name представляет собой обязательное поле, которое не может быть пустым.
* Поле *«chapter\_number»* является обязательным для заполнения.
* Поле *«volume\_number»* является обязательным для заполнения.
* Поле *«date\_added»* является обязательным для заполнения.

Вывод: В таблице *«chapters»*, предназначенной для хранения информации о главах комиксов, поля *«comic\_id»* связывает главы с соответствующими комиксами. Поля *«chapter\_number»* и *«volume\_number»* определяют номер главы и тома, а поле *«date\_added»* отражает дату добавления главы.

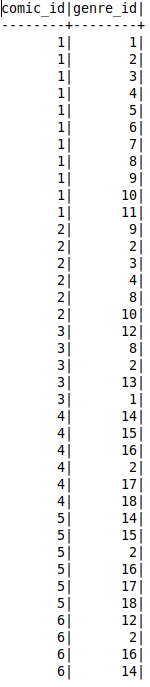
Рисунок 12 – Заполнение данными таблицы «chapters»

## Таблица «comic\_genres»

Для проверки созданных ограничений было произведено заполнение таблицы *«comic\_genres»* (см. рис. 13). В результате были проверены следующие ограничения:

* Поле *«comic\_id»* и *«genre\_id»* вместе составляют составной первичный ключ, гарантирующий уникальность комбинации жанра и комикса.
* Поле *«comic\_id»* является обязательным для заполнения, ссылающимся на идентификатор комикса в таблице *«comics»*.
* Поле *«genre\_id»* является обязательным для заполнения, ссылающимся на идентификатор жанра в таблице *«genres»*.

Вывод: В таблице *«comic\_genres»*, предназначенной для связи комиксов с их жанрами, составной первичный ключ из *«comic\_id»* и *«genre\_id»* гарантирует уникальность каждой записи. Связь с таблицами *«comics»* и *«genres»* позволяет определить, какие жанры присвоены каждому комиксу.

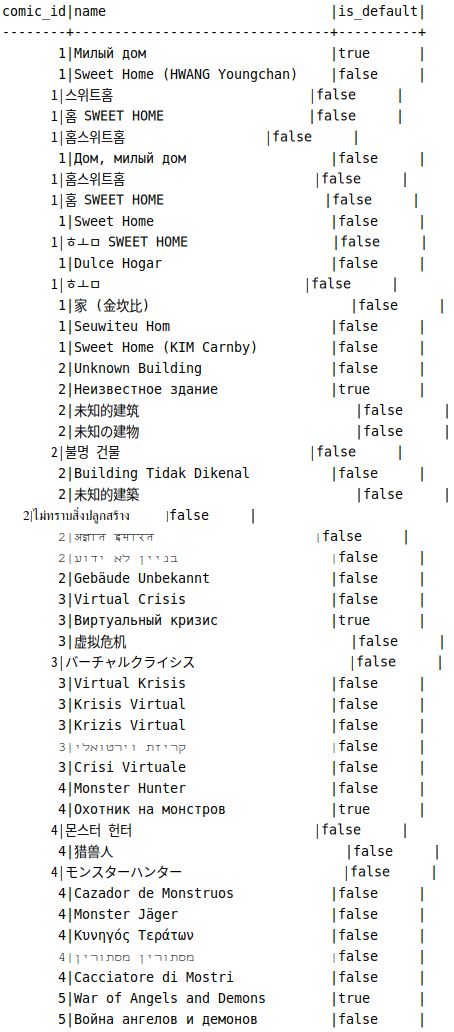
Рисунок 13 – Заполнение данными таблицы «comic\_genres»

## Таблица «comic\_names»

Для проверки созданных ограничений было произведено заполнение таблицы *«comic\_names»* (см. рис. 14). В результате были проверены следующие ограничения:

* Поле *«user\_id»* и *«comic\_id»* вместе составляют составной первичный ключ, гарантирующий уникальность комбинации пользователя и комикса.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице *«users»*.
* Поле *«comic\_id»* является обязательным для заполнения, ссылающимся на идентификатор комикса в таблице *«comics»*.
* Поле *«value»* является обязательным для заполнения.

Вывод: В таблице *«comic\_names»*, предназначенной для хранения названий комиксов, составной первичный ключ из *«user\_id»* и *«comic\_id»* гарантирует уникальность каждой записи названия. Связь с таблицами *«users»* и *«comics»* позволяет определить, какие названия назначены каждому комиксу каждым пользователем.

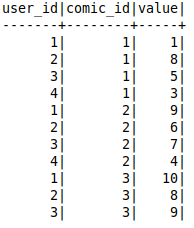
Рисунок 14 – Заполнение данными таблицы «comic\_names»

## Таблица «comic\_ratings»

Для проверки созданных ограничений было произведено заполнение таблицы *«comic\_ratings»* (см. рис. 15). В результате были проверены следующие ограничения:

* Поле *«user\_id»* и *«comic\_id»* вместе составляют составной первичный ключ, гарантирующий уникальность комбинации пользователя и комикса.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице *«users»*.
* Поле *«comic\_id»* является обязательным для заполнения, ссылающимся на идентификатор комикса в таблице *«comics»*.
* Поле *«value»* является обязательным для заполнения.

Вывод: В таблице *«comic\_ratings»*, предназначенной для хранения рейтингов пользователей для комиксов, составной первичный ключ из *«user\_id»* и *«comic\_id»* гарантирует уникальность каждой записи рейтинга. Связь с таблицами *«users»* и *«comics»* позволяет определить, какие рейтинги установлены каждому комиксу каждым пользователем.

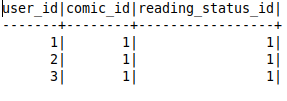
Рисунок 15 – Заполнение данными таблицы «comic\_ratings»

## Таблица «comic\_reading\_statuses»

Для проверки созданных ограничений было произведено заполнение таблицы *«comic\_reading\_statuses»* (см. рис. 16). В результате были проверены следующие ограничения:

* Поле *«user\_id»* и *«comic\_id»* вместе составляют составной первичный ключ, гарантирующий уникальность комбинации пользователя и комикса в контексте статуса чтения.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице *«users»*.
* Поле *«comic\_id»* является обязательным для заполнения, ссылающимся на идентификатор комикса в таблице *«comics»*.
* Поле *«eading\_status\_id»* является обязательным для заполнения, ссылающимся на идентификатор статуса чтения в таблице *«reading\_statuses»*.

Вывод: В таблице *«comic\_reading\_statuses»*, предназначенной для отслеживания статусов чтения комиксов пользователями, составной первичный ключ из *«user\_id»* и *«comic\_id»* обеспечивает уникальность каждого статуса чтения. Связь с таблицами *«users»*, *«comics»* и *«reading\_statuses»* позволяет определить, какие статусы чтения установлены для каждого комикса каждым пользователем.

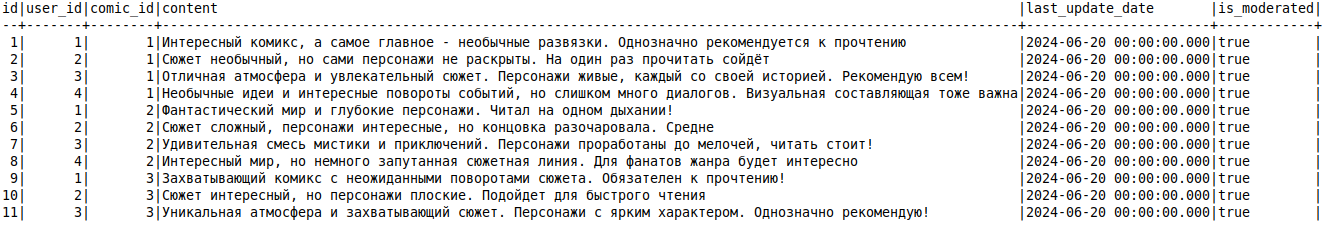
Рисунок 16 – Заполнение данными таблицы «comic\_reading\_statuses»

## Таблица «comic\_reviews»

Для проверки созданных ограничений было произведено заполнение таблицы *«Отзывы о комиксах»* (см. рис. 17). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице *«users»*.
* Поле *«comic\_id»* является обязательным для заполнения, ссылающимся на идентификатор комикса в таблице *«comics»*.
* Поле *«content»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«last\_update\_date»* является обязательным для заполнения.
* Поле *«is\_moderated»* представляет собой поле, которое показывает, прошел ли отзыв модерацию.

Вывод: В таблице *«comic\_reviews»*, предназначенной для хранения отзывов пользователей о комиксах, поля *«user\_id»* и *«comic\_id»* связывают отзывы с соответствующими пользователями и комиксами. Поле *«last\_update\_date»* отслеживает время последнего обновления отзыва, а поле *«is\_moderated»* указывает на статус модерации отзыва.

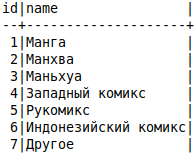
Рисунок 17 – Заполнение данными таблицы «comic\_reviews»

## Таблица «comic\_types»

Для проверки созданных ограничений было произведено заполнение таблицы *«comic\_types»* (см. рис. 18). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«name»* представляет собой обязательное поле, которое не может быть пустым.

Вывод: В таблице *«comic\_types»*, предназначенной для хранения типов комиксов, поле name определяет название каждого типа. Поле *«id»* является автоинкрементным идентификатором, гарантирующим уникальность записей.

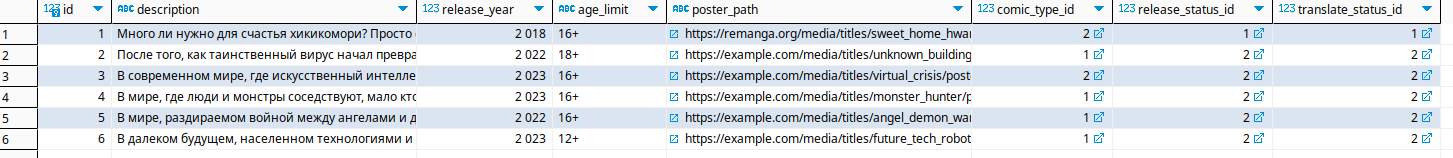
Рисунок 18 – Заполнение данными таблицы «comic\_types»

## Таблица «comics»

Для проверки созданных ограничений было произведено заполнение таблицы *«comics»* (см. рис. 19). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«description»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«release\_year»* является обязательным для заполнения.
* Поле *«age\_limit»* представляет собой обязательное поле.
* Поле *«poster\_path»* представляет собой обязательное поле.
* Поле *«comic\_type\_id»* является обязательным для заполнения, ссылающимся на идентификатор типа комикса в таблице *«comic\_types»*.
* Поля *«release\_status\_id»* и *«translate\_status\_id»* являются обязательными для заполнения, ссылающимися на идентификаторы статусов выпуска и перевода в таблицах *«release\_statuses»* и *«translate\_statuses»* соответственно.

Вывод: В таблице *«comics»*, предназначенной для хранения информации о комиксах, поля *«comic\_type\_id»*, *«release\_status\_id»* и *«translate\_status\_id»* связывают комиксы с их типами, статусами выпуска и перевода. Поля *«description»*, *«release\_year»*, *«age\_limit»* и *«poster\_path»* определяют описание, год выпуска, возрастное ограничение и путь к постеру комикса соответственно.

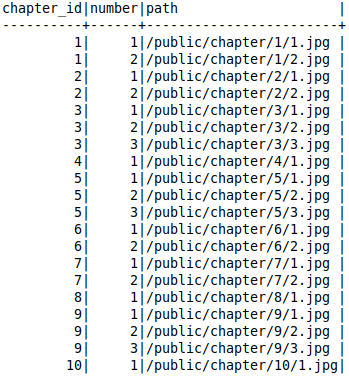
Рисунок 19 – Заполнение данными таблицы «comics»

## Таблица «frames»

Для проверки созданных ограничений было произведено заполнение таблицы *«frames»* (см. рис. 20). В результате были проверены следующие ограничения:

* Поле *«chapter\_id»* и *«number»* вместе составляют составной первичный ключ, гарантирующий уникальность комбинации главы и номера кадра.
* Поле *«chapter\_id»* является обязательным для заполнения, ссылающимся на идентификатор главы в таблице *«chapters»*.
* Поле *«number»* является обязательным для заполнения.
* Поле *«path»* представляет собой обязательное поле, которое не может быть пустым.

Вывод: В таблице *«frames»*, предназначенной для хранения кадров комиксов, составной первичный ключ из *«chapter\_id»* и *«number»* гарантирует уникальность каждого кадра в рамках главы. Поле *«path»* хранит путь к изображению кадра, а поля *«chapter\_id»* и *«number»* связывают кадры с соответствующими главами.

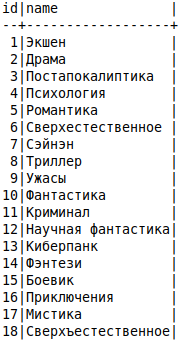
Рисунок 20 – Заполнение данными таблицы «frames»

## Таблица «genres»

Для проверки созданных ограничений было произведено заполнение таблицы *«genres»* (см. рис. 21). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«name»* представляет собой обязательное поле, которое не может быть пустым.

Вывод: В таблице *«genres»*, предназначенной для хранения информации о жанрах комиксов, поле name определяет название каждого жанра. Поле *«id»* является автоинкрементным идентификатором, обеспечивающим уникальность записей.

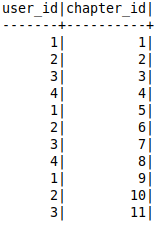
Рисунок 21 – Заполнение данными таблицы «genres»

## Таблица «like\_chapters»

Для проверки созданных ограничений было произведено заполнение таблицы *«like\_chapters»* (см. рис. 22). В результате были проверены следующие ограничения:

* Поле *«user\_id»* и *«chapter\_id»* вместе составляют составной первичный ключ, гарантирующий уникальность комбинации пользователя и главы комикса, которую он лайкнул.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице «users».
* Поле *«chapter\_id»* является обязательным для заполнения, ссылающимся на идентификатор главы в таблице *«chapters»*.

Вывод: В таблице *«like\_chapters»*, предназначенной для отслеживания лайков пользователей к главам комиксов, составной первичный ключ из *«user\_id»* и *«chapter\_id»* обеспечивает уникальность каждого лайка. Связь с таблицами *«users»* и *«chapters»* позволяет определить, какие главы комиксов были лайкнуты каждым пользователем.

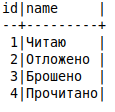
Рисунок 22 – Заполнение данными таблицы «like\_chapters»

## Таблица «reading\_statuses»

Для проверки созданных ограничений было произведено заполнение таблицы *«reading\_statuses»* (см. рис. 23). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«name»* представляет собой обязательное поле, которое не может быть пустым.

Вывод: В таблице *«reading\_statuses»*, предназначенной для хранения статусов чтения комиксов, поле name определяет название каждого статуса. Поле *«id»* является автоинкрементным идентификатором, обеспечивающим уникальность записей.

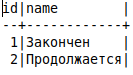
Рисунок 23 – Заполнение данными таблицы «reading\_statuses»

## Таблица «release\_statuses»

Для проверки созданных ограничений было произведено заполнение таблицы *«release\_statuses»* (см. рис. 24). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«name»* представляет собой обязательное поле, которое не может быть пустым.

Вывод: В таблице *«release\_statuses»*, предназначенной для хранения статусов выпуска комиксов, поле name определяет название каждого статуса. Поле id является автоинкрементным идентификатором, обеспечивающим уникальность записей.

Рисунок 24 – Заполнение данными таблицы «release\_statuses»

## Таблица «roles»

Для проверки созданных ограничений было произведено заполнение таблицы *«roles»* (см. рис. 25). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«code»* представляет собой обязательное поле, которое не может быть пустым.

Вывод: В таблице *«roles»*, предназначенной для хранения ролей пользователей, поле code определяет код каждой роли. Поле *«id»* является автоинкрементным идентификатором, обеспечивающим уникальность записей.

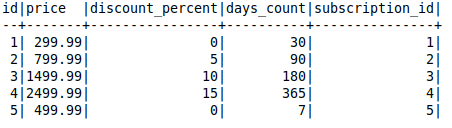
Рисунок 25 – Заполнение данными таблицы «roles»

## Таблица «subscription\_prices»

Для проверки созданных ограничений было произведено заполнение таблицы *«subscription\_prices»* (см. рис. 26). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«price»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«discount\_percent»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«days\_count»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«subscription\_id»* является обязательным для заполнения, ссылающимся на идентификатор подписки в таблице *«subscriptions»*.

Вывод: В таблице *«subscription\_prices»*, предназначенной для хранения цен подписок, поля *«price»*, *«discount\_percent»* и *«days\_count»* определяют соответственно цену, процент скидки и длительность подписки. Поле *«id»* является уникальным идентификатором записи, а *«subscription\_id»* связывает с таблицей *«subscriptions»*.

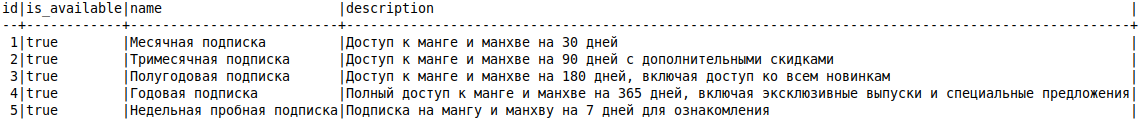
Рисунок 26 – Заполнение данными таблицы «subscription\_prices»

## Таблица «subscriptions»

Для проверки созданных ограничений было произведено заполнение таблицы *«subscriptions»* (см. рис. 27). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«price»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«discount\_percent*» представляет собой обязательное поле, которое не может быть пустым.
* Поле *«days\_count»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«subscription»* является обязательным для заполнения.

Вывод: В таблице *«subscriptions»*, предназначенной для хранения информации о доступных подписках, поля *«price»*, *«discount\_percent»* и *«days\_count»* определяют условия подписки. Поле *«id»* является автоинкрементным идентификатором записи.

Рисунок 27 – Заполнение данными таблицы «subscriptions»

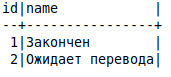
## Таблица «translate\_statuses»

Для проверки созданных ограничений было произведено заполнение таблицы *«translate\_statuses»* (см. рис. 28). В результате были проверены следующие ограничения:

Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.

Поле *«name»* представляет собой обязательное поле, которое не может быть пустым.

Вывод: В таблице *«translate\_statuses»*, предназначенной для хранения статусов перевода комиксов, поле name определяет название каждого статуса. Поле *«id»* является автоинкрементным идентификатором, обеспечивающим уникальность записей.

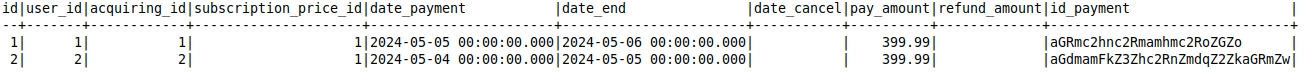
Рисунок 28 – Заполнение данными таблицы «translate\_statuses»

## Таблица «user\_subscriptions»

Для проверки созданных ограничений было произведено заполнение таблицы *«user\_subscriptions»* (см. рис. 29). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«user\_id»* является обязательным для заполнения, ссылающимся на идентификатор пользователя в таблице *«users»*.
* Поле *«acquiring\_id»* является обязательным для заполнения, ссылающимся на идентификатор приобретения в таблице acquirings.
* Поле *«subscription\_price\_id»* является обязательным для заполнения, ссылающимся на идентификатор цены подписки в таблице *«subscription\_prices»*.
* Поля *«date\_payment»*, *«date\_end»* и *«date\_cancel»* являются обязательными для заполнения.
* Поля *«pay\_amount»* и *«refund\_amount»* являются обязательными для заполнения.
* Поле *«id\_payment»* представляет собой обязательное поле типа.

Вывод: В таблице *«user\_subscriptions»*, предназначенной для хранения информации о подписках пользователей на комиксы, поля *«user\_id»*, *«acquiring\_id»* и *«subscription\_price\_id»* определяют связь с таблицами *«users»*, *«acquirings»* и *«subscription\_prices»* соответственно. Поля *«date\_payment»*, *«date\_end»*, *«date\_cancel»*, *«pay\_amount»*, *«refund\_amount»* и *«id\_payment»* отражают информацию о платежах и сроках действия подписки.

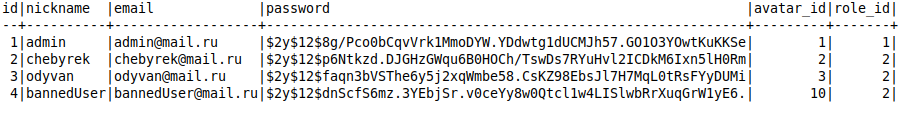
Рисунок 29 – Заполнение данными таблицы «user\_subscriptions»

## Таблица «users»

Для проверки созданных ограничений было произведено заполнение таблицы *«users»* (см. рис. 30). В результате были проверены следующие ограничения:

* Поле *«id»* является первичным ключом, обеспечивающим уникальность каждой записи.
* Поле *«nickname»* представляет собой обязательное поле, которое не может быть пустым.
* Поле *«email»* представляет собой обязательное поле, которое не может быть пустым и должно содержать корректный адрес электронной почты.
* Поле *«password»* представляет собой обязательное поле, которое не может быть пустым и хранит зашифрованный пароль пользователя.
* Поле *«avatar\_id»* является обязательным для заполнения, ссылающимся на идентификатор аватара в таблице *«avatars»*.
* Поле *«role\_id»* является обязательным для заполнения, ссылающимся на идентификатор роли в таблице *«roles»*.

Вывод: В таблице *«users»*, предназначенной для хранения информации о пользователях, поля *«nickname»*, *«email»* и *«password»* определяют основные учетные данные. Поля *«avatar\_id»* и *«role\_id»* связываются с таблицами *«avatars»* и *«roles»* соответственно для представления аватара пользователя и его роли.

Рисунок 30 – Заполнение данными таблицы «users»

# SQL-код создания базы данных

*Листинг 1 — SQL-код создания базы данных*

|  |
| --- |
| --  -- PostgreSQL database dump  --  -- Dumped from database version 16.3 (Debian 16.3-1.pgdg120+1)  -- Dumped by pg\_dump version 16.3 (Ubuntu 16.3-0ubuntu0.24.04.1)  -- Started on 2024-06-25 22:55:04 +07  SET statement\_timeout = 0;  SET lock\_timeout = 0;  SET idle\_in\_transaction\_session\_timeout = 0;  SET client\_encoding = 'UTF8';  SET standard\_conforming\_strings = on;  SELECT pg\_catalog.set\_config('search\_path', '', false);  SET check\_function\_bodies = false;  SET xmloption = content;  SET client\_min\_messages = warning;  SET row\_security = off;  --  -- TOC entry 4 (class 2615 OID 2200)  -- Name: public; Type: SCHEMA; Schema: -; Owner: -  --  CREATE SCHEMA public;  --  -- TOC entry 3643 (class 0 OID 0)  -- Dependencies: 4  -- Name: SCHEMA public; Type: COMMENT; Schema: -; Owner: -  --  COMMENT ON SCHEMA public IS 'standard public schema';  SET default\_tablespace = '';  SET default\_table\_access\_method = heap;  --  -- TOC entry 218 (class 1259 OID 56025)  -- Name: acquirings; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.acquirings (  id bigint NOT NULL,  code character varying(32) NOT NULL,  name character varying(65) NOT NULL  );  --  -- TOC entry 217 (class 1259 OID 56024)  -- Name: acquirings\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.acquirings\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3644 (class 0 OID 0)  -- Dependencies: 217  -- Name: acquirings\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.acquirings\_id\_seq OWNED BY public.acquirings.id;  --  -- TOC entry 231 (class 1259 OID 56103)  -- Name: auths; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.auths (  token character varying(25) NOT NULL,  user\_id bigint NOT NULL  );  --  -- TOC entry 226 (class 1259 OID 56058)  -- Name: avatars; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.avatars (  id bigint NOT NULL,  path character varying(255) NOT NULL  );  --  -- TOC entry 225 (class 1259 OID 56057)  -- Name: avatars\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.avatars\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3645 (class 0 OID 0)  -- Dependencies: 225  -- Name: avatars\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.avatars\_id\_seq OWNED BY public.avatars.id;  --  -- TOC entry 233 (class 1259 OID 56114)  -- Name: bans; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.bans (  id bigint NOT NULL,  date\_start date NOT NULL,  date\_end date NOT NULL,  reason\_description character varying(255) NOT NULL,  user\_id bigint NOT NULL  );  --  -- TOC entry 232 (class 1259 OID 56113)  -- Name: bans\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.bans\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3646 (class 0 OID 0)  -- Dependencies: 232  -- Name: bans\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.bans\_id\_seq OWNED BY public.bans.id;  --  -- TOC entry 257 (class 1259 OID 56302)  -- Name: chapter\_review\_likes; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.chapter\_review\_likes (  user\_id bigint NOT NULL,  chapter\_review\_id bigint NOT NULL  );  --  -- TOC entry 256 (class 1259 OID 56286)  -- Name: chapter\_reviews; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.chapter\_reviews (  id bigint NOT NULL,  user\_id bigint NOT NULL,  chapter\_id bigint NOT NULL,  content character varying(255) NOT NULL,  last\_update\_date timestamp(0) without time zone NOT NULL,  is\_moderated boolean NOT NULL  );  --  -- TOC entry 255 (class 1259 OID 56285)  -- Name: chapter\_reviews\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.chapter\_reviews\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3647 (class 0 OID 0)  -- Dependencies: 255  -- Name: chapter\_reviews\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.chapter\_reviews\_id\_seq OWNED BY public.chapter\_reviews.id;  --  -- TOC entry 249 (class 1259 OID 56221)  -- Name: chapters; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.chapters (  id bigint NOT NULL,  comic\_id bigint NOT NULL,  name character varying(45) NOT NULL,  chapter\_number integer NOT NULL,  volume\_number smallint NOT NULL,  date\_added date NOT NULL  );  --  -- TOC entry 248 (class 1259 OID 56220)  -- Name: chapters\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.chapters\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3648 (class 0 OID 0)  -- Dependencies: 248  -- Name: chapters\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.chapters\_id\_seq OWNED BY public.chapters.id;  --  -- TOC entry 263 (class 1259 OID 56351)  -- Name: comic\_creators; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.comic\_creators (  comic\_id bigint NOT NULL,  creator\_id bigint NOT NULL  );  --  -- TOC entry 247 (class 1259 OID 56205)  -- Name: comic\_genres; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.comic\_genres (  comic\_id bigint NOT NULL,  genre\_id bigint NOT NULL  );  --  -- TOC entry 252 (class 1259 OID 56257)  -- Name: comic\_names; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.comic\_names (  comic\_id bigint NOT NULL,  name character varying(75) NOT NULL,  is\_default boolean DEFAULT false NOT NULL  );  --  -- TOC entry 258 (class 1259 OID 56317)  -- Name: comic\_ratings; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.comic\_ratings (  user\_id bigint NOT NULL,  comic\_id bigint NOT NULL,  value numeric(2,0) NOT NULL  );  --  -- TOC entry 244 (class 1259 OID 56178)  -- Name: comic\_reading\_statuses; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.comic\_reading\_statuses (  user\_id bigint NOT NULL,  comic\_id bigint NOT NULL,  reading\_status\_id bigint NOT NULL  );  --  -- TOC entry 254 (class 1259 OID 56269)  -- Name: comic\_reviews; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.comic\_reviews (  id bigint NOT NULL,  user\_id bigint NOT NULL,  comic\_id bigint NOT NULL,  content character varying(255) NOT NULL,  last\_update\_date timestamp(0) without time zone NOT NULL,  is\_moderated boolean  );  --  -- TOC entry 253 (class 1259 OID 56268)  -- Name: comic\_reviews\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.comic\_reviews\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3649 (class 0 OID 0)  -- Dependencies: 253  -- Name: comic\_reviews\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.comic\_reviews\_id\_seq OWNED BY public.comic\_reviews.id;  --  -- TOC entry 237 (class 1259 OID 56133)  -- Name: comic\_types; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.comic\_types (  id smallint NOT NULL,  name character varying(25) NOT NULL  );  --  -- TOC entry 236 (class 1259 OID 56132)  -- Name: comic\_types\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.comic\_types\_id\_seq  AS smallint  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3650 (class 0 OID 0)  -- Dependencies: 236  -- Name: comic\_types\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.comic\_types\_id\_seq OWNED BY public.comic\_types.id;  --  -- TOC entry 241 (class 1259 OID 56147)  -- Name: comics; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.comics (  id bigint NOT NULL,  description text NOT NULL,  release\_year numeric(4,0) NOT NULL,  age\_limit character varying(255) NOT NULL,  poster\_path character varying(255) NOT NULL,  comic\_type\_id bigint NOT NULL,  release\_status\_id bigint NOT NULL,  translate\_status\_id bigint NOT NULL,  CONSTRAINT comics\_age\_limit\_check CHECK (((age\_limit)::text = ANY ((ARRAY['18+'::character varying, '16+'::character varying, '12+'::character varying, '6+'::character varying, '0+'::character varying])::text[])))  );  --  -- TOC entry 240 (class 1259 OID 56146)  -- Name: comics\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.comics\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3651 (class 0 OID 0)  -- Dependencies: 240  -- Name: comics\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.comics\_id\_seq OWNED BY public.comics.id;  --  -- TOC entry 260 (class 1259 OID 56333)  -- Name: creator\_roles; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.creator\_roles (  id smallint NOT NULL,  name character varying(25) NOT NULL  );  --  -- TOC entry 259 (class 1259 OID 56332)  -- Name: creator\_roles\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.creator\_roles\_id\_seq  AS smallint  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3652 (class 0 OID 0)  -- Dependencies: 259  -- Name: creator\_roles\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.creator\_roles\_id\_seq OWNED BY public.creator\_roles.id;  --  -- TOC entry 262 (class 1259 OID 56340)  -- Name: creators; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.creators (  id bigint NOT NULL,  name character varying(255) NOT NULL,  creator\_role\_id bigint NOT NULL  );  --  -- TOC entry 261 (class 1259 OID 56339)  -- Name: creators\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.creators\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3653 (class 0 OID 0)  -- Dependencies: 261  -- Name: creators\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.creators\_id\_seq OWNED BY public.creators.id;  --  -- TOC entry 250 (class 1259 OID 56232)  -- Name: frames; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.frames (  chapter\_id bigint NOT NULL,  number smallint NOT NULL,  path character varying(255) NOT NULL  );  --  -- TOC entry 246 (class 1259 OID 56199)  -- Name: genres; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.genres (  id smallint NOT NULL,  name character varying(25) NOT NULL  );  --  -- TOC entry 245 (class 1259 OID 56198)  -- Name: genres\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.genres\_id\_seq  AS smallint  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3654 (class 0 OID 0)  -- Dependencies: 245  -- Name: genres\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.genres\_id\_seq OWNED BY public.genres.id;  --  -- TOC entry 251 (class 1259 OID 56242)  -- Name: like\_chapters; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.like\_chapters (  user\_id bigint NOT NULL,  chapter\_id bigint NOT NULL  );  --  -- TOC entry 216 (class 1259 OID 56018)  -- Name: migrations; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.migrations (  id integer NOT NULL,  migration character varying(255) NOT NULL,  batch integer NOT NULL  );  --  -- TOC entry 215 (class 1259 OID 56017)  -- Name: migrations\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.migrations\_id\_seq  AS integer  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3655 (class 0 OID 0)  -- Dependencies: 215  -- Name: migrations\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.migrations\_id\_seq OWNED BY public.migrations.id;  --  -- TOC entry 243 (class 1259 OID 56172)  -- Name: reading\_statuses; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.reading\_statuses (  id smallint NOT NULL,  name character varying(25) NOT NULL  );  --  -- TOC entry 242 (class 1259 OID 56171)  -- Name: reading\_statuses\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.reading\_statuses\_id\_seq  AS smallint  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3656 (class 0 OID 0)  -- Dependencies: 242  -- Name: reading\_statuses\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.reading\_statuses\_id\_seq OWNED BY public.reading\_statuses.id;  --  -- TOC entry 239 (class 1259 OID 56140)  -- Name: release\_statuses; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.release\_statuses (  id smallint NOT NULL,  name character varying(25) NOT NULL  );  --  -- TOC entry 238 (class 1259 OID 56139)  -- Name: release\_statuses\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.release\_statuses\_id\_seq  AS smallint  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3657 (class 0 OID 0)  -- Dependencies: 238  -- Name: release\_statuses\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.release\_statuses\_id\_seq OWNED BY public.release\_statuses.id;  --  -- TOC entry 224 (class 1259 OID 56051)  -- Name: roles; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.roles (  id smallint NOT NULL,  code character varying(25) NOT NULL  );  --  -- TOC entry 223 (class 1259 OID 56050)  -- Name: roles\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.roles\_id\_seq  AS smallint  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3658 (class 0 OID 0)  -- Dependencies: 223  -- Name: roles\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.roles\_id\_seq OWNED BY public.roles.id;  --  -- TOC entry 222 (class 1259 OID 56039)  -- Name: subscription\_prices; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.subscription\_prices (  id bigint NOT NULL,  price numeric(16,2) NOT NULL,  discount\_percent numeric(2,0) NOT NULL,  days\_count smallint NOT NULL,  subscription\_id bigint NOT NULL  );  --  -- TOC entry 221 (class 1259 OID 56038)  -- Name: subscription\_prices\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.subscription\_prices\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3659 (class 0 OID 0)  -- Dependencies: 221  -- Name: subscription\_prices\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.subscription\_prices\_id\_seq OWNED BY public.subscription\_prices.id;  --  -- TOC entry 220 (class 1259 OID 56032)  -- Name: subscriptions; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.subscriptions (  id bigint NOT NULL,  is\_available boolean NOT NULL,  name character varying(65) NOT NULL,  description character varying(255) NOT NULL  );  --  -- TOC entry 219 (class 1259 OID 56031)  -- Name: subscriptions\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.subscriptions\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3660 (class 0 OID 0)  -- Dependencies: 219  -- Name: subscriptions\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.subscriptions\_id\_seq OWNED BY public.subscriptions.id;  --  -- TOC entry 235 (class 1259 OID 56126)  -- Name: translate\_statuses; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.translate\_statuses (  id smallint NOT NULL,  name character varying(25) NOT NULL  );  --  -- TOC entry 234 (class 1259 OID 56125)  -- Name: translate\_statuses\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.translate\_statuses\_id\_seq  AS smallint  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3661 (class 0 OID 0)  -- Dependencies: 234  -- Name: translate\_statuses\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.translate\_statuses\_id\_seq OWNED BY public.translate\_statuses.id;  --  -- TOC entry 230 (class 1259 OID 56082)  -- Name: user\_subscriptions; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.user\_subscriptions (  id bigint NOT NULL,  user\_id bigint NOT NULL,  acquiring\_id bigint NOT NULL,  subscription\_price\_id bigint NOT NULL,  date\_payment timestamp(0) without time zone NOT NULL,  date\_end timestamp(0) without time zone NOT NULL,  date\_cancel timestamp(0) without time zone,  pay\_amount numeric(16,2) NOT NULL,  refund\_amount numeric(16,2),  id\_payment character varying(255)  );  --  -- TOC entry 229 (class 1259 OID 56081)  -- Name: user\_subscriptions\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.user\_subscriptions\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3662 (class 0 OID 0)  -- Dependencies: 229  -- Name: user\_subscriptions\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.user\_subscriptions\_id\_seq OWNED BY public.user\_subscriptions.id;  --  -- TOC entry 228 (class 1259 OID 56065)  -- Name: users; Type: TABLE; Schema: public; Owner: -  --  CREATE TABLE public.users (  id bigint NOT NULL,  nickname character varying(25) NOT NULL,  email character varying(65) NOT NULL,  password character varying(255) NOT NULL,  avatar\_id bigint NOT NULL,  role\_id bigint NOT NULL  );  --  -- TOC entry 227 (class 1259 OID 56064)  -- Name: users\_id\_seq; Type: SEQUENCE; Schema: public; Owner: -  --  CREATE SEQUENCE public.users\_id\_seq  START WITH 1  INCREMENT BY 1  NO MINVALUE  NO MAXVALUE  CACHE 1;  --  -- TOC entry 3663 (class 0 OID 0)  -- Dependencies: 227  -- Name: users\_id\_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: -  --  ALTER SEQUENCE public.users\_id\_seq OWNED BY public.users.id;  --  -- TOC entry 3335 (class 2604 OID 56028)  -- Name: acquirings id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.acquirings ALTER COLUMN id SET DEFAULT nextval('public.acquirings\_id\_seq'::regclass);  --  -- TOC entry 3339 (class 2604 OID 56061)  -- Name: avatars id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.avatars ALTER COLUMN id SET DEFAULT nextval('public.avatars\_id\_seq'::regclass);  --  -- TOC entry 3342 (class 2604 OID 56117)  -- Name: bans id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.bans ALTER COLUMN id SET DEFAULT nextval('public.bans\_id\_seq'::regclass);  --  -- TOC entry 3352 (class 2604 OID 56289)  -- Name: chapter\_reviews id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapter\_reviews ALTER COLUMN id SET DEFAULT nextval('public.chapter\_reviews\_id\_seq'::regclass);  --  -- TOC entry 3349 (class 2604 OID 56224)  -- Name: chapters id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapters ALTER COLUMN id SET DEFAULT nextval('public.chapters\_id\_seq'::regclass);  --  -- TOC entry 3351 (class 2604 OID 56272)  -- Name: comic\_reviews id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_reviews ALTER COLUMN id SET DEFAULT nextval('public.comic\_reviews\_id\_seq'::regclass);  --  -- TOC entry 3344 (class 2604 OID 56136)  -- Name: comic\_types id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_types ALTER COLUMN id SET DEFAULT nextval('public.comic\_types\_id\_seq'::regclass);  --  -- TOC entry 3346 (class 2604 OID 56150)  -- Name: comics id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comics ALTER COLUMN id SET DEFAULT nextval('public.comics\_id\_seq'::regclass);  --  -- TOC entry 3353 (class 2604 OID 56336)  -- Name: creator\_roles id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.creator\_roles ALTER COLUMN id SET DEFAULT nextval('public.creator\_roles\_id\_seq'::regclass);  --  -- TOC entry 3354 (class 2604 OID 56343)  -- Name: creators id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.creators ALTER COLUMN id SET DEFAULT nextval('public.creators\_id\_seq'::regclass);  --  -- TOC entry 3348 (class 2604 OID 56202)  -- Name: genres id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.genres ALTER COLUMN id SET DEFAULT nextval('public.genres\_id\_seq'::regclass);  --  -- TOC entry 3334 (class 2604 OID 56021)  -- Name: migrations id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.migrations ALTER COLUMN id SET DEFAULT nextval('public.migrations\_id\_seq'::regclass);  --  -- TOC entry 3347 (class 2604 OID 56175)  -- Name: reading\_statuses id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.reading\_statuses ALTER COLUMN id SET DEFAULT nextval('public.reading\_statuses\_id\_seq'::regclass);  --  -- TOC entry 3345 (class 2604 OID 56143)  -- Name: release\_statuses id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.release\_statuses ALTER COLUMN id SET DEFAULT nextval('public.release\_statuses\_id\_seq'::regclass);  --  -- TOC entry 3338 (class 2604 OID 56054)  -- Name: roles id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.roles ALTER COLUMN id SET DEFAULT nextval('public.roles\_id\_seq'::regclass);  --  -- TOC entry 3337 (class 2604 OID 56042)  -- Name: subscription\_prices id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.subscription\_prices ALTER COLUMN id SET DEFAULT nextval('public.subscription\_prices\_id\_seq'::regclass);  --  -- TOC entry 3336 (class 2604 OID 56035)  -- Name: subscriptions id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.subscriptions ALTER COLUMN id SET DEFAULT nextval('public.subscriptions\_id\_seq'::regclass);  --  -- TOC entry 3343 (class 2604 OID 56129)  -- Name: translate\_statuses id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.translate\_statuses ALTER COLUMN id SET DEFAULT nextval('public.translate\_statuses\_id\_seq'::regclass);  --  -- TOC entry 3341 (class 2604 OID 56085)  -- Name: user\_subscriptions id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.user\_subscriptions ALTER COLUMN id SET DEFAULT nextval('public.user\_subscriptions\_id\_seq'::regclass);  --  -- TOC entry 3340 (class 2604 OID 56068)  -- Name: users id; Type: DEFAULT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.users ALTER COLUMN id SET DEFAULT nextval('public.users\_id\_seq'::regclass);  --  -- TOC entry 3592 (class 0 OID 56025)  -- Dependencies: 218  -- Data for Name: acquirings; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.acquirings (id, code, name) FROM stdin;  1 sber-aq sber-acquiring  2 alfa-aq alfa-acquiring  \.  --  -- TOC entry 3605 (class 0 OID 56103)  -- Dependencies: 231  -- Data for Name: auths; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.auths (token, user\_id) FROM stdin;  IG5icnRlbWJ2bS1ibHQ= 1  bWtucnRqZ25yZXZuIHI= 2  LG5iIGtrdHJl2a3RybA= 3  \.  --  -- TOC entry 3600 (class 0 OID 56058)  -- Dependencies: 226  -- Data for Name: avatars; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.avatars (id, path) FROM stdin;  1 /public/avatars/1.jpg  2 /public/avatars/2.jpg  3 /public/avatars/3.jpg  4 /public/avatars/4.jpg  5 /public/avatars/5.jpg  6 /public/avatars/6.jpg  7 /public/avatars/7.jpg  8 /public/avatars/8.jpg  9 /public/avatars/9.jpg  10 /public/avatars/10.jpg  11 /public/avatars/11.jpg  12 /public/avatars/Default.jpg  \.  --  -- TOC entry 3607 (class 0 OID 56114)  -- Dependencies: 233  -- Data for Name: bans; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.bans (id, date\_start, date\_end, reason\_description, user\_id) FROM stdin;  1 2024-05-05 2024-10-10 Использование стороннего ПО 4  \.  --  -- TOC entry 3631 (class 0 OID 56302)  -- Dependencies: 257  -- Data for Name: chapter\_review\_likes; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.chapter\_review\_likes (user\_id, chapter\_review\_id) FROM stdin;  1 1  2 2  3 3  4 4  1 5  2 6  3 7  4 8  1 9  2 10  3 11  \.  --  -- TOC entry 3630 (class 0 OID 56286)  -- Dependencies: 256  -- Data for Name: chapter\_reviews; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.chapter\_reviews (id, user\_id, chapter\_id, content, last\_update\_date, is\_moderated) FROM stdin;  1 3 3 Хахаха, Иона так ему ответила, что я чуть не проавлилась 🤣 2024-06-20 00:00:00 t  2 1 5 Ого, какой поворот сюжета! Никогда не думал, что она так поступит. 2024-06-24 00:00:00 t  3 2 10 Этот момент заставил меня задуматься над собственными поступками. Очень сильно. 2024-06-24 00:00:00 t  4 3 15 Эта глава меня просто потрясла. Я не мог перестать читать до самого конца! 2024-06-23 00:00:00 t  5 4 7 Вот это я называю персонаж с характером! Его поступки всегда удивляют и вдохновляют. 2024-06-22 00:00:00 t  6 2 3 С такими диалогами книга просто летит на одном дыхании. Невероятная игра слов! 2024-06-21 00:00:00 t  7 1 12 Этот момент заслуживает овации. Автору удалось создать настоящую эмоциональную бурю. 2024-06-20 00:00:00 t  8 3 18 Не могу поверить, что такой поворот случился! Жду продолжения с нетерпением. 2024-06-19 00:00:00 t  9 4 9 Какие персонажи! Каждый из них - отдельная история, каждый шаг - шаг к новым открытиям. 2024-06-18 00:00:00 t  10 2 20 Эта глава заставила меня задуматься о смысле жизни. Очень глубокая и философская. 2024-06-17 00:00:00 t  11 1 6 Концовка этой главы - полное откровение! Так много эмоций и неожиданностей. 2024-06-16 00:00:00 t  \.  --  -- TOC entry 3623 (class 0 OID 56221)  -- Dependencies: 249  -- Data for Name: chapters; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.chapters (id, comic\_id, name, chapter\_number, volume\_number, date\_added) FROM stdin;  1 1 Глава 1. Начало 1 1 2024-06-01  2 1 Глава 2. Первый подвиг 2 1 2024-06-05  3 1 Глава 3. Встреча с противником 3 1 2024-06-10  4 1 Глава 4. Испытание силы 4 1 2024-06-15  5 1 Глава 5. Загадочный союзник 5 1 2024-06-20  6 1 Глава 6. Подземные тропы 6 1 2024-06-25  7 1 Глава 7. Решающая схватка 7 1 2024-07-01  8 1 Глава 8. Возвращение героя 8 1 2024-07-05  9 1 Глава 9. Подарок судьбы 9 1 2024-07-10  10 1 Глава 10. Новое начало 10 1 2024-07-15  11 2 Глава 1. Тайная миссия 1 1 2024-06-01  12 2 Глава 2. Спасение союзников 2 1 2024-06-05  13 2 Глава 3. Погоня за артефактом 3 1 2024-06-10  14 2 Глава 4. Ловушка в городе 4 1 2024-06-15  15 2 Глава 5. Сражение в долине 5 1 2024-06-20  16 2 Глава 6. Загадочная пещера 6 1 2024-06-25  17 2 Глава 7. Встреча с предателем 7 1 2024-07-01  18 2 Глава 8. Решающий шаг 8 1 2024-07-05  19 2 Глава 9. Неожиданное союзничество 9 1 2024-07-10  20 2 Глава 10. Победа и предательство 10 1 2024-07-15  21 3 Глава 1. Загадка временного портала 1 1 2024-06-01  22 3 Глава 2. Путешествие в прошлое 2 1 2024-06-05  23 3 Глава 3. Встреча с историческим героем 3 1 2024-06-10  24 3 Глава 4. Злодей из будущего 4 1 2024-06-15  25 3 Глава 5. Разгадка временной паутины 5 1 2024-06-20  26 3 Глава 6. Схватка на рубеже эпох 6 1 2024-06-25  27 3 Глава 7. Поворот временных судеб 7 1 2024-07-01  28 3 Глава 8. Возвращение в настоящее 8 1 2024-07-05  29 3 Глава 9. Путешествие в будущее 9 1 2024-07-10  30 3 Глава 10. Великий временной круг 10 1 2024-07-15  \.  --  -- TOC entry 3637 (class 0 OID 56351)  -- Dependencies: 263  -- Data for Name: comic\_creators; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.comic\_creators (comic\_id, creator\_id) FROM stdin;  1 1  1 2  1 3  2 1  3 1  3 2  4 2  5 1  5 2  \.  --  -- TOC entry 3621 (class 0 OID 56205)  -- Dependencies: 247  -- Data for Name: comic\_genres; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.comic\_genres (comic\_id, genre\_id) FROM stdin;  1 1  1 2  1 3  1 4  1 5  1 6  1 7  1 8  1 9  1 10  1 11  2 9  2 2  2 3  2 4  2 8  2 10  3 12  3 8  3 2  3 13  3 1  4 14  4 15  4 16  4 2  4 17  4 18  5 14  5 15  5 2  5 16  5 17  5 18  6 12  6 2  6 16  6 14  \.  --  -- TOC entry 3626 (class 0 OID 56257)  -- Dependencies: 252  -- Data for Name: comic\_names; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.comic\_names (comic\_id, name, is\_default) FROM stdin;  1 Милый дом t  1 Sweet Home (HWANG Youngchan) f  1 스위트홈 f  1 홈 SWEET HOME f  1 홈스위트홈 f  1 Дом, милый дом f  1 홈스위트홈 f  1 홈 SWEET HOME f  1 Sweet Home f  1 ㅎㅗㅁ SWEET HOME f  1 Dulce Hogar f  1 ㅎㅗㅁ f  1 家 (金坎比) f  1 Seuwiteu Hom f  1 Sweet Home (KIM Carnby) f  2 Unknown Building f  2 Неизвестное здание t  2 未知的建筑 f  2 未知の建物 f  2 불명 건물 f  2 Building Tidak Dikenal f  2 未知的建築 f  2 ไม่ทราบสิ่งปลูกสร้าง f  2 अज्ञात इमारत f  2 בניין לא ידוע f  2 Gebäude Unbekannt f  3 Virtual Crisis f  3 Виртуальный кризис t  3 虚拟危机 f  3 バーチャルクライシス f  3 Virtual Krisis f  3 Krisis Virtual f  3 Krizis Virtual f  3 קריזת וירטואלי f  3 Crisi Virtuale f  4 Monster Hunter f  4 Охотник на монстров t  4 몬스터 헌터 f  4 猎兽人 f  4 モンスターハンター f  4 Cazador de Monstruos f  4 Monster Jäger f  4 Κυνηγός Τεράτων f  4 מסתורין מסתורין f  4 Cacciatore di Mostri f  5 War of Angels and Demons t  5 Война ангелов и демонов f  5 천사와 악마의 전쟁 f  5 天使与魔鬼之战 f  5 戦いの天使と悪魔 f  5 Guerra de Ángeles y Demonios f  5 Krieg der Engel und Dämonen f  5 Πόλεμος των Αγγέλων και Δαιμόνων f  5 מלחמת המלאכים והשדים f  5 Guerra tra Angeli e Demoni f  6 Tech Innovations f  6 Технологические инновации t  6 기술 혁신 f  6 技术创新 f  6 テクノイノベーション f  6 Innovaciones Tecnológicas f  6 Technologische Innovationen f  6 Τεχνολογικές Καινοτομίες f  6 חדשנות טכנולוגית f  6 Innovazioni Tecnologiche f  \.  --  -- TOC entry 3632 (class 0 OID 56317)  -- Dependencies: 258  -- Data for Name: comic\_ratings; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.comic\_ratings (user\_id, comic\_id, value) FROM stdin;  1 1 1  2 1 8  3 1 5  4 1 3  1 2 9  2 2 6  3 2 7  4 2 4  1 3 10  2 3 8  3 3 9  \.  --  -- TOC entry 3618 (class 0 OID 56178)  -- Dependencies: 244  -- Data for Name: comic\_reading\_statuses; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.comic\_reading\_statuses (user\_id, comic\_id, reading\_status\_id) FROM stdin;  1 1 1  2 1 1  3 1 1  \.  --  -- TOC entry 3628 (class 0 OID 56269)  -- Dependencies: 254  -- Data for Name: comic\_reviews; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.comic\_reviews (id, user\_id, comic\_id, content, last\_update\_date, is\_moderated) FROM stdin;  1 1 1 Интересный комикс, а самое главное - необычные развязки. Однозначно рекомендуется к прочтению 2024-06-20 00:00:00 t  2 2 1 Сюжет необычный, но сами персонажи не раскрыты. На один раз прочитать сойдёт 2024-06-20 00:00:00 t  3 3 1 Отличная атмосфера и увлекательный сюжет. Персонажи живые, каждый со своей историей. Рекомендую всем! 2024-06-20 00:00:00 t  4 4 1 Необычные идеи и интересные повороты событий, но слишком много диалогов. Визуальная составляющая тоже важна 2024-06-20 00:00:00 t  5 1 2 Фантастический мир и глубокие персонажи. Читал на одном дыхании! 2024-06-20 00:00:00 t  6 2 2 Сюжет сложный, персонажи интересные, но концовка разочаровала. Средне 2024-06-20 00:00:00 t  7 3 2 Удивительная смесь мистики и приключений. Персонажи проработаны до мелочей, читать стоит! 2024-06-20 00:00:00 t  8 4 2 Интересный мир, но немного запутанная сюжетная линия. Для фанатов жанра будет интересно 2024-06-20 00:00:00 t  9 1 3 Захватывающий комикс с неожиданными поворотами сюжета. Обязателен к прочтению! 2024-06-20 00:00:00 t  10 2 3 Сюжет интересный, но персонажи плоские. Подойдет для быстрого чтения 2024-06-20 00:00:00 t  11 3 3 Уникальная атмосфера и захватывающий сюжет. Персонажи с ярким характером. Однозначно рекомендую! 2024-06-20 00:00:00 t  \.  --  -- TOC entry 3611 (class 0 OID 56133)  -- Dependencies: 237  -- Data for Name: comic\_types; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.comic\_types (id, name) FROM stdin;  1 Манга  2 Манхва  3 Маньхуа  4 Западный комикс  5 Рукомикс  6 Индонезийский комикс  7 Другое  \.  --  -- TOC entry 3615 (class 0 OID 56147)  -- Dependencies: 241  -- Data for Name: comics; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.comics (id, description, release\_year, age\_limit, poster\_path, comic\_type\_id, release\_status\_id, translate\_status\_id) FROM stdin;  1 Много ли нужно для счастья хикикомори? Просто сиди в своих четырех стенах, отгоняй всех и погружайся в свой компьютер. Но что, если внезапно родители с сестрой умирают? Ну и пусть, они оставили "копейки". Пришлось переехать в дешевую общагу, питаться фастфудом, и даже назначить день Х, чтобы покинуть этот мир. Однако наш герой не подозревал, что умереть человеческими методами ему не суждено. Ведь, как говорится, "ночь темна и полна ужасов", а днем — тем более. 2018 16+ https://remanga.org/media/titles/sweet\_home\_hwang\_youngchan/8526d40b663441e5189f6263cefca146.jpg 2 1 1  2 После того, как таинственный вирус начал превращать людей в живых мертвецов, мир погрузился в хаос. Группа выживших оказалась заблокированной в темном подвале высотного здания. Они вынуждены объединить усилия, чтобы найти выход и выжить в этом аду. Но чем дольше они находятся в закрытом пространстве, тем сильнее начинают проявляться их внутренние конфликты. Они должны принять сложные решения, чтобы сохранить свою человечность и обрести новую надежду в мире, погруженном в ужас и беспомощность. 2022 18+ https://example.com/media/titles/unknown\_building/poster.jpg 1 2 2  3 В современном мире, где искусственный интеллект уже вовсю интегрирован в повседневную жизнь, молодой программист сталкивается с загадочным вирусом, который начинает заражать целые компьютерные системы, превращая их в орудия хаоса. В попытках разгадать загадку заражения, он вынужден исследовать самые глубокие уровни виртуальной реальности, где каждый шаг открывает новые тайны и угрозы. Его борьба с вирусом ставит под угрозу не только его собственную жизнь, но и будущее всего человечества. 2023 16+ https://example.com/media/titles/virtual\_crisis/poster.jpg 2 2 2  4 В мире, где люди и монстры соседствуют, мало кто знает о темных силах, скрывающихся под поверхностью. Молодой охотник на монстров, чья семья погибла от нападения неизвестного зверя, отправляется в опасное путешествие, чтобы разгадать тайну прошлого и защитить мир от новой угрозы. Его путешествие наполнено сражениями, загадками и встречами с загадочными существами, которые меняют его представление о добре и зле. 2023 16+ https://example.com/media/titles/monster\_hunter/poster.jpg 1 2 2  5 В мире, раздираемом войной между ангелами и демонами, группа обычных людей оказывается втянутой в борьбу за выживание. Они обладают уникальными способностями, способными изменить ход событий. Им предстоит раскрыть свои силы, принять решение — спасти мир или погибнуть в его пучинах. Их действия определят будущее всего человечества. 2022 16+ https://example.com/media/titles/angel\_demon\_war/poster.jpg 1 2 2  6 В далеком будущем, населенном технологиями и роботами, молодой инженер ведет исследования в области искусственного интеллекта. Он создает уникального робота, который обладает человекоподобными чувствами и эмоциями. Вместе они исследуют мир, встречаются с различными опасностями и решают глобальные проблемы, стоящие перед человечеством. 2023 12+ https://example.com/media/titles/future\_tech\_robot/poster.jpg 1 2 2  \.  --  -- TOC entry 3634 (class 0 OID 56333)  -- Dependencies: 260  -- Data for Name: creator\_roles; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.creator\_roles (id, name) FROM stdin;  1 Издатель  2 Автор  3 Художник  \.  --  -- TOC entry 3636 (class 0 OID 56340)  -- Dependencies: 262  -- Data for Name: creators; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.creators (id, name, creator\_role\_id) FROM stdin;  1 Webtoon 1  2 Naver 1  3 Carnby 2  \.  --  -- TOC entry 3624 (class 0 OID 56232)  -- Dependencies: 250  -- Data for Name: frames; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.frames (chapter\_id, number, path) FROM stdin;  1 1 /public/chapter/1/1.jpg  1 2 /public/chapter/1/2.jpg  2 1 /public/chapter/2/1.jpg  2 2 /public/chapter/2/2.jpg  3 1 /public/chapter/3/1.jpg  3 2 /public/chapter/3/2.jpg  3 3 /public/chapter/3/3.jpg  4 1 /public/chapter/4/1.jpg  5 1 /public/chapter/5/1.jpg  5 2 /public/chapter/5/2.jpg  5 3 /public/chapter/5/3.jpg  6 1 /public/chapter/6/1.jpg  6 2 /public/chapter/6/2.jpg  7 1 /public/chapter/7/1.jpg  7 2 /public/chapter/7/2.jpg  8 1 /public/chapter/8/1.jpg  9 1 /public/chapter/9/1.jpg  9 2 /public/chapter/9/2.jpg  9 3 /public/chapter/9/3.jpg  10 1 /public/chapter/10/1.jpg  \.  --  -- TOC entry 3620 (class 0 OID 56199)  -- Dependencies: 246  -- Data for Name: genres; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.genres (id, name) FROM stdin;  1 Экшен  2 Драма  3 Постапокалиптика  4 Психология  5 Романтика  6 Сверхестественное  7 Сэйнэн  8 Триллер  9 Ужасы  10 Фантастика  11 Криминал  12 Научная фантастика  13 Киберпанк  14 Фэнтези  15 Боевик  16 Приключения  17 Мистика  18 Сверхъестественное  \.  --  -- TOC entry 3625 (class 0 OID 56242)  -- Dependencies: 251  -- Data for Name: like\_chapters; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.like\_chapters (user\_id, chapter\_id) FROM stdin;  1 1  2 2  3 3  4 4  1 5  2 6  3 7  4 8  1 9  2 10  3 11  \.  --  -- TOC entry 3590 (class 0 OID 56018)  -- Dependencies: 216  -- Data for Name: migrations; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.migrations (id, migration, batch) FROM stdin;  1 2024\_06\_23\_072636\_create\_acquirings\_table 1  2 2024\_06\_23\_072711\_create\_subscriptions\_table 1  3 2024\_06\_23\_072714\_create\_subscription\_prices\_table 1  4 2024\_06\_23\_072735\_create\_roles\_table 1  5 2024\_06\_23\_072742\_create\_avatars\_table 1  6 2024\_06\_23\_072743\_create\_users\_table 1  7 2024\_06\_23\_072800\_create\_user\_subscriptions\_table 1  8 2024\_06\_23\_072811\_create\_auths\_table 1  9 2024\_06\_23\_072816\_create\_bans\_table 1  10 2024\_06\_23\_073512\_create\_translate\_statuses\_table 1  11 2024\_06\_23\_073534\_create\_comic\_types\_table 1  12 2024\_06\_23\_073535\_create\_release\_statuses\_table 1  13 2024\_06\_23\_073542\_create\_comics\_table 1  14 2024\_06\_23\_073542\_create\_reading\_statuses\_table 1  15 2024\_06\_23\_073543\_create\_comic\_reading\_statuses\_table 1  16 2024\_06\_23\_073555\_create\_genres\_table 1  17 2024\_06\_23\_073607\_create\_comic\_genres\_table 1  18 2024\_06\_23\_073623\_create\_chapters\_table 1  19 2024\_06\_23\_073635\_create\_frames\_table 1  20 2024\_06\_23\_073658\_create\_like\_chapters\_table 1  21 2024\_06\_23\_073712\_create\_comic\_names\_table 1  22 2024\_06\_23\_073730\_create\_comic\_reviews\_table 1  23 2024\_06\_23\_073752\_create\_chapter\_reviews\_table 1  24 2024\_06\_23\_073753\_create\_chapter\_review\_likes\_table 1  25 2024\_06\_23\_073807\_create\_comic\_ratings\_table 1  26 2024\_06\_25\_025343\_create\_creator\_roles\_table 1  27 2024\_06\_25\_025354\_create\_creators\_table 1  28 2024\_06\_25\_025402\_create\_comic\_creators\_table 1  \.  --  -- TOC entry 3617 (class 0 OID 56172)  -- Dependencies: 243  -- Data for Name: reading\_statuses; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.reading\_statuses (id, name) FROM stdin;  1 Читаю  2 Отложено  3 Брошено  4 Прочитано  \.  --  -- TOC entry 3613 (class 0 OID 56140)  -- Dependencies: 239  -- Data for Name: release\_statuses; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.release\_statuses (id, name) FROM stdin;  1 Закончен  2 Продолжается  \.  --  -- TOC entry 3598 (class 0 OID 56051)  -- Dependencies: 224  -- Data for Name: roles; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.roles (id, code) FROM stdin;  1 admin  2 user  \.  --  -- TOC entry 3596 (class 0 OID 56039)  -- Dependencies: 222  -- Data for Name: subscription\_prices; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.subscription\_prices (id, price, discount\_percent, days\_count, subscription\_id) FROM stdin;  1 299.99 0 30 1  2 799.99 5 90 2  3 1499.99 10 180 3  4 2499.99 15 365 4  5 499.99 0 7 5  \.  --  -- TOC entry 3594 (class 0 OID 56032)  -- Dependencies: 220  -- Data for Name: subscriptions; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.subscriptions (id, is\_available, name, description) FROM stdin;  1 t Месячная подписка Доступ к манге и манхве на 30 дней  2 t Тримесячная подписка Доступ к манге и манхве на 90 дней с дополнительными скидками  3 t Полугодовая подписка Доступ к манге и манхве на 180 дней, включая доступ ко всем новинкам  4 t Годовая подписка Полный доступ к манге и манхве на 365 дней, включая эксклюзивные выпуски и специальные предложения  5 t Недельная пробная подписка Подписка на мангу и манхву на 7 дней для ознакомления  \.  --  -- TOC entry 3609 (class 0 OID 56126)  -- Dependencies: 235  -- Data for Name: translate\_statuses; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.translate\_statuses (id, name) FROM stdin;  1 Закончен  2 Ожидает перевода  \.  --  -- TOC entry 3604 (class 0 OID 56082)  -- Dependencies: 230  -- Data for Name: user\_subscriptions; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.user\_subscriptions (id, user\_id, acquiring\_id, subscription\_price\_id, date\_payment, date\_end, date\_cancel, pay\_amount, refund\_amount, id\_payment) FROM stdin;  1 1 1 1 2024-05-05 00:00:00 2024-05-06 00:00:00 \N 399.99 \N aGRmc2hnc2Rmamhmc2RoZGZo  2 2 2 1 2024-05-04 00:00:00 2024-05-05 00:00:00 \N 399.99 \N aGdmamFkZ3Zhc2RnZmdqZ2ZkaGRmZw  \.  --  -- TOC entry 3602 (class 0 OID 56065)  -- Dependencies: 228  -- Data for Name: users; Type: TABLE DATA; Schema: public; Owner: -  --  COPY public.users (id, nickname, email, password, avatar\_id, role\_id) FROM stdin;  1 admin admin@mail.ru $2y$12$8g/Pco0bCqvVrk1MmoDYW.YDdwtg1dUCMJh57.GO1O3YOwtKuKKSe 1 1  2 chebyrek chebyrek@mail.ru $2y$12$p6Ntkzd.DJGHzGWqu6B0HOCh/TswDs7RYuHvl2ICDkM6Ixn5lH0Rm 2 2  3 odyvan odyvan@mail.ru $2y$12$faqn3bVSThe6y5j2xqWmbe58.CsKZ98EbsJl7H7MqL0tRsFYyDUMi 3 2  4 bannedUser bannedUser@mail.ru $2y$12$dnScfS6mz.3YEbjSr.v0ceYy8w0Qtcl1w4LISlwbRrXuqGrW1yE6. 10 2  \.  --  -- TOC entry 3664 (class 0 OID 0)  -- Dependencies: 217  -- Name: acquirings\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.acquirings\_id\_seq', 2, true);  --  -- TOC entry 3665 (class 0 OID 0)  -- Dependencies: 225  -- Name: avatars\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.avatars\_id\_seq', 12, true);  --  -- TOC entry 3666 (class 0 OID 0)  -- Dependencies: 232  -- Name: bans\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.bans\_id\_seq', 1, true);  --  -- TOC entry 3667 (class 0 OID 0)  -- Dependencies: 255  -- Name: chapter\_reviews\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.chapter\_reviews\_id\_seq', 11, true);  --  -- TOC entry 3668 (class 0 OID 0)  -- Dependencies: 248  -- Name: chapters\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.chapters\_id\_seq', 30, true);  --  -- TOC entry 3669 (class 0 OID 0)  -- Dependencies: 253  -- Name: comic\_reviews\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.comic\_reviews\_id\_seq', 11, true);  --  -- TOC entry 3670 (class 0 OID 0)  -- Dependencies: 236  -- Name: comic\_types\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.comic\_types\_id\_seq', 7, true);  --  -- TOC entry 3671 (class 0 OID 0)  -- Dependencies: 240  -- Name: comics\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.comics\_id\_seq', 6, true);  --  -- TOC entry 3672 (class 0 OID 0)  -- Dependencies: 259  -- Name: creator\_roles\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.creator\_roles\_id\_seq', 3, true);  --  -- TOC entry 3673 (class 0 OID 0)  -- Dependencies: 261  -- Name: creators\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.creators\_id\_seq', 3, true);  --  -- TOC entry 3674 (class 0 OID 0)  -- Dependencies: 245  -- Name: genres\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.genres\_id\_seq', 18, true);  --  -- TOC entry 3675 (class 0 OID 0)  -- Dependencies: 215  -- Name: migrations\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.migrations\_id\_seq', 28, true);  --  -- TOC entry 3676 (class 0 OID 0)  -- Dependencies: 242  -- Name: reading\_statuses\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.reading\_statuses\_id\_seq', 4, true);  --  -- TOC entry 3677 (class 0 OID 0)  -- Dependencies: 238  -- Name: release\_statuses\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.release\_statuses\_id\_seq', 2, true);  --  -- TOC entry 3678 (class 0 OID 0)  -- Dependencies: 223  -- Name: roles\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.roles\_id\_seq', 2, true);  --  -- TOC entry 3679 (class 0 OID 0)  -- Dependencies: 221  -- Name: subscription\_prices\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.subscription\_prices\_id\_seq', 5, true);  --  -- TOC entry 3680 (class 0 OID 0)  -- Dependencies: 219  -- Name: subscriptions\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.subscriptions\_id\_seq', 5, true);  --  -- TOC entry 3681 (class 0 OID 0)  -- Dependencies: 234  -- Name: translate\_statuses\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.translate\_statuses\_id\_seq', 2, true);  --  -- TOC entry 3682 (class 0 OID 0)  -- Dependencies: 229  -- Name: user\_subscriptions\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.user\_subscriptions\_id\_seq', 2, true);  --  -- TOC entry 3683 (class 0 OID 0)  -- Dependencies: 227  -- Name: users\_id\_seq; Type: SEQUENCE SET; Schema: public; Owner: -  --  SELECT pg\_catalog.setval('public.users\_id\_seq', 4, true);  --  -- TOC entry 3359 (class 2606 OID 56030)  -- Name: acquirings acquirings\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.acquirings  ADD CONSTRAINT acquirings\_pkey PRIMARY KEY (id);  --  -- TOC entry 3373 (class 2606 OID 56112)  -- Name: auths auths\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.auths  ADD CONSTRAINT auths\_pkey PRIMARY KEY (token);  --  -- TOC entry 3367 (class 2606 OID 56063)  -- Name: avatars avatars\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.avatars  ADD CONSTRAINT avatars\_pkey PRIMARY KEY (id);  --  -- TOC entry 3375 (class 2606 OID 56124)  -- Name: bans bans\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.bans  ADD CONSTRAINT bans\_pkey PRIMARY KEY (id);  --  -- TOC entry 3405 (class 2606 OID 56316)  -- Name: chapter\_review\_likes chapter\_review\_likes\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapter\_review\_likes  ADD CONSTRAINT chapter\_review\_likes\_pkey PRIMARY KEY (user\_id, chapter\_review\_id);  --  -- TOC entry 3403 (class 2606 OID 56301)  -- Name: chapter\_reviews chapter\_reviews\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapter\_reviews  ADD CONSTRAINT chapter\_reviews\_pkey PRIMARY KEY (id);  --  -- TOC entry 3393 (class 2606 OID 56231)  -- Name: chapters chapters\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapters  ADD CONSTRAINT chapters\_pkey PRIMARY KEY (id);  --  -- TOC entry 3413 (class 2606 OID 56365)  -- Name: comic\_creators comic\_creators\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_creators  ADD CONSTRAINT comic\_creators\_pkey PRIMARY KEY (comic\_id, creator\_id);  --  -- TOC entry 3391 (class 2606 OID 56219)  -- Name: comic\_genres comic\_genres\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_genres  ADD CONSTRAINT comic\_genres\_pkey PRIMARY KEY (comic\_id, genre\_id);  --  -- TOC entry 3399 (class 2606 OID 56267)  -- Name: comic\_names comic\_names\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_names  ADD CONSTRAINT comic\_names\_pkey PRIMARY KEY (comic\_id, name);  --  -- TOC entry 3407 (class 2606 OID 56331)  -- Name: comic\_ratings comic\_ratings\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_ratings  ADD CONSTRAINT comic\_ratings\_pkey PRIMARY KEY (user\_id, comic\_id);  --  -- TOC entry 3387 (class 2606 OID 56197)  -- Name: comic\_reading\_statuses comic\_reading\_statuses\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_reading\_statuses  ADD CONSTRAINT comic\_reading\_statuses\_pkey PRIMARY KEY (user\_id, comic\_id);  --  -- TOC entry 3401 (class 2606 OID 56284)  -- Name: comic\_reviews comic\_reviews\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_reviews  ADD CONSTRAINT comic\_reviews\_pkey PRIMARY KEY (id);  --  -- TOC entry 3379 (class 2606 OID 56138)  -- Name: comic\_types comic\_types\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_types  ADD CONSTRAINT comic\_types\_pkey PRIMARY KEY (id);  --  -- TOC entry 3383 (class 2606 OID 56155)  -- Name: comics comics\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comics  ADD CONSTRAINT comics\_pkey PRIMARY KEY (id);  --  -- TOC entry 3409 (class 2606 OID 56338)  -- Name: creator\_roles creator\_roles\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.creator\_roles  ADD CONSTRAINT creator\_roles\_pkey PRIMARY KEY (id);  --  -- TOC entry 3411 (class 2606 OID 56345)  -- Name: creators creators\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.creators  ADD CONSTRAINT creators\_pkey PRIMARY KEY (id);  --  -- TOC entry 3395 (class 2606 OID 56241)  -- Name: frames frames\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.frames  ADD CONSTRAINT frames\_pkey PRIMARY KEY (chapter\_id, number);  --  -- TOC entry 3389 (class 2606 OID 56204)  -- Name: genres genres\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.genres  ADD CONSTRAINT genres\_pkey PRIMARY KEY (id);  --  -- TOC entry 3397 (class 2606 OID 56256)  -- Name: like\_chapters like\_chapters\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.like\_chapters  ADD CONSTRAINT like\_chapters\_pkey PRIMARY KEY (user\_id, chapter\_id);  --  -- TOC entry 3357 (class 2606 OID 56023)  -- Name: migrations migrations\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.migrations  ADD CONSTRAINT migrations\_pkey PRIMARY KEY (id);  --  -- TOC entry 3385 (class 2606 OID 56177)  -- Name: reading\_statuses reading\_statuses\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.reading\_statuses  ADD CONSTRAINT reading\_statuses\_pkey PRIMARY KEY (id);  --  -- TOC entry 3381 (class 2606 OID 56145)  -- Name: release\_statuses release\_statuses\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.release\_statuses  ADD CONSTRAINT release\_statuses\_pkey PRIMARY KEY (id);  --  -- TOC entry 3365 (class 2606 OID 56056)  -- Name: roles roles\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.roles  ADD CONSTRAINT roles\_pkey PRIMARY KEY (id);  --  -- TOC entry 3363 (class 2606 OID 56049)  -- Name: subscription\_prices subscription\_prices\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.subscription\_prices  ADD CONSTRAINT subscription\_prices\_pkey PRIMARY KEY (id);  --  -- TOC entry 3361 (class 2606 OID 56037)  -- Name: subscriptions subscriptions\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.subscriptions  ADD CONSTRAINT subscriptions\_pkey PRIMARY KEY (id);  --  -- TOC entry 3377 (class 2606 OID 56131)  -- Name: translate\_statuses translate\_statuses\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.translate\_statuses  ADD CONSTRAINT translate\_statuses\_pkey PRIMARY KEY (id);  --  -- TOC entry 3371 (class 2606 OID 56102)  -- Name: user\_subscriptions user\_subscriptions\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.user\_subscriptions  ADD CONSTRAINT user\_subscriptions\_pkey PRIMARY KEY (id);  --  -- TOC entry 3369 (class 2606 OID 56070)  -- Name: users users\_pkey; Type: CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.users  ADD CONSTRAINT users\_pkey PRIMARY KEY (id);  --  -- TOC entry 3420 (class 2606 OID 56106)  -- Name: auths auths\_user\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.auths  ADD CONSTRAINT auths\_user\_id\_foreign FOREIGN KEY (user\_id) REFERENCES public.users(id) ON UPDATE CASCADE;  --  -- TOC entry 3421 (class 2606 OID 56118)  -- Name: bans bans\_user\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.bans  ADD CONSTRAINT bans\_user\_id\_foreign FOREIGN KEY (user\_id) REFERENCES public.users(id) ON UPDATE CASCADE;  --  -- TOC entry 3439 (class 2606 OID 56310)  -- Name: chapter\_review\_likes chapter\_review\_likes\_chapter\_review\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapter\_review\_likes  ADD CONSTRAINT chapter\_review\_likes\_chapter\_review\_id\_foreign FOREIGN KEY (chapter\_review\_id) REFERENCES public.chapter\_reviews(id) ON UPDATE CASCADE;  --  -- TOC entry 3440 (class 2606 OID 56305)  -- Name: chapter\_review\_likes chapter\_review\_likes\_user\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapter\_review\_likes  ADD CONSTRAINT chapter\_review\_likes\_user\_id\_foreign FOREIGN KEY (user\_id) REFERENCES public.users(id) ON UPDATE CASCADE;  --  -- TOC entry 3437 (class 2606 OID 56295)  -- Name: chapter\_reviews chapter\_reviews\_chapter\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapter\_reviews  ADD CONSTRAINT chapter\_reviews\_chapter\_id\_foreign FOREIGN KEY (chapter\_id) REFERENCES public.chapters(id) ON UPDATE CASCADE;  --  -- TOC entry 3438 (class 2606 OID 56290)  -- Name: chapter\_reviews chapter\_reviews\_user\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapter\_reviews  ADD CONSTRAINT chapter\_reviews\_user\_id\_foreign FOREIGN KEY (user\_id) REFERENCES public.users(id) ON UPDATE CASCADE;  --  -- TOC entry 3430 (class 2606 OID 56225)  -- Name: chapters chapters\_comic\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.chapters  ADD CONSTRAINT chapters\_comic\_id\_foreign FOREIGN KEY (comic\_id) REFERENCES public.comics(id) ON UPDATE CASCADE;  --  -- TOC entry 3444 (class 2606 OID 56354)  -- Name: comic\_creators comic\_creators\_comic\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_creators  ADD CONSTRAINT comic\_creators\_comic\_id\_foreign FOREIGN KEY (comic\_id) REFERENCES public.comics(id) ON UPDATE CASCADE;  --  -- TOC entry 3445 (class 2606 OID 56359)  -- Name: comic\_creators comic\_creators\_creator\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_creators  ADD CONSTRAINT comic\_creators\_creator\_id\_foreign FOREIGN KEY (creator\_id) REFERENCES public.creators(id) ON UPDATE CASCADE;  --  -- TOC entry 3428 (class 2606 OID 56208)  -- Name: comic\_genres comic\_genres\_comic\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_genres  ADD CONSTRAINT comic\_genres\_comic\_id\_foreign FOREIGN KEY (comic\_id) REFERENCES public.comics(id) ON UPDATE CASCADE;  --  -- TOC entry 3429 (class 2606 OID 56213)  -- Name: comic\_genres comic\_genres\_genre\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_genres  ADD CONSTRAINT comic\_genres\_genre\_id\_foreign FOREIGN KEY (genre\_id) REFERENCES public.genres(id) ON UPDATE CASCADE;  --  -- TOC entry 3434 (class 2606 OID 56261)  -- Name: comic\_names comic\_names\_comic\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_names  ADD CONSTRAINT comic\_names\_comic\_id\_foreign FOREIGN KEY (comic\_id) REFERENCES public.comics(id) ON UPDATE CASCADE;  --  -- TOC entry 3441 (class 2606 OID 56325)  -- Name: comic\_ratings comic\_ratings\_comic\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_ratings  ADD CONSTRAINT comic\_ratings\_comic\_id\_foreign FOREIGN KEY (comic\_id) REFERENCES public.comics(id) ON UPDATE CASCADE;  --  -- TOC entry 3442 (class 2606 OID 56320)  -- Name: comic\_ratings comic\_ratings\_user\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_ratings  ADD CONSTRAINT comic\_ratings\_user\_id\_foreign FOREIGN KEY (user\_id) REFERENCES public.users(id) ON UPDATE CASCADE;  --  -- TOC entry 3425 (class 2606 OID 56186)  -- Name: comic\_reading\_statuses comic\_reading\_statuses\_comic\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_reading\_statuses  ADD CONSTRAINT comic\_reading\_statuses\_comic\_id\_foreign FOREIGN KEY (comic\_id) REFERENCES public.comics(id) ON UPDATE CASCADE;  --  -- TOC entry 3426 (class 2606 OID 56191)  -- Name: comic\_reading\_statuses comic\_reading\_statuses\_reading\_status\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_reading\_statuses  ADD CONSTRAINT comic\_reading\_statuses\_reading\_status\_id\_foreign FOREIGN KEY (reading\_status\_id) REFERENCES public.reading\_statuses(id) ON UPDATE CASCADE;  --  -- TOC entry 3427 (class 2606 OID 56181)  -- Name: comic\_reading\_statuses comic\_reading\_statuses\_user\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_reading\_statuses  ADD CONSTRAINT comic\_reading\_statuses\_user\_id\_foreign FOREIGN KEY (user\_id) REFERENCES public.users(id) ON UPDATE CASCADE;  --  -- TOC entry 3435 (class 2606 OID 56278)  -- Name: comic\_reviews comic\_reviews\_comic\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_reviews  ADD CONSTRAINT comic\_reviews\_comic\_id\_foreign FOREIGN KEY (comic\_id) REFERENCES public.comics(id) ON UPDATE CASCADE;  --  -- TOC entry 3436 (class 2606 OID 56273)  -- Name: comic\_reviews comic\_reviews\_user\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comic\_reviews  ADD CONSTRAINT comic\_reviews\_user\_id\_foreign FOREIGN KEY (user\_id) REFERENCES public.users(id) ON UPDATE CASCADE;  --  -- TOC entry 3422 (class 2606 OID 56156)  -- Name: comics comics\_comic\_type\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comics  ADD CONSTRAINT comics\_comic\_type\_id\_foreign FOREIGN KEY (comic\_type\_id) REFERENCES public.comic\_types(id) ON UPDATE CASCADE;  --  -- TOC entry 3423 (class 2606 OID 56161)  -- Name: comics comics\_release\_status\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comics  ADD CONSTRAINT comics\_release\_status\_id\_foreign FOREIGN KEY (release\_status\_id) REFERENCES public.release\_statuses(id) ON UPDATE CASCADE;  --  -- TOC entry 3424 (class 2606 OID 56166)  -- Name: comics comics\_translate\_status\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.comics  ADD CONSTRAINT comics\_translate\_status\_id\_foreign FOREIGN KEY (translate\_status\_id) REFERENCES public.translate\_statuses(id) ON UPDATE CASCADE;  --  -- TOC entry 3443 (class 2606 OID 56346)  -- Name: creators creators\_creator\_role\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.creators  ADD CONSTRAINT creators\_creator\_role\_id\_foreign FOREIGN KEY (creator\_role\_id) REFERENCES public.creator\_roles(id) ON UPDATE CASCADE;  --  -- TOC entry 3431 (class 2606 OID 56235)  -- Name: frames frames\_chapter\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.frames  ADD CONSTRAINT frames\_chapter\_id\_foreign FOREIGN KEY (chapter\_id) REFERENCES public.chapters(id) ON UPDATE CASCADE;  --  -- TOC entry 3432 (class 2606 OID 56250)  -- Name: like\_chapters like\_chapters\_chapter\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.like\_chapters  ADD CONSTRAINT like\_chapters\_chapter\_id\_foreign FOREIGN KEY (chapter\_id) REFERENCES public.chapters(id) ON UPDATE CASCADE;  --  -- TOC entry 3433 (class 2606 OID 56245)  -- Name: like\_chapters like\_chapters\_user\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.like\_chapters  ADD CONSTRAINT like\_chapters\_user\_id\_foreign FOREIGN KEY (user\_id) REFERENCES public.users(id) ON UPDATE CASCADE;  --  -- TOC entry 3414 (class 2606 OID 56043)  -- Name: subscription\_prices subscription\_prices\_subscription\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.subscription\_prices  ADD CONSTRAINT subscription\_prices\_subscription\_id\_foreign FOREIGN KEY (subscription\_id) REFERENCES public.subscriptions(id) ON UPDATE CASCADE;  --  -- TOC entry 3417 (class 2606 OID 56091)  -- Name: user\_subscriptions user\_subscriptions\_acquiring\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.user\_subscriptions  ADD CONSTRAINT user\_subscriptions\_acquiring\_id\_foreign FOREIGN KEY (acquiring\_id) REFERENCES public.acquirings(id) ON UPDATE CASCADE;  --  -- TOC entry 3418 (class 2606 OID 56096)  -- Name: user\_subscriptions user\_subscriptions\_subscription\_price\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.user\_subscriptions  ADD CONSTRAINT user\_subscriptions\_subscription\_price\_id\_foreign FOREIGN KEY (subscription\_price\_id) REFERENCES public.subscription\_prices(id) ON UPDATE CASCADE;  --  -- TOC entry 3419 (class 2606 OID 56086)  -- Name: user\_subscriptions user\_subscriptions\_user\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.user\_subscriptions  ADD CONSTRAINT user\_subscriptions\_user\_id\_foreign FOREIGN KEY (user\_id) REFERENCES public.users(id) ON UPDATE CASCADE;  --  -- TOC entry 3415 (class 2606 OID 56071)  -- Name: users users\_avatar\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.users  ADD CONSTRAINT users\_avatar\_id\_foreign FOREIGN KEY (avatar\_id) REFERENCES public.avatars(id) ON UPDATE CASCADE;  --  -- TOC entry 3416 (class 2606 OID 56076)  -- Name: users users\_role\_id\_foreign; Type: FK CONSTRAINT; Schema: public; Owner: -  --  ALTER TABLE ONLY public.users  ADD CONSTRAINT users\_role\_id\_foreign FOREIGN KEY (role\_id) REFERENCES public.roles(id) ON UPDATE CASCADE;  -- Completed on 2024-06-25 22:55:04 +07  --  -- PostgreSQL database dump complete  -- |

# Заключение

В результате разработки базы данных для платформы чтения комиксов с функциями социального взаимодействия были достигнуты следующие задачи:

* Создана структура данных, которая эффективно обеспечивает хранение информации о пользователях, комиксах и их взаимодействиях.
* Определены необходимые таблицы, поля и связи для обеспечения удобства доступа.
* Разработаны механизмы безопасности, включая хеширование паролей, что обеспечивает защиту конфиденциальной информации пользователей.
* Реализованы функции социального взаимодействия, такие как комментарии, оценки и лайки, способствующие активному взаимодействию пользователей.

Таким образом, проект успешно реализовал поставленные задачи, что способствовало достижению основной цели разработки, а именно обеспечение эффективного хранения, обработки и дальнейшего предоставления информации о пользователях, их действиях и взаимодействиях на платформе.

# Источники

## *Интернет-ресурсы:*

1. W3Schools. "Учебник по SQL." https://www.w3schools.com/sql/. Дата доступа: 25 июня 2024 г.
2. Stack Overflow. "Лучшие практики проектирования баз данных." <https://ru.stackoverflow.com/questions/526834/>. Дата доступа: 25 июня 2024 г.
3. Medium. "Лучшие практики безопасности баз данных." https://medium.com/@security/sql-database-security-best-practices-8f7d605e87a7. Дата доступа: 25 июня 2024 г.
4. GitHub. "Примеры проектирования баз данных." <https://github.com/dbstewardship/database-design-examples>. Дата доступа: 25 июня 2024 г.

## *Литература:*

1. *Коннолли, Томас М., и Кэролин Э. Бегг. "Системы баз данных: Практический подход к проектированию, реализации и управлению." 6-е изд., Питер, 2014.*
2. Дейт, К. Дж. "Введение в системы баз данных." 8-е изд., Питер, 2003.
3. Эльмасри, Рамез, и Шамкант Б. Наватхе. "Основы систем баз данных." 7-е изд., Питер, 2015.
4. Роб, Питер, Карлос Коронель и Стивен Моррис. "Системы управления базами данных: Проектирование, реализация и управление." 13-е изд., Вильямс, 2020.
5. Шевцов, А. "Реляционная модель данных." БХВ-Петербург, 2007.