A black and white drawing of a building

Description automatically generated

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського» ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

# Кафедра системного програмування та спеціалізованих комп’ютерних систем

**Лабораторна робота №1**

з дисципліни **Бази даних і засоби управління**

*на тему: “Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL”*

Виконав:

студент ІІI курсу

групи КВ-12

Бичко К.В.

Перевірив:

Павловский В. І.

Київ – 2023

**Мета:** здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

**Завдання:**

1. Розробити модель «сутність-зв’язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

**Опис предметної області**

Дана предметна область – система продажу квитків на громадський транспорт, а точніше - залізницю. Вона охоплює сукупність предметів та інструментів, які спрямовані на ведення обліку продажів квитків та потягів за різними маршрутами. Це допоможе моніторити кількість пасажирів, прибутків та формувати звітність згідно до вимог.

**Опис сутностей**

Для побудови бази даних обраної області, були виділені такі сутності:

1. Потяг (Train)

Атрибути: номер потягу, кількість місць.

Призначення: збереження даних щодо маршрутів по яким курсують

потяги.

1. Квиток (Ticket)

Атрибути: ідентифікатор квитка, ціна, ПІБ пасажира.

Призначення: збереження даних щодо квитків та проданих місць за

напрямком.

1. Станція (Station)

Атрибути: ідентифікатор станції, назва станції.

Призначення: збереження даних щодо станцій та курсуючих по ній потягів.

**Опис зв’язків між сутностями**

Зв’язок-сутність «Потяг» - «Станція» є зв’язком N:M. В конкретну Дату один потяг може їхати через багато станцій, і одна станція може приймати багато потягів.

Зв’язок «Потяг-Станція» - «Квиток» є зв’язком 1:N. Один квиток продається на багато пар Потяг-Станція. З цього автоматично випливає:

Зв’язок «Потяг» - «Квиток» є зв’язком 1:N. Один квиток може бути проданим лише на один потяг, але на один потяг можна продати багато квитків.

Зв’язок «Станція» - «Квиток» є зв’язком 1:N. Один квиток може бути проданим лише до однієї станції, але до однієї станції може продаватися багато квитків.

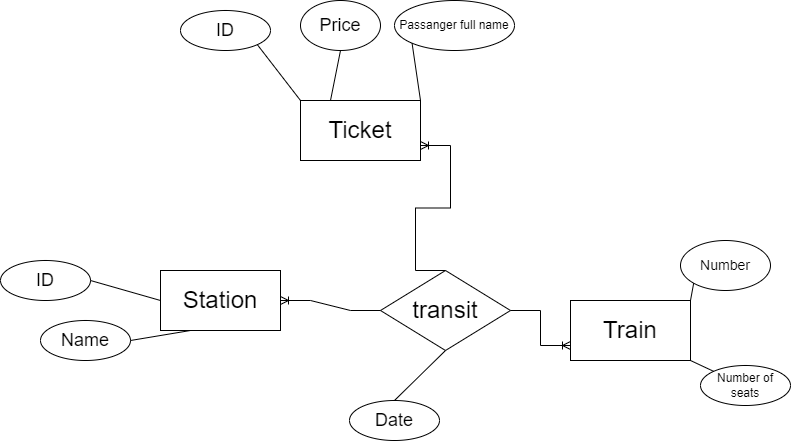


Рисунок 1 – ER-діаграма, побудована за нотацією Чена

**Перетворення концептуальної моделі у логічну модель та схему бази даних**

Сутність Station перетворено в таблицю Station з первинним ключем ID та атрибутом name.

Сутність Train перетворено в таблицю Train з первинним ключем ID та атрибутом seats amount.

Сутність Ticket перетворено в таблицю Ticket з первинним ключем ID та атрибутами price, pas full name, та зовнішнім ключем transit\_id.

Оскільки в логічній моделі безпосередній зв’язок N:M є неможливим, а в концептуальній моделі він існує між сутностями Train i Station, то для його реалізації було створено таблицю Transit, з первинним ключем tab\_id, атрибутом Date та зовнішніми ключами station\_id, train\_number.

Графічне подання логічної моделі «Сутність-зв’язок» зображено на рисунку 2.

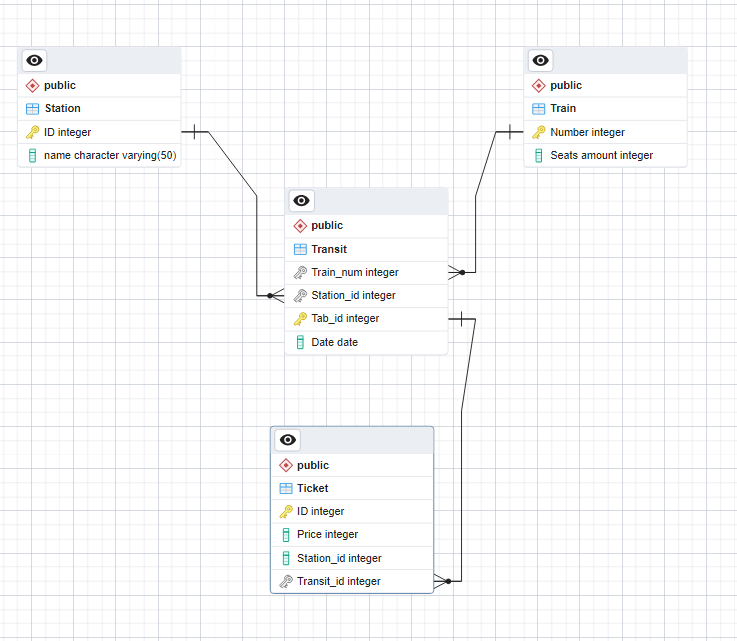


Рисунок 2 – Схема бази даних

Таблиця 1 ілюструє детальний перехід від однієї моделі до іншої.

Таблиця 1 – Опис об’єктів бази даних

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| *Station* – містить дані про станцію. | *id* – унікальний ідентифікатор станції | *integer* (число) |
| *name* – назва станції | *character varying* (рядок) |
| *Train* – містить дані про потяг за напрямком. | *number* – унікальний ідентифікатор потягу | *integer* (число) |
| *Seats amount* – кількість студентів у групі | *integer* (число) |
| *Ticket* | *id* – унікальний ідентифікатор квитка | *integer* (число) |
| *price* – ціна квитка | *integer* (число) |
| *pas\_full\_name* – ПІБ пасажира. | *character varying* (рядок) |
| *transit\_id –* ідентифікатор запису між потягом і станцією | *integer* (число) |
| *Transit* – містить інформацію про відповідність потяга та станції, через яку він їде. | *tab\_id* – унікальний ідентифікатор відповідності | *integer* (число) |
| *station\_id* – ідентифікатор станції | *integer* (число) |
| *train\_number* – ідентифікатор потягу | *integer* (число) |
| *date* – дата проїзду | *timestamp* (дата та час) |

**Функціональні залежності для кожної таблиці**

1. Station:
   1. id → {name}
2. Train:
   1. number → {Seats amount}
3. Ticket:
   1. id → {price, pas\_full\_name, transit\_id}
4. Transit:
   1. tab\_id → {station\_id, train\_number, date}

Ці функціональні залежності вказують на те, які атрибути в кожній таблиці визначаються від інших атрибутів. Це важливо для нормалізації та управління базою даних.

Транзитивні функціональні залежності виникають, коли один атрибут функціонально визначає інший через інший атрибут. Іншими словами, якщо A визначає B, а B визначає C, то ми можемо сказати, що A транзитивно визначає C. Дані таблиці не мають транзитивних функціональних залежностей.

**Відповідність схеми нормальним формам**

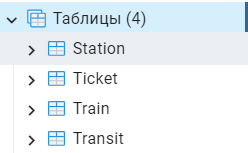
Вона відповідає 1НФ, оскільки всі атрибути є атомарними (недільними) та мають просту структуру, не містять складних об'єктів або масивів.

Вона відповідає 2НФ, оскільки всі неключові атрибути залежать від цілісних ключів цих таблиць. Наприклад, атрибут "name" в таблиці "Station" залежить від ідентифікатора "id", і атрибути в таблиці "Ticket" залежать від композитного ключа {id, station\_id, train\_number}.

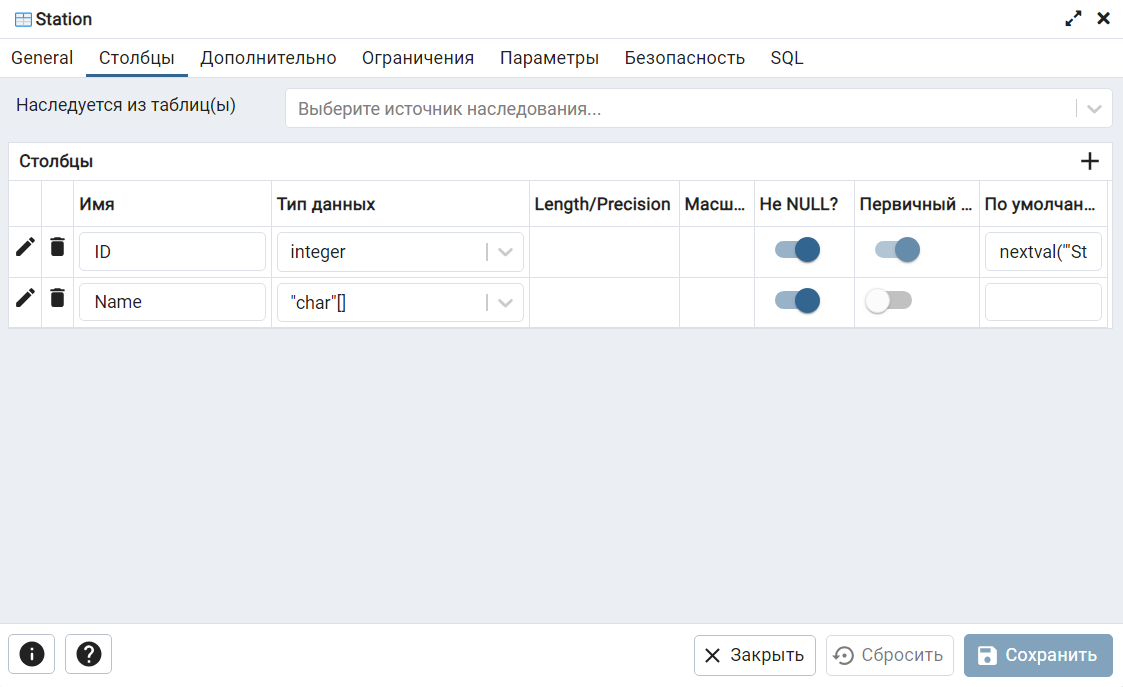
Вона також відповідає 3НФ, оскільки усі неключові атрибути мають бути функціонально залежними від первинного ключа, тобто не може бути взаємозалежностей між неключовими атрибутами. Кожен атрибут безпосередньо пов'язаний з ключами, від яких він залежить, що і виконується у нас.

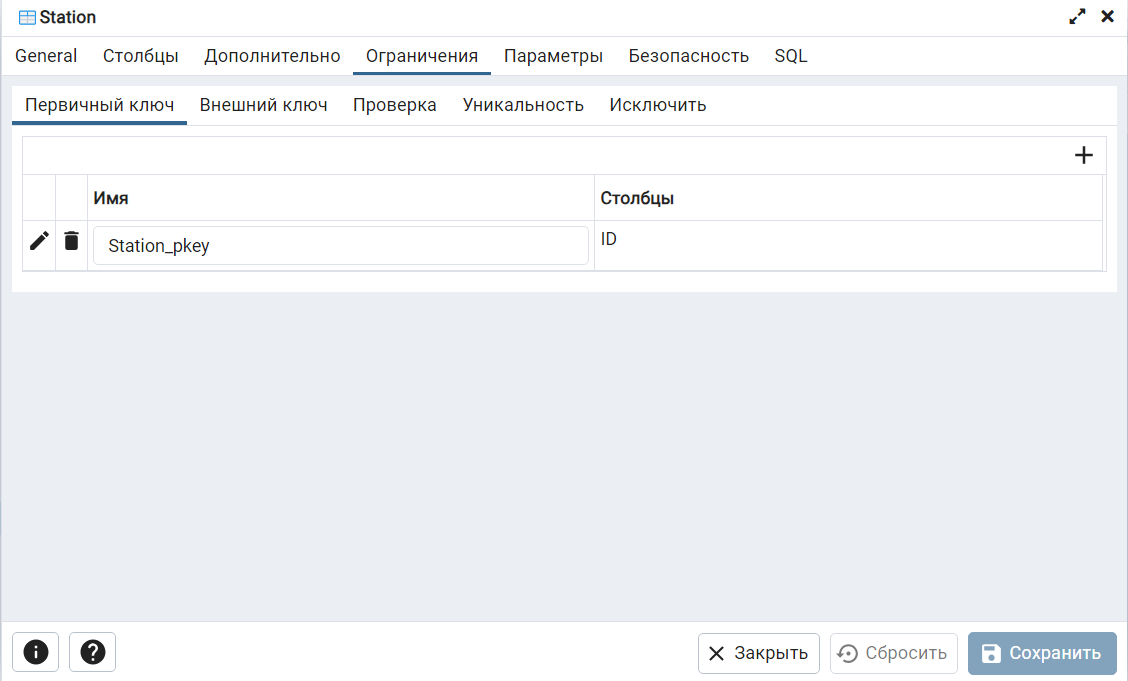
Загальний висновок полягає в тому, що ця база даних добре структурована і нормалізована, що робить її ефективною та надійною для зберігання і обробки даних.

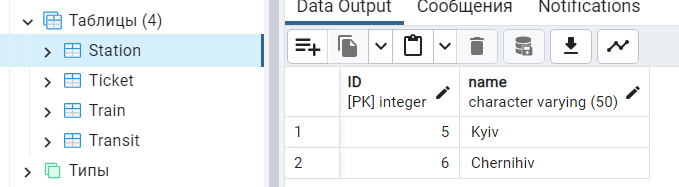
**Таблиці бази даних у pgAdmin4**

****

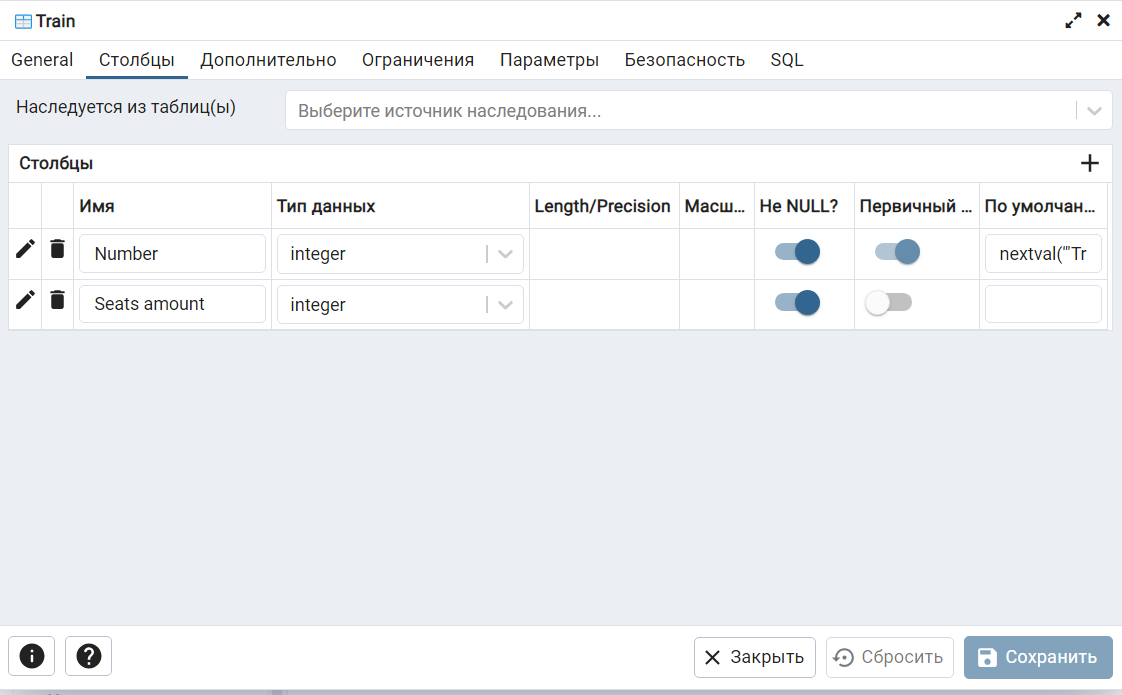
Station

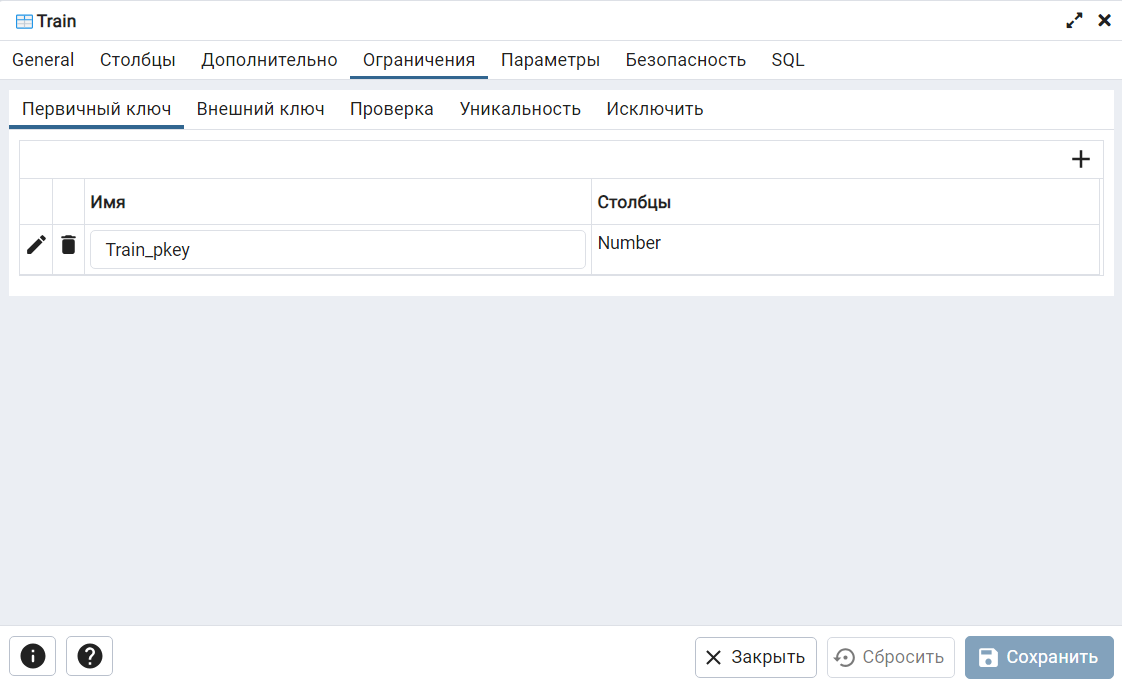
****

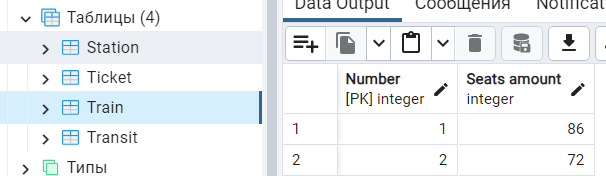
****



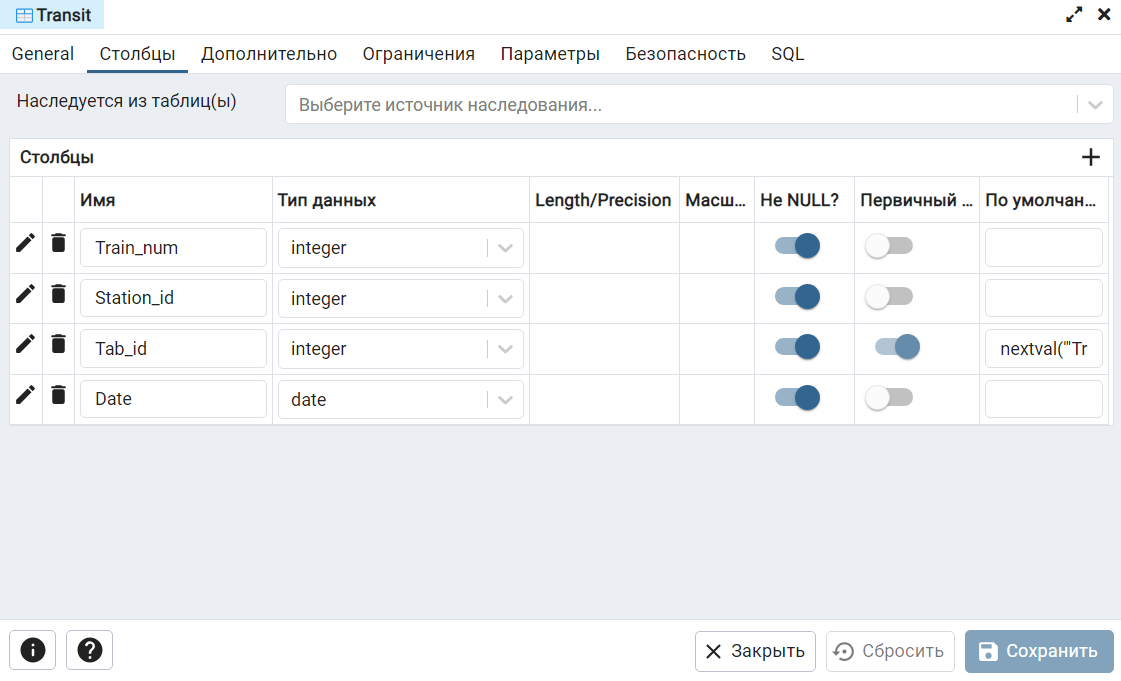
Train

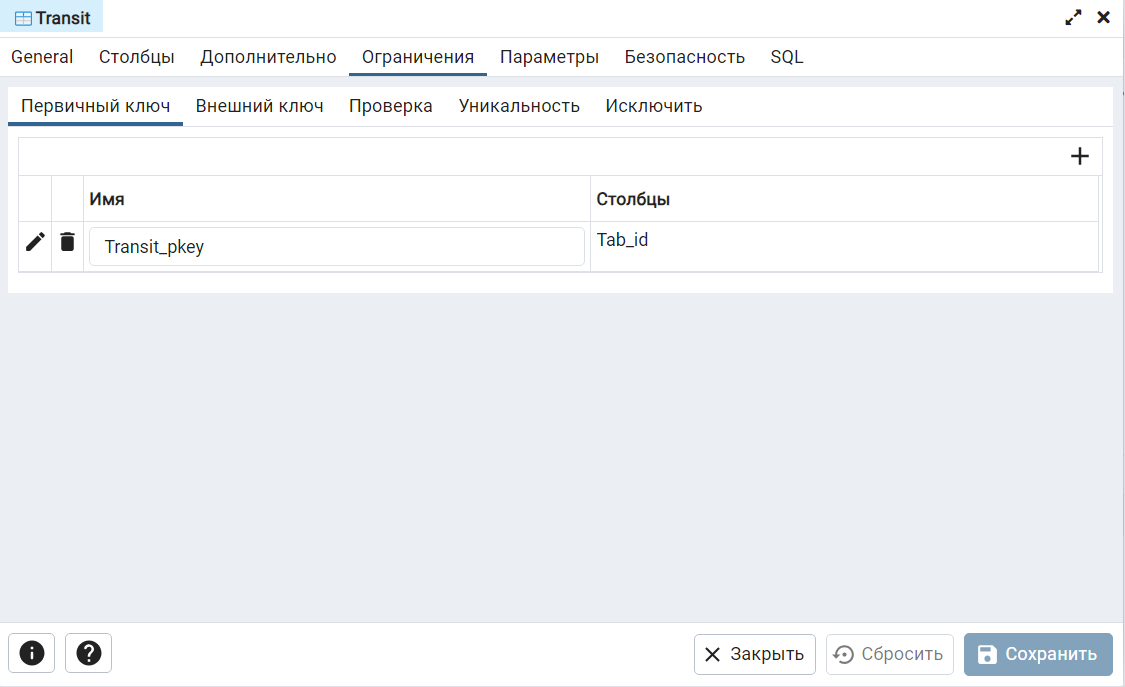
****

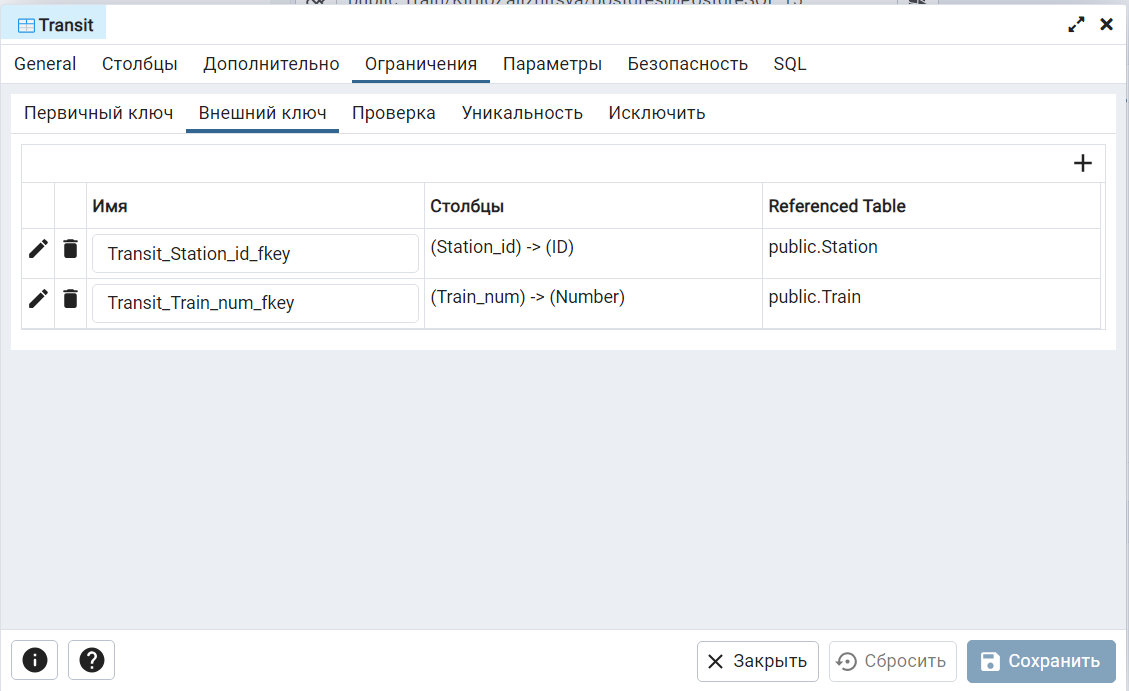
****

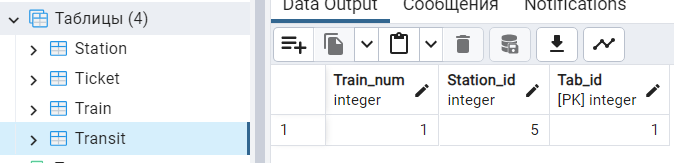
****

Transit

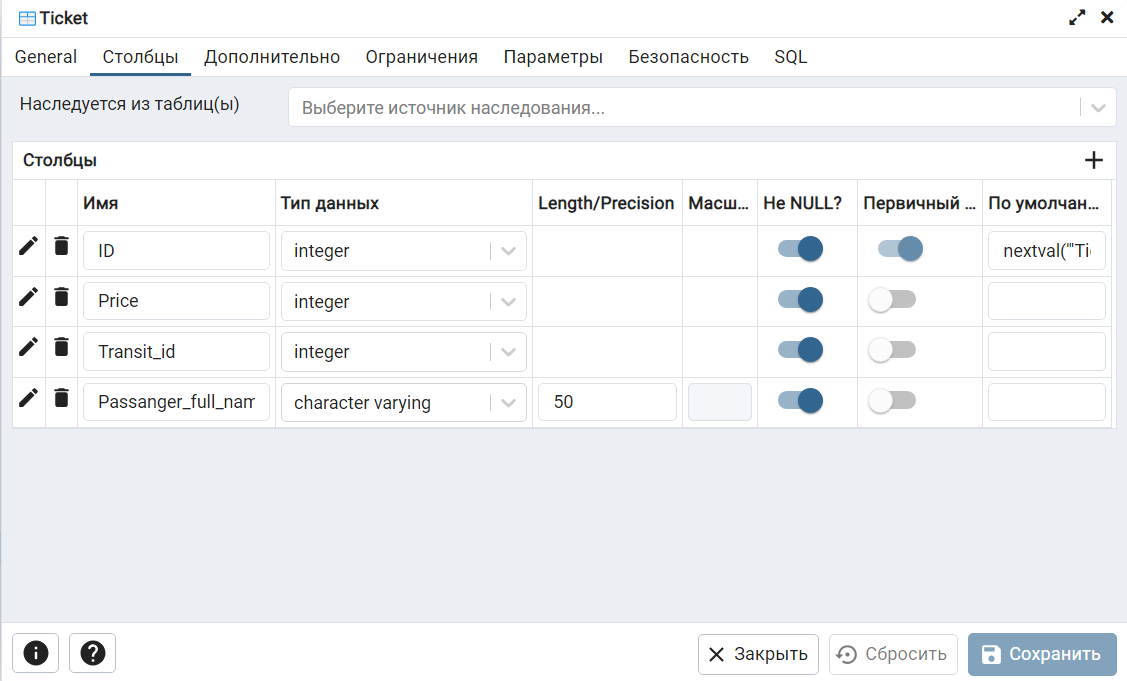
****

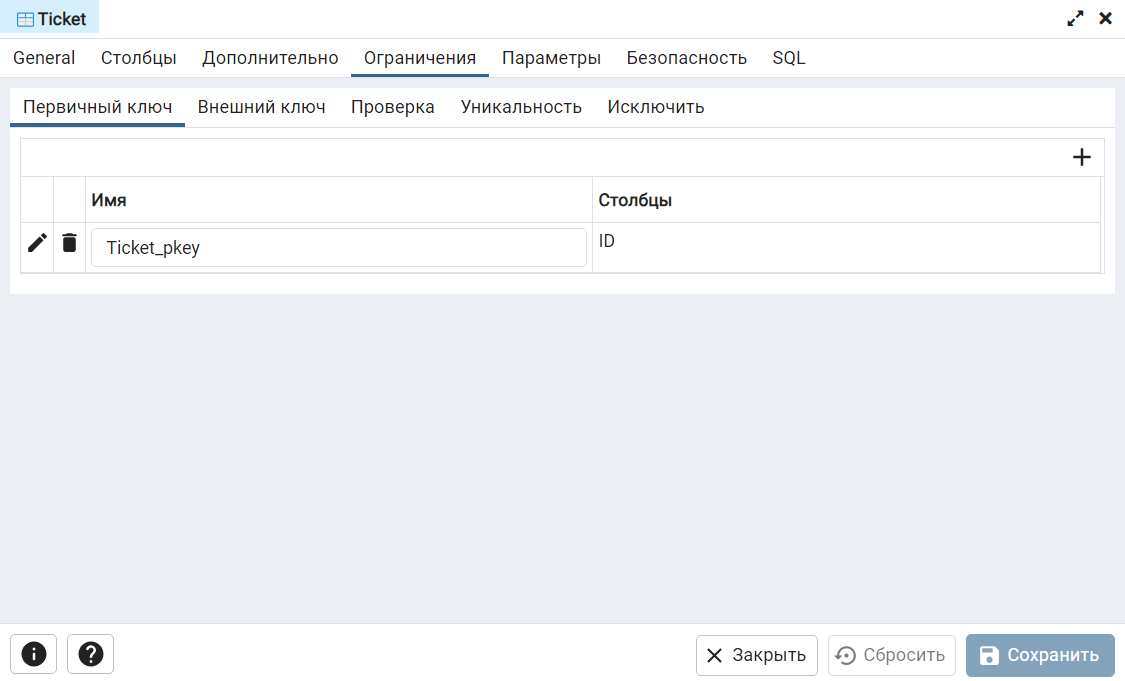
****

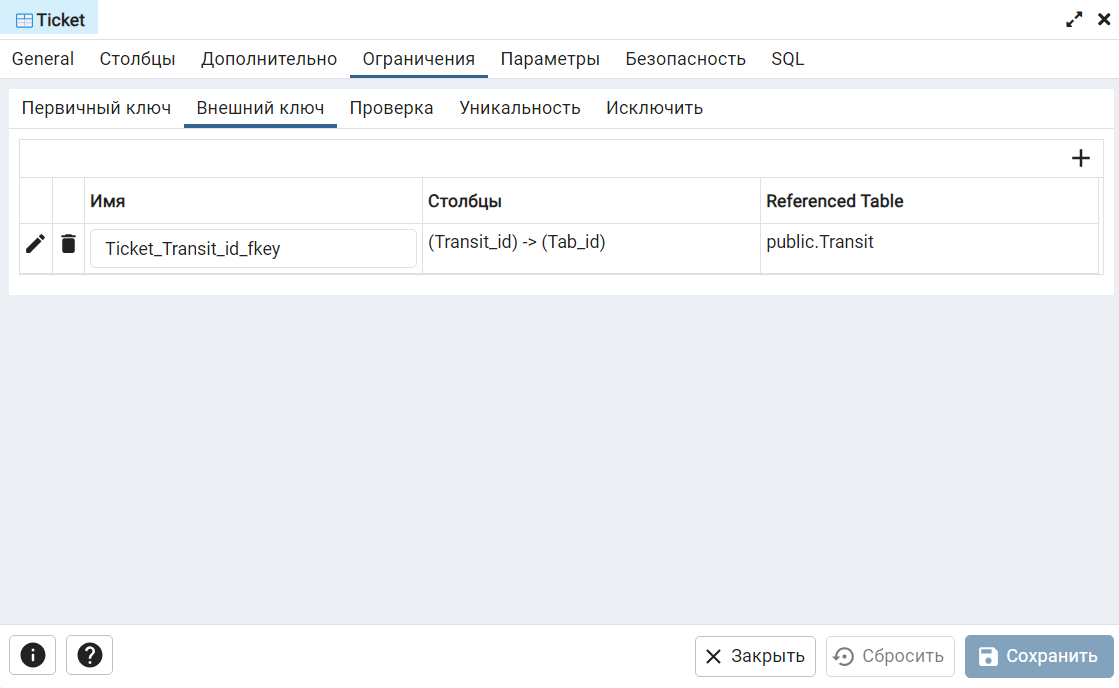
****

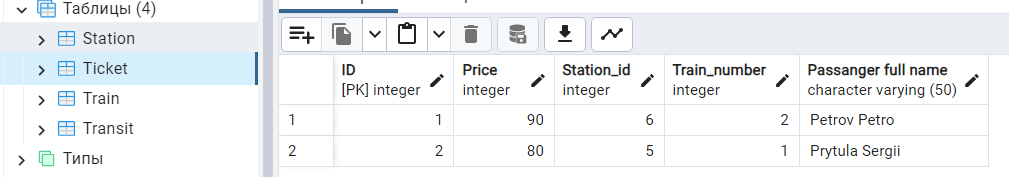
****

Ticket

****

****

****

****

**Таблиці в коді SQL**

**Висновок**

Під час виконання лабораторної роботи була створена база даних для системи продажу квитків на громадський транспорт. Ця база даних містить інформацію про поїзди, квитки, станції та їх взаємозв'язки.

Діаграма сутностей і взаємозв'язків (ER-діаграма) була використана для графічного представлення структури бази даних. Ми використовували нотацію Чена для позначення зв'язків і атрибутів.

Схема бази даних була перевірена на відповідність нормальним формам НФ1, НФ2 і НФ3. Усі таблиці відповідають цим нормальним формам, що гарантує надійне та ефективне зберігання даних.

Також була представлена оновлена версія схеми бази даних, де була включена таблиця "Transit" для відображення взаємозв'язків між поїздами та станціями, через які вони рухаються.

У кінці були надані знімки екрана з pgAdmin4, що демонструють властивості стовпців та обмеження, а також вміст таблиць бази даних PostgreSQL.

Загалом, розроблена база даних відповідає всім вимогам та завданням і готова до подальшого використання.