

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих**

**комп’ютерних систем**

**ЛАБОРАТОРНА РОБОТА №1**

**«Проектування бази даних та ознайомлення з базовими операціями СУБД PostgreSQL»**

з дисципліни

**«Бази даних та засоби управління»**

Виконав:

студент III курсу

групи КВ-12

Чернишков О.О.

Перевірив:

Павловський В.І.

*Метою роботи* є здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

*Завдання* роботи полягає у наступному:

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.

*Вимоги* до ER-моделі:

1. Сутності моделі предметної галузі мають містити зв’язки типу 1:N або N:M.
2. Кількість сутностей у моделі – 3-4. Кількість атрибутів у кожній сутності: від двох до п’яти.
3. Передбачити наявність зв’язку з атрибутом.
4. Для побудови ER-діаграм використовувати одну із нотацій: Чена, “Пташиної лапки (Crow’s foot)”, UML.

*Посилання на репозиторій GitHub:* <https://github.com/SashaBlack01/DBLabs>

*Контакт студента в Телеграм:* @Sasha\_Black01

**Модель «сутність-зв’язок» галузі Система відстеження посилок кур’єрської служби**

Обрана предметна галузь передбачає зберігання та обробку інформації про наявність, місцезнаходження, та інших даних про посилки в системі відстеження посилок кур’єрської служби.

**Сутності**

Згідно з обраною галуззю для побудови бази даних було виділено наступні сутності:

* Посилка (Parcel) з атрибутами: номер посилки, статус, дата створення, вартість, ідентифікатор кур’єра, ідентифікатор отримувача, ідентифікатор складу. Призначення: збереження даних щодо посилок.
* Отримувач (Recipient) з атрибутами: ідентифікатор, ім’я, адреса. Призначення: збереження даних отримувачів посилок.
* Відправник (Sender) з атрибутами: ідентифікатор, ім’я, адреса. Призначення: збереження даних відправників посилок.
* Кур’єр (Courier) з атрибутами: службовий номер, ім’я, номер телефону.
* Склад (Warehouse) з атрибутами: номер складу, адреса, номер телефону.

**Опис зв’язків**

Посилка має найбільшу кількість зв’язків з іншими сутностями:

* Кожна посилка має зв’язок з кур’єром який доставляє посилку, причому тільки з одним, але кожен кур’єр може доставляти декілька посилок, тому зв’язок 1:N.
* Кожна посилка має зв’язок з отримувачем, до якого вона прямує, при чому тільки з одним, але до отримувача може прямувати багато посилок, тому зв’язок 1:N.
* Кожна посилка має зв’язок з відправником, який створив та надіслав посилку, при чому відправник може бути тільки один для кожної посилки, тому зв’язок 1:N.
* Кожна посилка має зв’язок зі складом, на якому вона тимчасово зберігається, при чому склад для конкретної посилки може бути тільки один, але на одному складі може бути декілька посилок, тому зв’язок 1:N.
* Кожен кур’єр має зв’язок зі складом, до якого від закріплений, щоб доставляти посилки з конкретного складу, при чому кожен кур’єр може бути закріпленим за декількома складами, а за кожним складом може бути закріплено декілька кур’єрів, тому зв’язок M:N.

Графічне подання моделі «сутність зв’язок» побудованої за нотацією Чена представлена на рисунку 1.

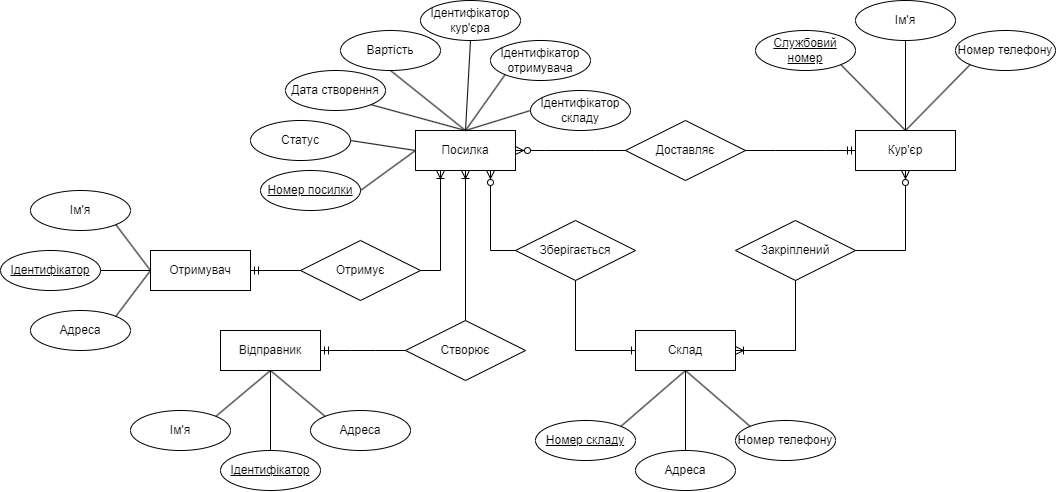


Рисунок 1 – ER-діаграма, побудована за нотацією Чена

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

Для подальшої зручності, назви сутностей та атрибутів було перекладено англійською мовою.

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **recipient** - містить інформацію про отримувача посилки | **recipient\_id** – унікальний ідентифікатор отримувача | Integer |
| **name** – ім’я отримувача | character varying (string) |
| **address** – адреса отримувача | character varying (string) |
| **sender** – містить інформацію про відправника посилки | **sender\_id** – унікальний ідентифікатор відправника | Integer |
| **name** – ім’я відправника | character varying (string) |
| **address** – адреса відправника | character varying (string) |
| **parcel** – містить інформацію про посилку | **parcel\_id** – унікальний ідентифікатор посилки | Integer |
| **status** – статус посилки | character varying (string) |
| **creation\_date** – дата створення посилки | Date |
| **parcel\_cost** – вартість посилки | Money |
| **courier\_id** – ідентифікатор кур’єра | Integer |
| **recipient\_id** – ідентифікатор отримувача | Integer |
| **warehouse\_id** – ідентифікатор складу | Integer |
| **courier** – містить інформацію про кур’єра | **courier\_id** – унікальний службовий ідентифікатор кур’єра | Integer |
| **name** – ім’я кур’єра | character varying (string) |
| **phone** – номер телефону кур’єра | character varying (string) |
| **warehouse** – містить інформацію про склад | **warehouse\_id** – унікальний ідентифікатор складу | Integer |
| **address** – адреса складу | character varying (string) |
| **phone** – номер телефону складу | character varying (string) |
| **courier\_warehouse** – містить інформацію про кур’єрів та склади, за якими вони закріплені | **courier\_warehouse\_id** – унікальний ідентифікатор рядка | Integer |
| **courier\_id** – службовий ідентифікатор кур’єра | Integer |
| **warehouse\_id** – ідентифікатор складу | Integer |

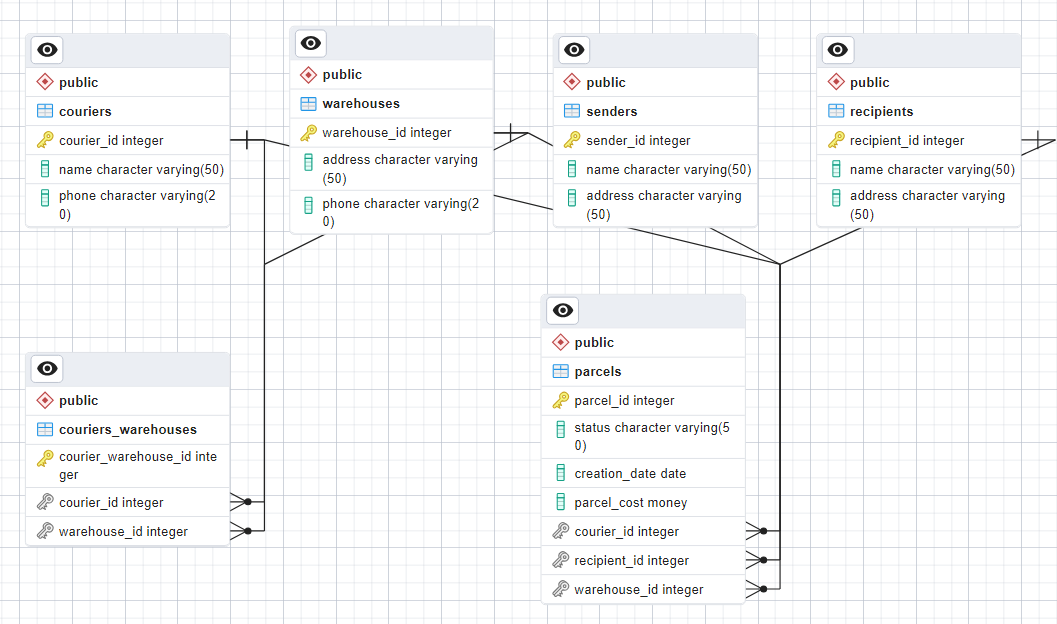
****

Рисунок 2 – Схема бази даних у графічному вигляді

Сутність "Parcel" було перетворено у таблицю "parcels"(сутність "Recipient" -> табл. "recipients"; сутність "Sender" -> табл. "senders"; сутність "Courier" -> табл. "couriers"; сутність "Warehouse" -> табл. "warehouses). Як бачимо, було реалізовано усі, описані вище, зв’язки між таблицями у базі даних.

Зв’язок «багато - до багатьох» між таблицями "couriers" і "warehouses" зумовив появу додаткової таблиці " couriers\_warehouses". "courier\_warehouse\_id" – це унікальний ідентифікатор рядка, "courier\_id" - це зовнішній ключ, що посилається на атрибут "courier\_id" в таблиці "couriers", а " warehouse\_id" - зовнішній ключ, що посилається на атрибут " warehouse\_id" в таблиці " warehouses". Ця таблиця дозволяє встановлювати зв'язок між конкретними кур’єрами (у "couriers") і їх складами (у "warehouses").

**Відповідність схеми бази даних нормальним формам (НФ1, НФ2 і НФ3)**

Таблиця "parcels" (parcel\_id, status, creation\_date, parcel\_cost, courier\_id, recipient\_id, warehouse\_id)

parcel\_id → status, creation\_date, parcel\_cost, courier\_id, recipient\_id, warehouse\_id;

parcel\_id → status;

parcel\_id → creation\_date;

parcel\_id → parcel\_cost;

parcel\_id → courier\_id;

parcel\_id → recipient\_id;

parcel\_id → warehouse\_id;

status ↮ creation\_date;

creation\_date ↮ parcel\_cost;

parcel\_cost ↮ courier\_id;

courier\_id ↮ recipient\_id;

recipient\_id ↮ warehouse\_id;

warehouse\_id ↮ status;

Таблиця "recipients" (recipient\_id, name, address)

recipient\_id → name, address;

recipient\_id → name;

recipient\_id → address;

name ↮ address;

Таблиця "senders" (sender\_id, name, address)

sender\_id → name, address;

sender\_id → name;

sender\_id → address;

name ↮ address;

Таблиця "couriers" (courier\_id, name, phone)

courier\_id → name, phone;

courier\_id → name;

courier\_id → phone;

name ↮ phone;

Таблиця "warehouse" (warehouse\_id, address, phone)

warehouse\_id → address, phone;

warehouse\_id → address;

warehouse\_id → phone;

address ↮ phone;

Таблиця "couriers\_warehouses" (courier\_warehouse\_id, courier\_id, warehouse\_id)

courier\_warehouse\_id → courier\_id, warehouse\_id;

courier\_warehouse\_id → courier\_id;

courier\_warehouse\_id → warehouse\_id;

courier\_id ↮ warehouse\_id;

warehouse\_id ↮ courier\_id

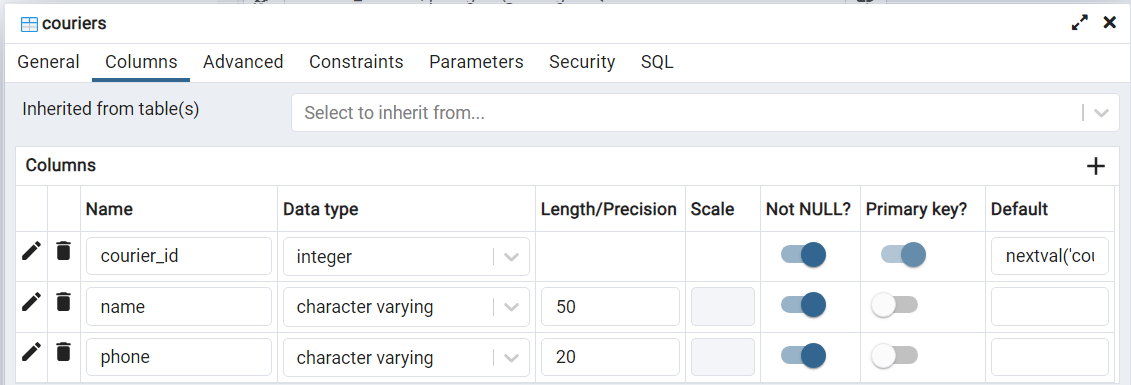
Таблиці бази даних відповідають НФ1 тому, що всі рядки унікальні, всі атрибути не мають нереляційні типи (масиви, об’єкти тощо).

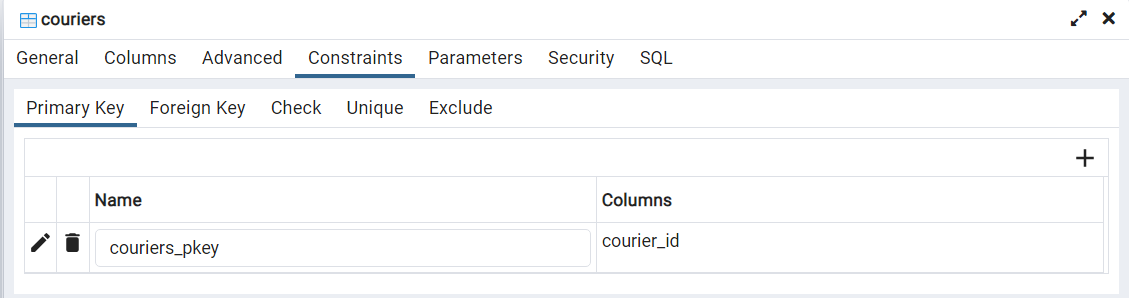
Таблиці бази даних відповідає НФ2 тому, що всі таблиці мають первинні ключі, і всі неключові атрибути повністю залежать від ключа і не мають підмножин.

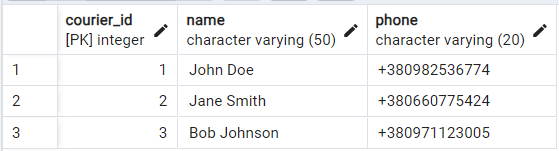
Схема бази даних відповідає НФ3 тому, що всі не ключові атрибути залежні тільки від первинного ключа і нетранзитивно залежні від нього.

**Копії екрану з pgAdmin4**

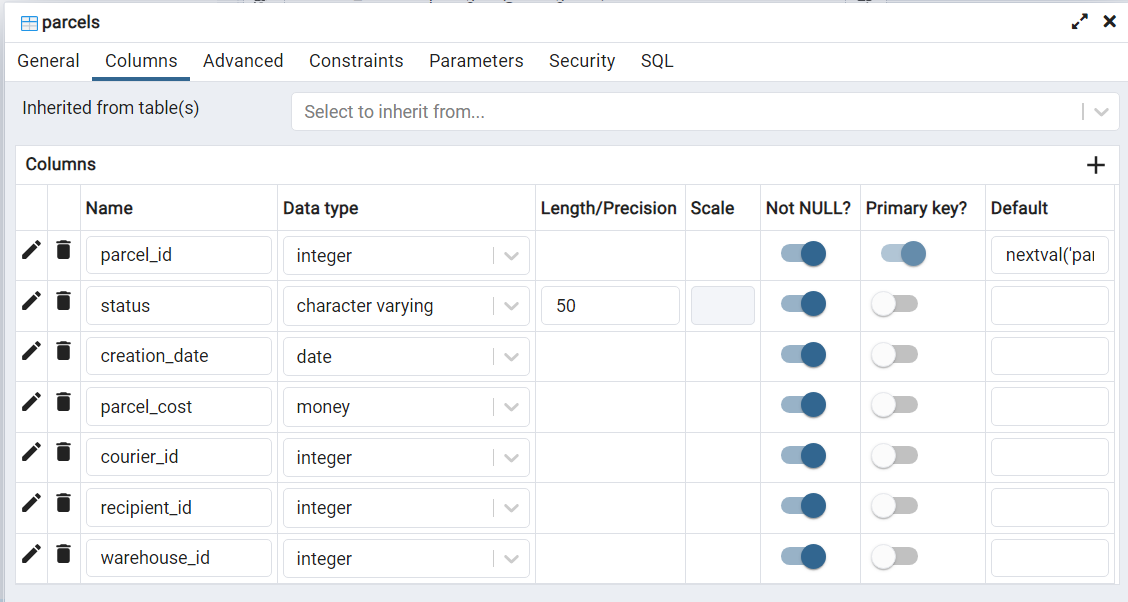
***Таблиця "couriers":***

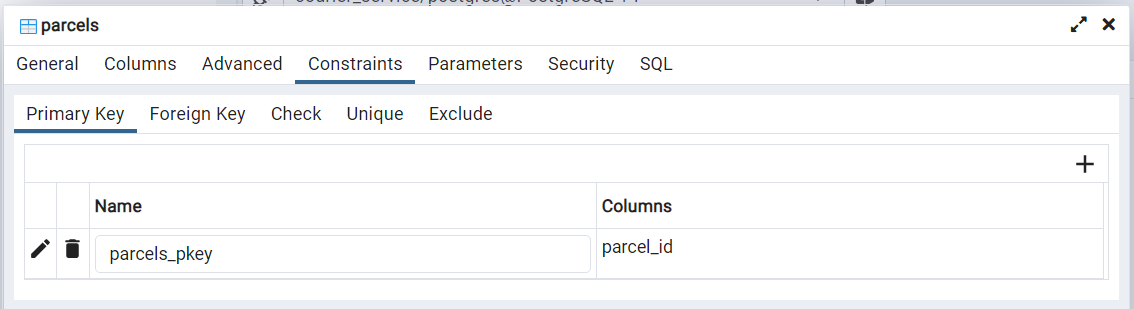


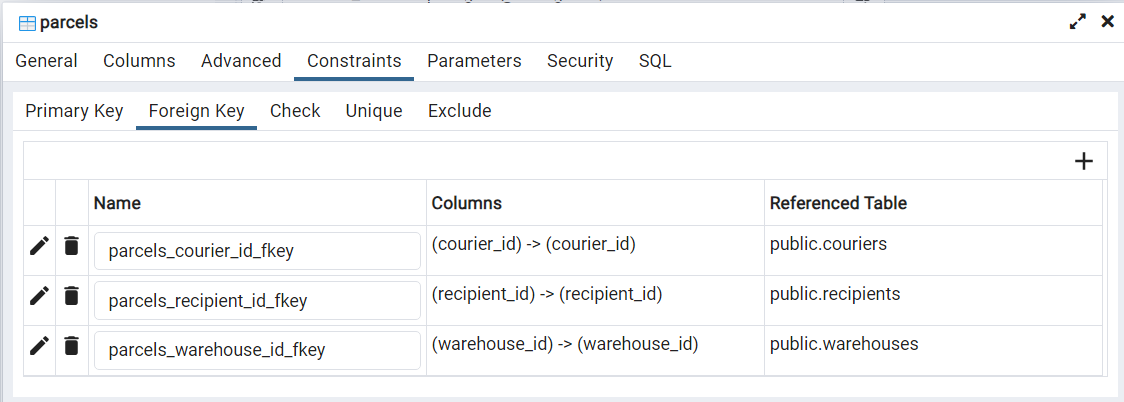


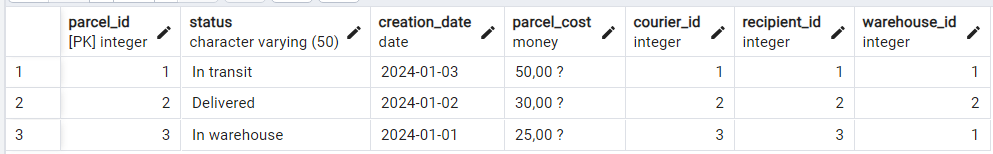


***Таблиця "parcels":***

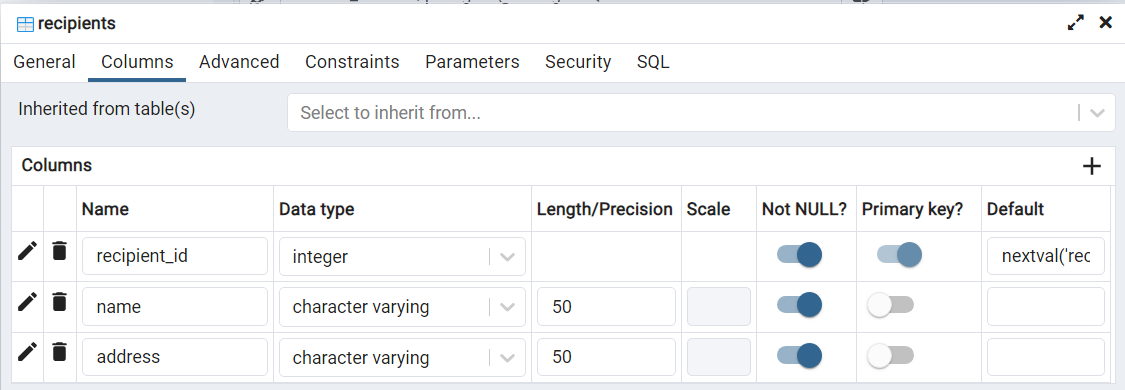


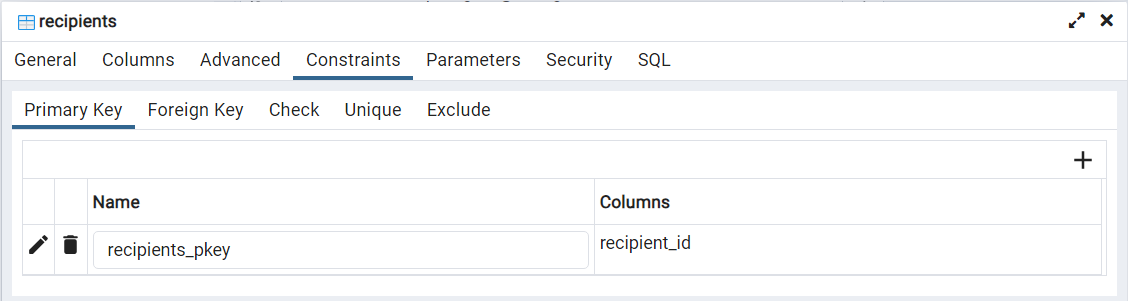


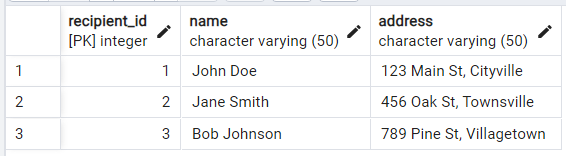




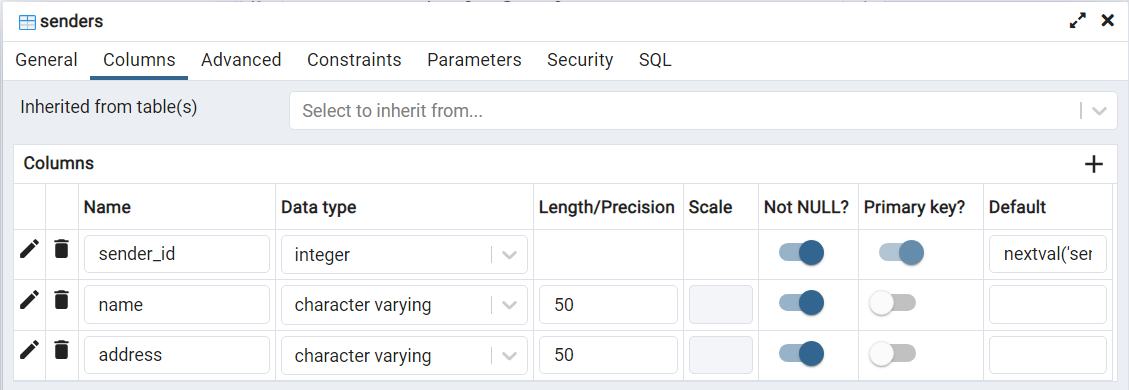
***Таблиця "recipients":***

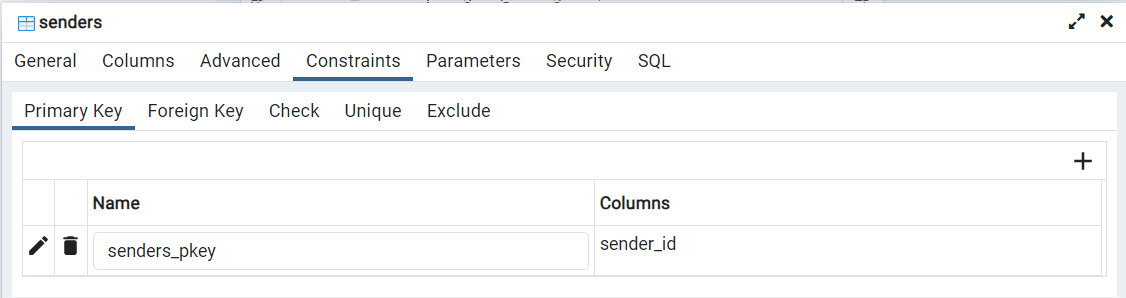


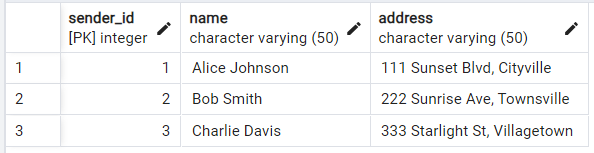




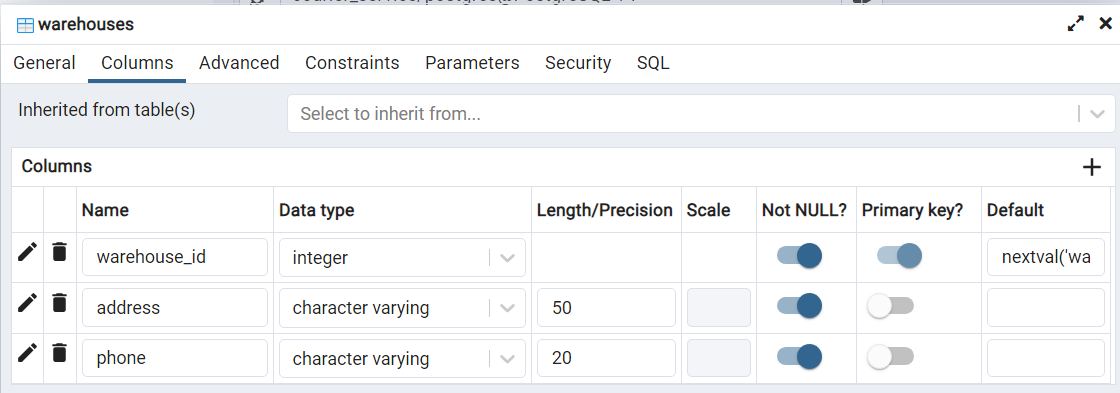
***Таблиця "senders":***

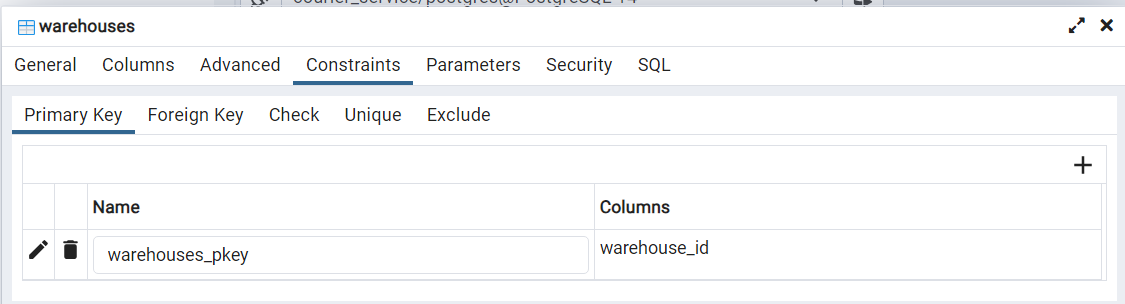


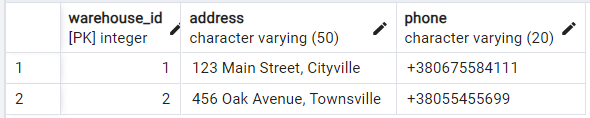




***Таблиця "warehouses":***







***Таблиця "couriers\_warehouses":***

