

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. 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”*

Виконала: студентка ІІІ курсу

ФПМ групи КВ-12

Ус В. О.

Перевірив:

Павловський В. І.

Київ 2023

*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.

**Модель «сутність-зв’язок» для платформи замовлення та доставки продуктів харчування**

Обрана предметна галузь передбачає моделювання платформи для замовлення та доставки продуктів харчування.

**Сутності**

Згідно цієї області для побудови бази даних було виділено наступні сутності:

*Користувач:* Представляє користувачів платформи. Має атрибути, такі як ID користувача, прізвище та ім’я і номер телефону.

*Заклад:* Представляє магазини або заклади, які пропонують продукти харчування для замовлення. Має атрибути, такі як ID закладу, назва закладу та номер телефону для замовлення.

*Продукт:* Описує продукти, доступні для замовлення в закладах. Має атрибути, такі як ID продукту, ID закладу, назва та ціна.

*Замовлення:* Представляє інформацію про замовлення користувачів. Має такі атрибути, як ID замовлення, ID користувача, ID продуктів та сума для оплати.

*Доставка:* Відображає інформацію про саму доставку. Має атрибути, такі як ID доставки, ID замовлення, час доставки, адреса доставки.

**Опис зв’язків**

*Зв'язок між "Користувач" і "Замовлення" є 1:N*, що означає, що один користувач може мати багато замовлень, але кожне замовлення належить лише одному користувачу. Тобто один користувач може робити багато замовлень, і кожне замовлення належить конкретному користувачеві.

*Зв'язок між "Заклад" і "Продукт" є 1:N*, що означає, що кожен заклад може пропонувати багато продуктів, але кожен продукт належить лише одному закладу. Цей зв'язок визначає, які продукти доступні в кожному закладі.

*Зв'язок між "Замовлення" і "Продукт" є M:N*, що означає, що кожне замовлення може містити багато різних продуктів, і в той же час, кожен продукт може бути частиною багатьох різних замовлень. Тобто кілька замовлень можуть включати один і той самий продукт, і одне замовлення може містити різні продукти.

*Зв'язок між "Замовлення" і "Доставка" N:1.* Це означає, що кожне замовлення має лише одну доставку, а доставка може мати багато замовлень з різних закладів харчування.

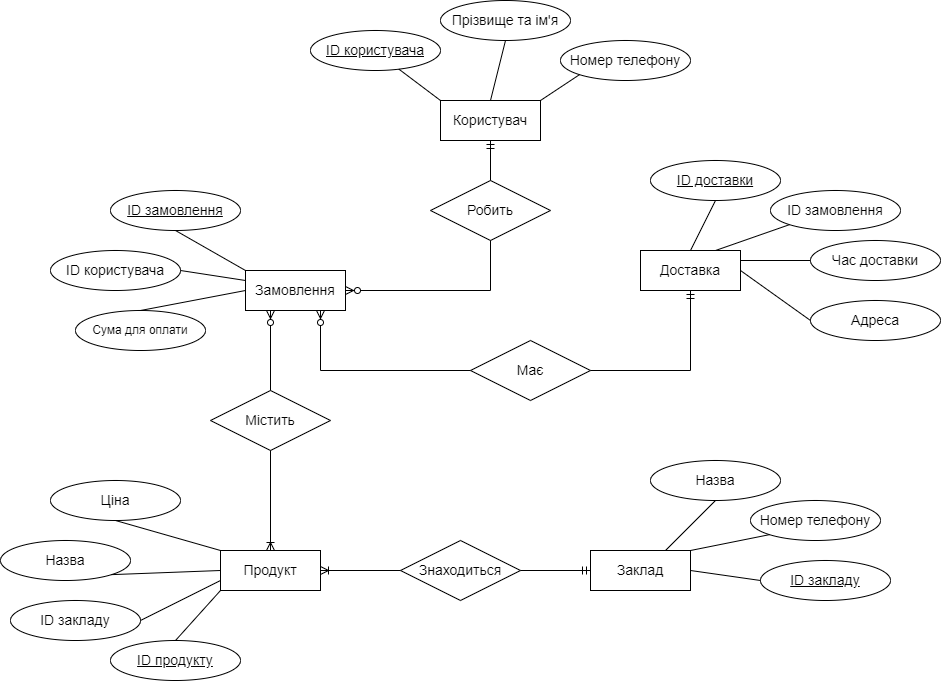


Рисунок 1 - ER-діаграма, побудована за нотацією Чена

**Опис процесу перетворення**

Сутність Користувач було перетворено в таблицю «User», яка має первинний ключ id\_user та атрибути name\_user, phone\_user.

Сутність Замовлення було перетворено в таблицю «Order», яка має первинний ключ id\_order, зовнішній ключ id\_user та атрибут sum.

Сутність Продукт було перетворено в таблицю «Product», яка має первинний ключ id\_product, зовнішній ключ id\_shop та атрибути name\_product, price.

В логічній моделі неможливий безпосередній зв'язок М:N, а в концептуальній моделі він існує між сутностями Замовлення і Продукт. Для його представлення було введено допоміжну таблицю Замовлення\_Продукт, для збереження даних про продукти в кожному замовленні (рис. 2).

Сутність Заклад було перетворено в таблицю «Shop», яка має первинний ключ id\_shop та атрибути name\_shop, phone\_shop.

Сутність Доставка було перетворено в таблицю «Delivery», яка має первинний ключ id\_delivery, зовнішній ключ id\_order та атрибути time, adress.

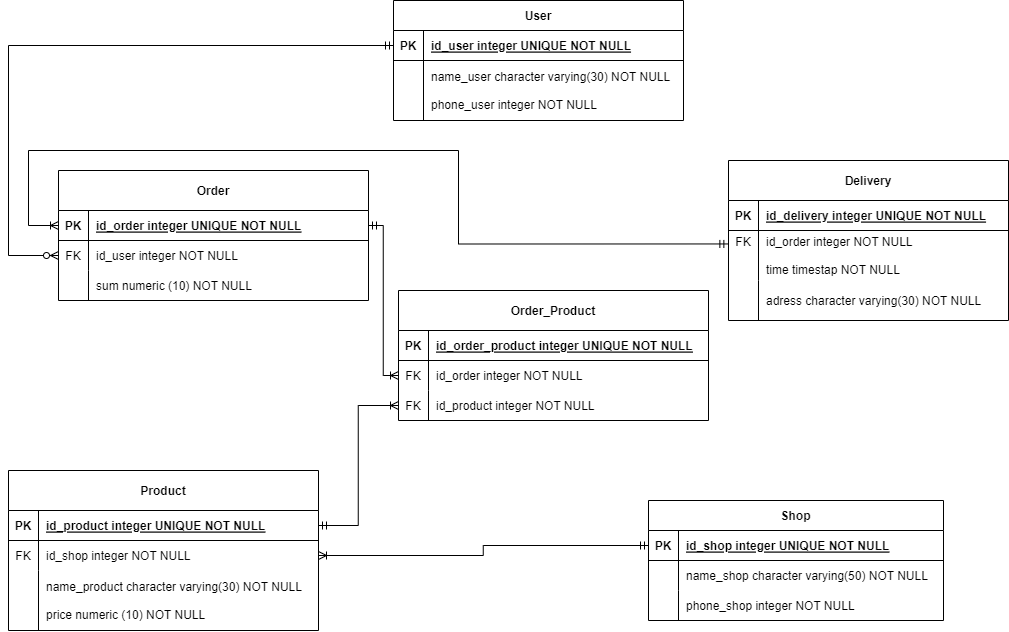


Рисунок 2 – Схема бази даних

**Функціональні залежності**

Для оцінки відповідності схеми бази даних нормальним формам НФ1, НФ2 та НФ3, спершу розглянемо функціональні залежності в даній схемі:

**Сутність "User":**

Первинний ключ: id\_user

Атрибути: name\_user, phone\_user

Нормальна форма НФ1: Для НФ1 всі атрибути повинні бути атомарними, тобто не розкладатися на більш дрібні складові. У цій сутності немає атрибутів, які розкладаються, тож вона відповідає НФ1.

Нормальна форма НФ2: Для НФ2 повинні виконуватися всі вимоги НФ1, і кожен неключовий атрибут повинен повністю залежати від первинного ключа. В даній сутності атрибути name\_user і phone\_user обидва повністю залежать від id\_user, тож вона також відповідає НФ2.

Нормальна форма НФ3: Для НФ3 повинні виконуватися всі вимоги НФ2, і кожен неключовий атрибут повинен бути прямо залежний від первинного ключа, а не через інші неключові атрибути. У даній сутності атрибути name\_user і phone\_user не залежать від інших атрибутів, окрім id\_user, тому сутність також відповідає НФ3.

**Сутність "Order":**

Первинний ключ: id\_order

Зовнішній ключ: id\_user

Атрибути: sum

Нормальна форма НФ1: Сутність відповідає НФ1, оскільки всі атрибути є атомарними.

Нормальна форма НФ2: Сутність відповідає НФ2, оскільки sum повністю залежить від id\_order.

Нормальна форма НФ3: Сутність відповідає НФ3, оскільки sum не залежить від жодного іншого атрибута окрім id\_order.

**Сутність "Product":**

Первинний ключ: id\_product

Зовнішній ключ: id\_shop

Атрибути: name\_product, price

Нормальна форма НФ1: Сутність відповідає НФ1, оскільки всі атрибути є атомарними.

Нормальна форма НФ2: Сутність відповідає НФ2, оскільки name\_product і price повністю залежать від id\_product.

Нормальна форма НФ3: Сутність відповідає НФ3, оскільки name\_product і price не залежать від жодного іншого атрибута окрім id\_product.

**Сутність "Shop":**

Первинний ключ: id\_shop

Атрибути: name\_shop, phone\_shop

Нормальна форма НФ1: Сутність відповідає НФ1, оскільки всі атрибути є атомарними.

Нормальна форма НФ2: Сутність відповідає НФ2, оскільки name\_shop і phone\_shop повністю залежать від id\_shop.

Нормальна форма НФ3: Сутність відповідає НФ3, оскільки name\_shop і phone\_shop не залежать від жодного іншого атрибута окрім id\_shop.

**Сутність "Delivery":**

Первинний ключ: id\_delivery

Зовнішній ключ: id\_order

Атрибути: time, address

Нормальна форма НФ1: Сутність відповідає НФ1, оскільки всі атрибути є атомарними.

Нормальна форма НФ2: Сутність відповідає НФ2, оскільки time і address повністю залежать від id\_delivery.

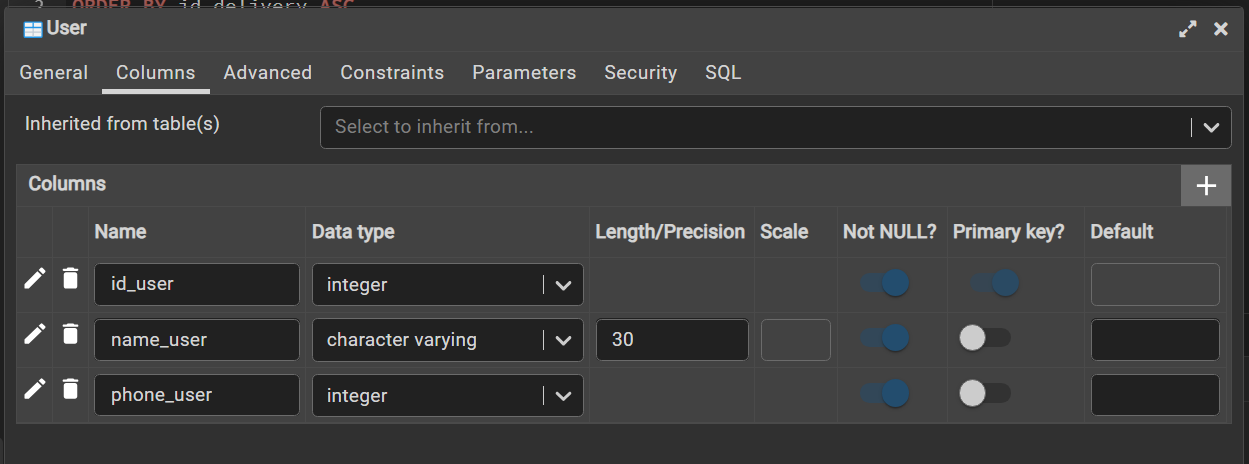
Нормальна форма НФ3: Сутність відповідає НФ3, оскільки time і address не залежать від жодного іншого атрибута окрім id\_delivery.

**Загальний висновок:** Всі сутності схеми бази даних відповідають НФ1, НФ2 та НФ3, оскільки всі атрибути є атомарними і виконують умови нормальних форм. Тобто, схема бази даних є добре структурованою з точки зору нормалізації даних.

**Таблиці бази даних у pgAdmin4**

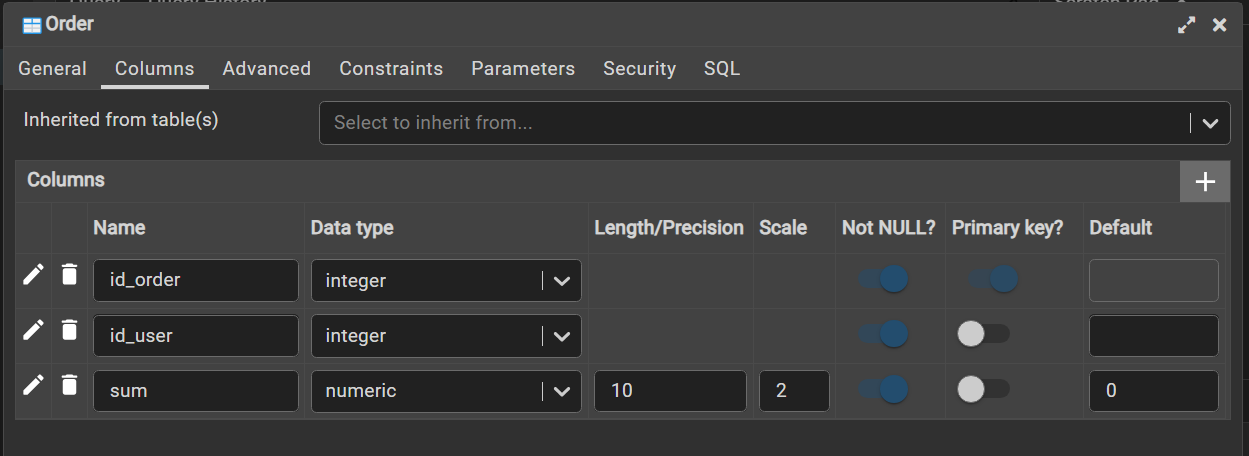
*Копії екрану з pgAdmin4, що відображають назви, типи та обмеження на стовпці:*

**User**

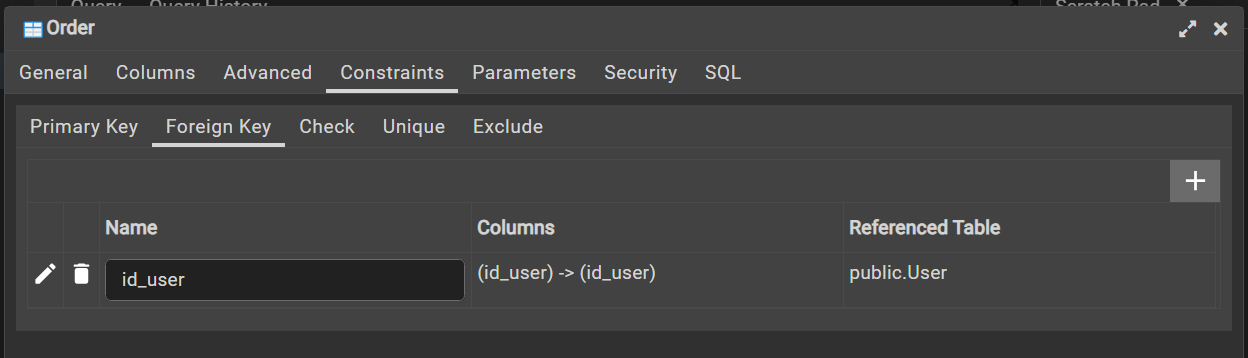


Немає зовнішніх ключів.

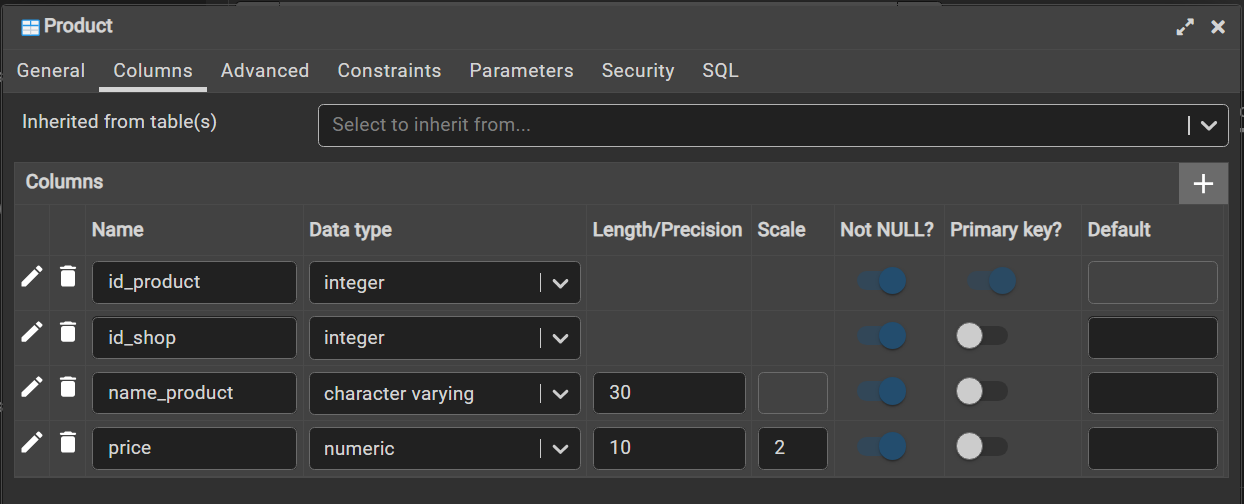
**Order**

****

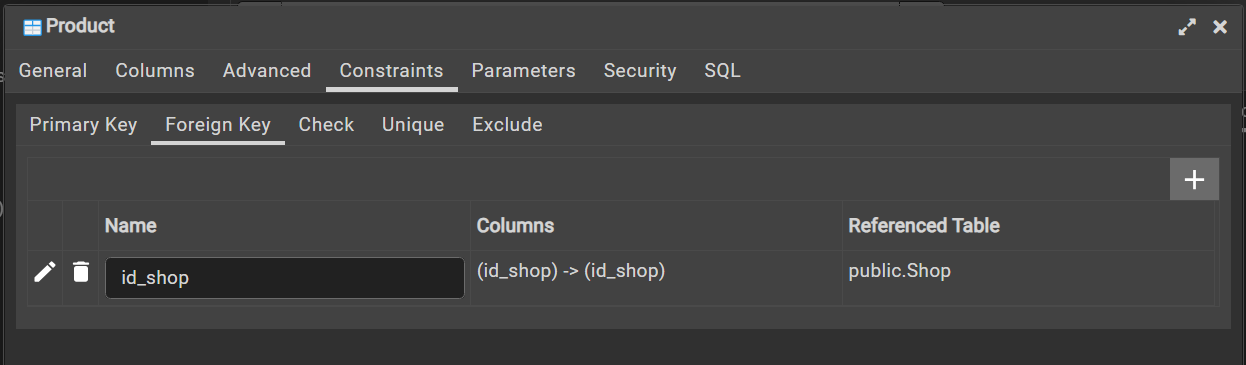
Має зовнішній ключ



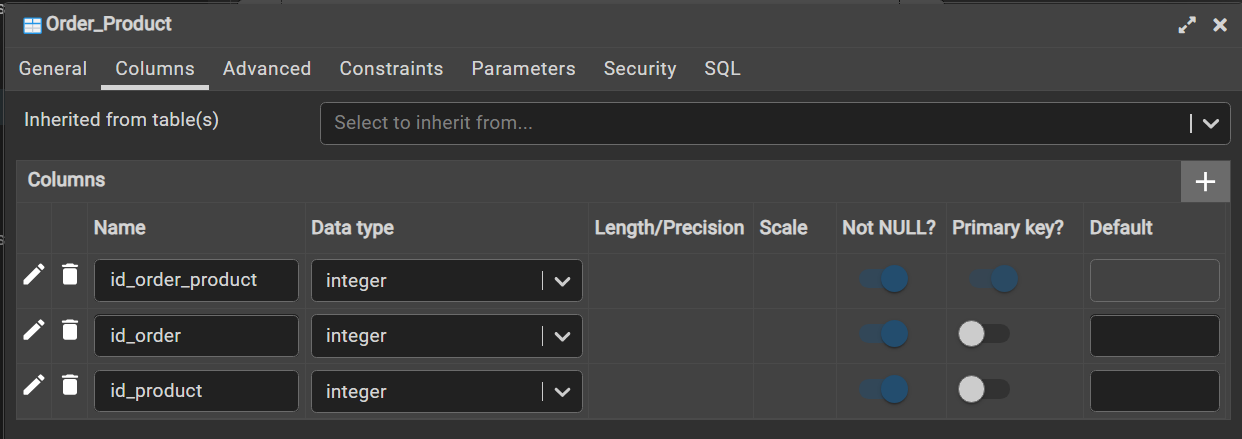
**Product**



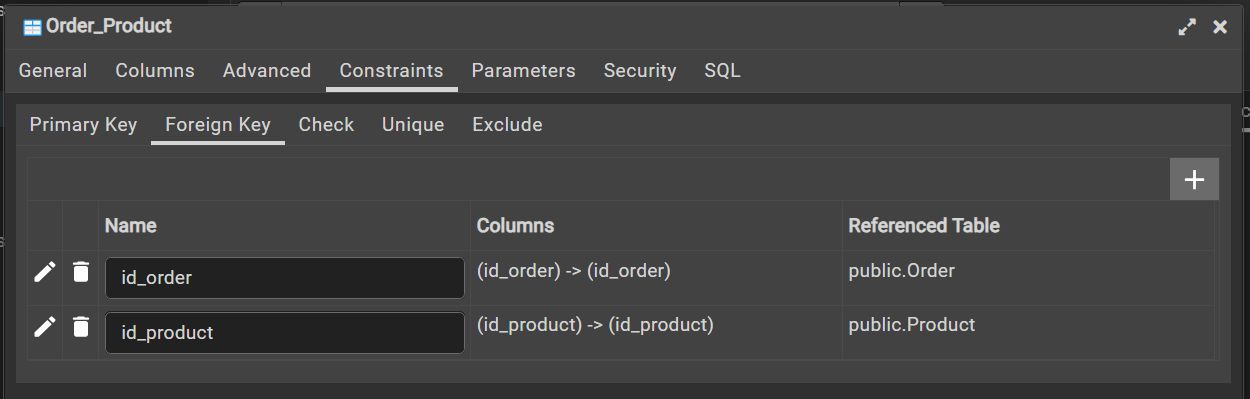
Має зовнішній ключ



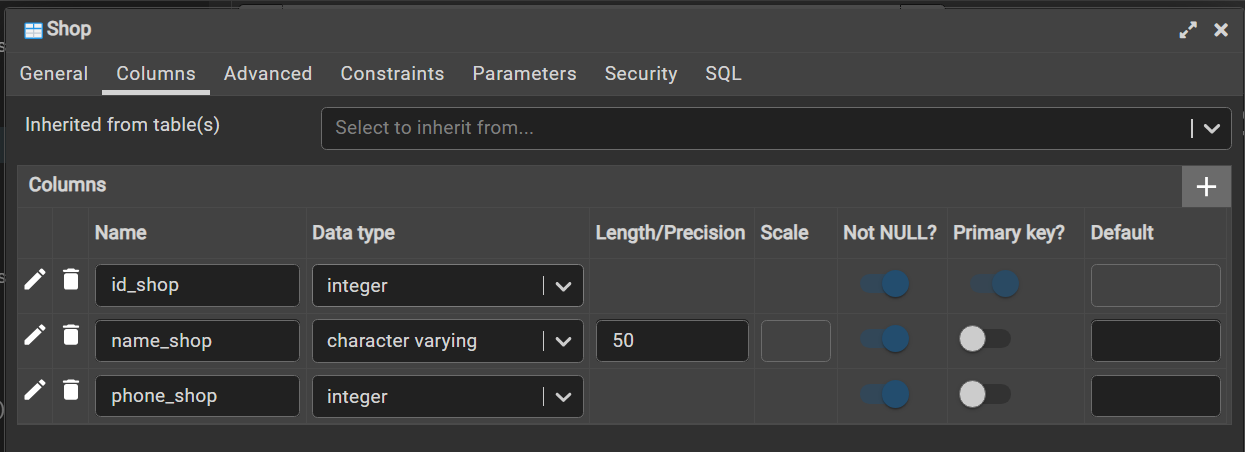
**Order\_Product**



Має зовнішні ключі

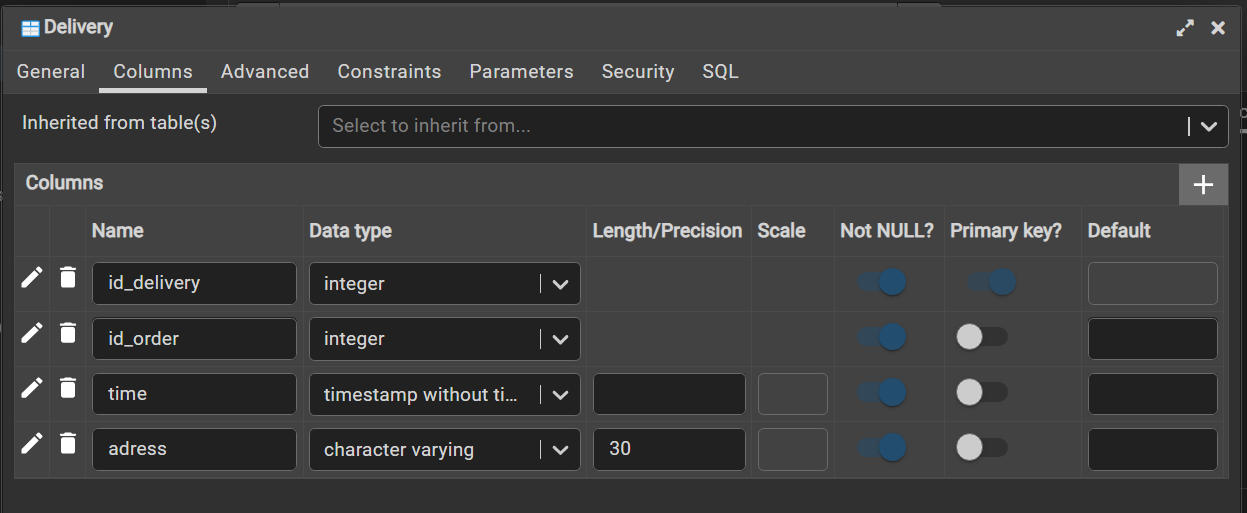


**Shop**

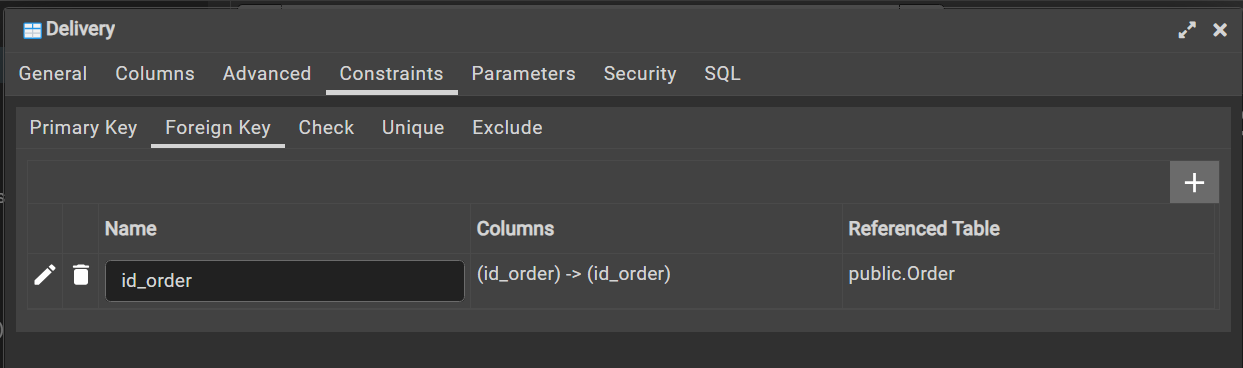


Немає зовнішніх ключів

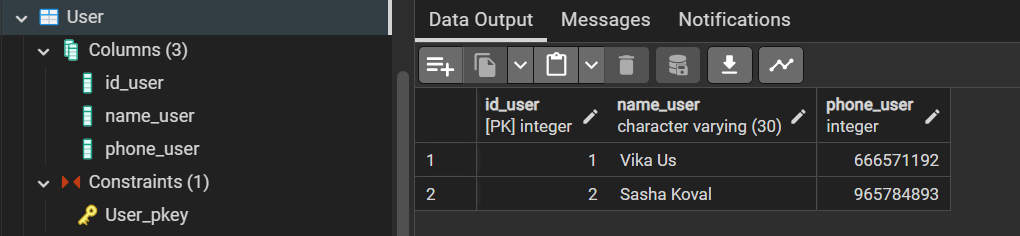
**Delivery**



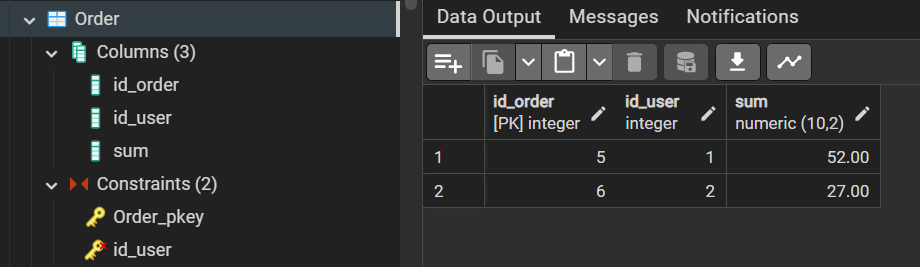
Має зовнішній ключ



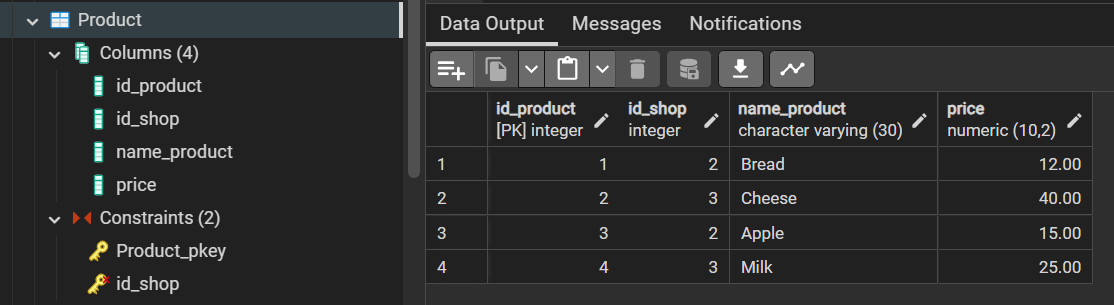
*Копії екрану з pgAdmin4, що відображають вміст таблиць бази даних у PostgreSQL*



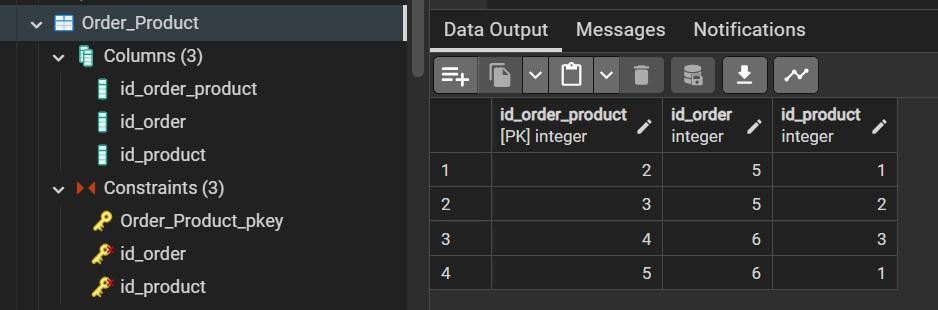
Таблиця «User»



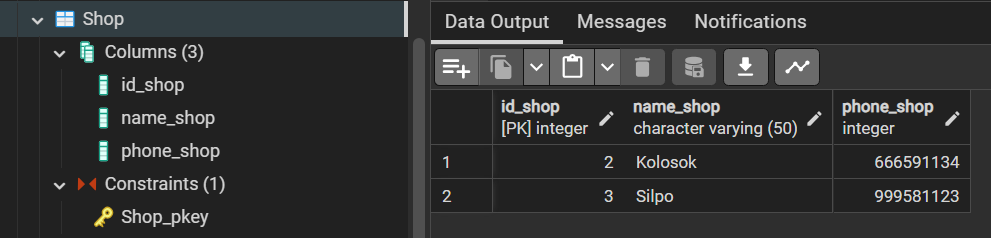
Таблиця «Order»



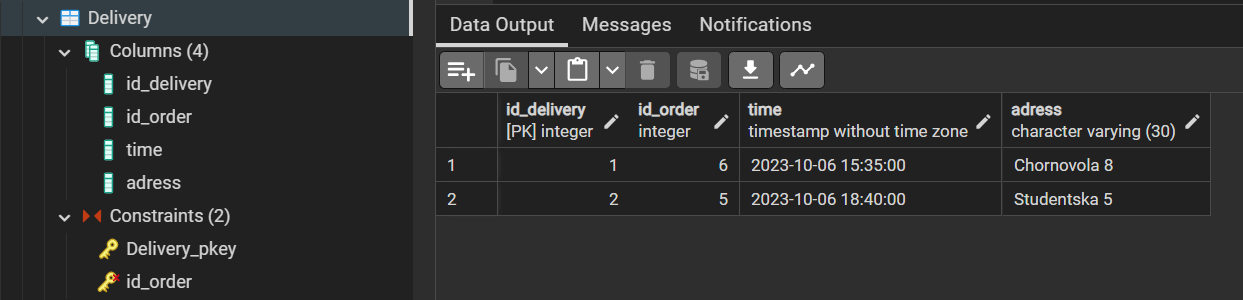
Таблиця «Product»



Таблиця «Order\_Product»



Таблиця «Shop»



Таблиця «Delivery»

**Контакти:**

GitHub: <https://github.com/vickaus/DataBase>

Telegram: @vicka\_us