

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 ІII курсу

групи **КВ-22 Деркач А. І**. Перевірив:

**Павловский В. І.**

Київ – 2024

*Метою роботи* є здобуття вмінь проектування бази даних 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 (3 НФ).

4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin4 та внести декілька рядків даних у кожну з таблиць засобами pgAdmin4.

**Модель «сутність-зв’язок»**

Обрана предметна галузь передбачає фінансову систему обліку транзакцій, зберігання даних про акаунти користувачів, рахунки користувачів, транзакції.

**Сутності**

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

Рахунок з атрибутами: ідентифікатор рахунку, баланс, тип валюти, ідентифікатор власнику рахунку. Призначення: збереження інформації про рахунки які беруть участь в транзакціях.

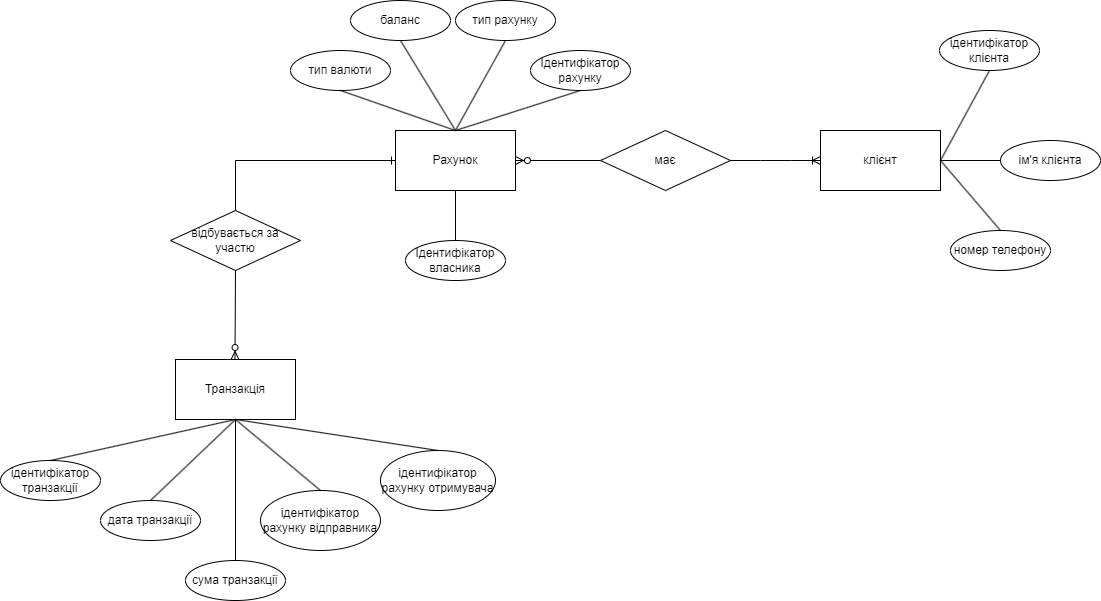
Клієнт з атрибутами ідентифікатор клієнта, ім’я клієнта, адреса, номер телефону. Призначення: зберігання інформації про клієнтів фінансової системи.

Транзакція з атрибутами: ідентифікатор транзакції, дата транзакції, сума транзакції, ідентифікатор рахунку-відправника, ідентифікатор рахунку-отримувача. Призначення: збереження інформації про транзакції які відбулися.

**Опис зв’язків**

Кожен рахунок має одного клієнта-власника. Інші клієнти також можуть користуватися цим рахунком, якщо їм було надано доступ. Таким чином, між сутностями Клієнт і Рахунок існує зв'язок M:N.

Один рахунок може здійснити багато транзакцій і одна транзакція може стосуватись тільки одного рахунку. Тому між сутностями Рахунок і Транзакція існує зв’язок 1:N.

Рисунок 1 - ER-діаграма, побудована за нотацією Чена

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

В логічній моделі неможливий безпосередній зв'язок М:N, а в концептуальній моделі він існує між сутностями Клієнт і Рахунок. Для його представлення було введено допоміжну таблицю Доступ до Рахунку. Ця таблиця буде зберігати інформацію про наявність у клієнта доступу до рахунку.

Назви таблиць і полів було перекладено англійською, для зручності наведено таблицю нижче.

Таблиця 1 – Опис структури бази даних «Система управління вакцинацією населення»

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **Сlient –** містить дані про клієнтів | **сlient\_id** – унікальний ідентифікатор того, хто має рахунок в системі  **name** – ім’я клієнта  **phone –** номер телефону клієнта | **integer** *(числовий)*  **character varying** *(рядок)*  **character varying** *(рядок)* |
| **Transaction –** зберігає дані про кожну транзакцію. | **transaction\_id –** унікальний ідентифікатор транзакції  **date –** дата транзакції  **sender\_id –** унікальний ідентифікатор рахунку, з якого знімаються кошти  **receiver\_id –** унікальний ідентифікатор рахунку, на який зараховуються кошти  **amount –** сума транзакції | **integer** *(числовий)*  **date**  **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)* |
| **Account –** містить інформацію про кожен рахунок | **account\_id –** унікальний ідентифікатор рахунку  **balanse –** поточний баланс рахунку  **currency\_type –** тип валюти  **account\_type –** тип рахунку  **owner\_id –** унікальний ідентифікатор власника рахунку | **integer** *(числовий)*  **integer** *(числовий)*  **character varying** *(рядок)*  **character varying** *(рядок)*  **integer** *(числовий)* |
| **Account\_access–** містить інформацію про відповідність клієнта і рахунку, до якого він має доступ. | **id\_table –** унікальний ідентифікатор таблиці вдіповідності  **client\_id –** унікальний ідентифікатор клієнта  **account\_id –** унікальний ідентифікатор рахунку | **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)* |

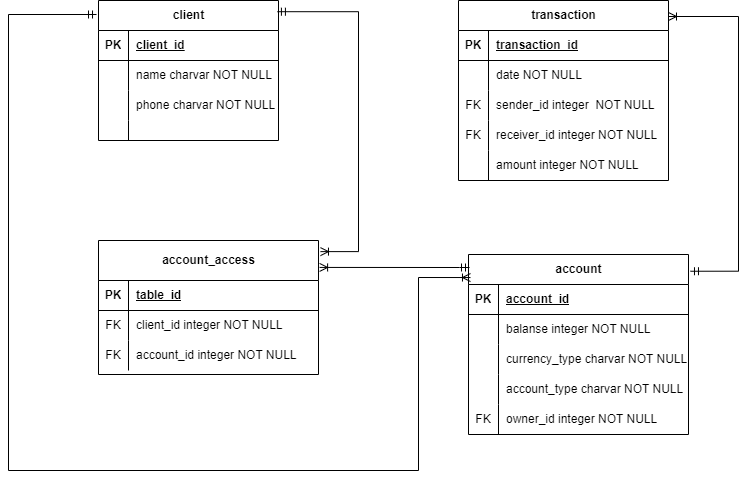


Рисунок 2 – Схема бази даних

**Функціональні залежності**

**Сlient:**

**client\_id –** унікальний ідентифікатор клієнта

**name –** ім’я клієнта

**phone –** номер телефону

**сlient\_id →** name, phone

client\_id **→**phone**→** name(транзитивне відношення, але включає в себе ключовий атрибут)

phone → citizen\_id, name

**Account:**

**account\_id –** унікальний ідентифікатор рахунку

**balanse –** баланс

**currency\_type** – тип валюти на рахунку

**account\_type** – тип рахунку

**owner\_id –** ідентифікатор власника акаунту FK

**account\_id** **→ balanse, currency\_type, account\_type, owner\_id**

**Transaction:**

**transaction\_id –** унікальний ідентифікатор транзакції

**date –** дата здійснення транзакції

**sender\_id –** унікальний ідентифікатор рахунку-відправника FK

**receiver\_id –** унікальний ідентифікатор рахунку-отримувача FK

**amount –** сума транзакції

**transaction\_id** **→ date, sender\_id, receiver\_id, amount**

**Account\_access:**

**table\_id –** унікальний ідентифікатор таблиці

client\_id – унікальний ідентифікатор клієнту FK

account\_id – унікальний ідентифікатор рахунку FK

table\_id → сlient\_id, account\_id

**Відповідність схеми нормальним формам:**

1НФ:

* Кожен атрибут в кожній таблиці є атомарним. Кожна клітинка містить єдине значення і кожен запис є унікальним.

2НФ:

* Схема перебуває в 1НФ
* Кожний неключовий атрибут функціонально залежить від цілого ключа.

У даному випадку кожна таблиця схеми має унікальний ідентифікатор від якого залежать всі її неключові атрибути.

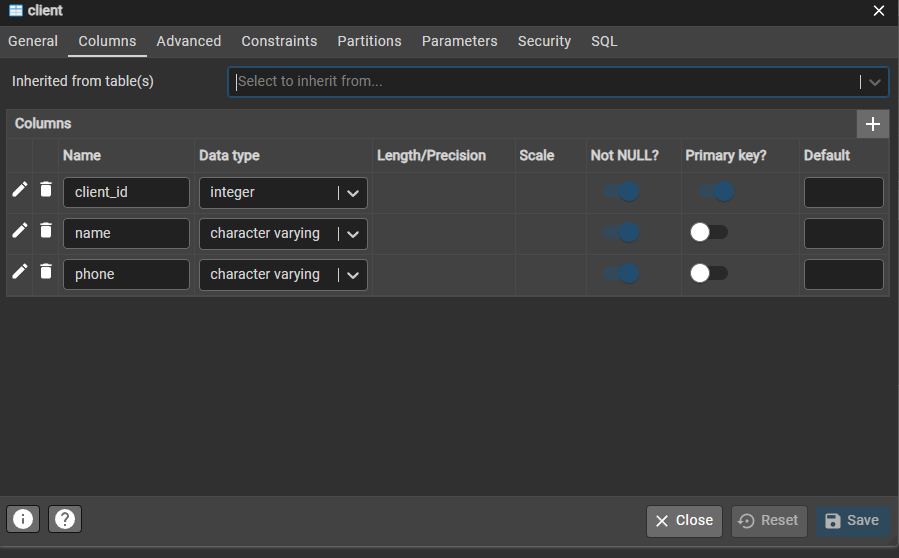
3НФ:

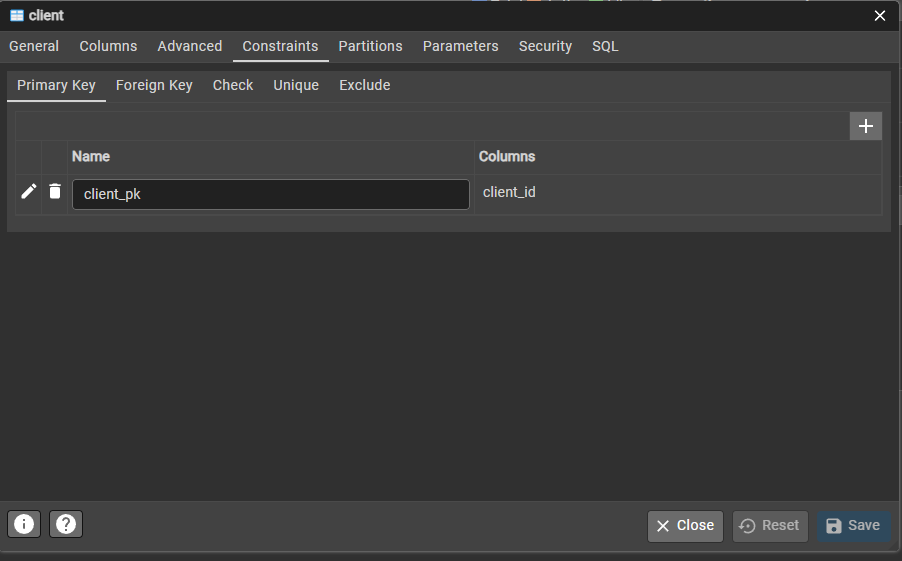
* Схема перебуває в 2НФ
* Кожен не первинний атрибут має бути не транзитивно залежним від кожного ключа.

Оскільки дана схема в 2НФ та неключові атрибути не транзитивно залежать від інших неключових атрибутів, схема також в 3НФ.

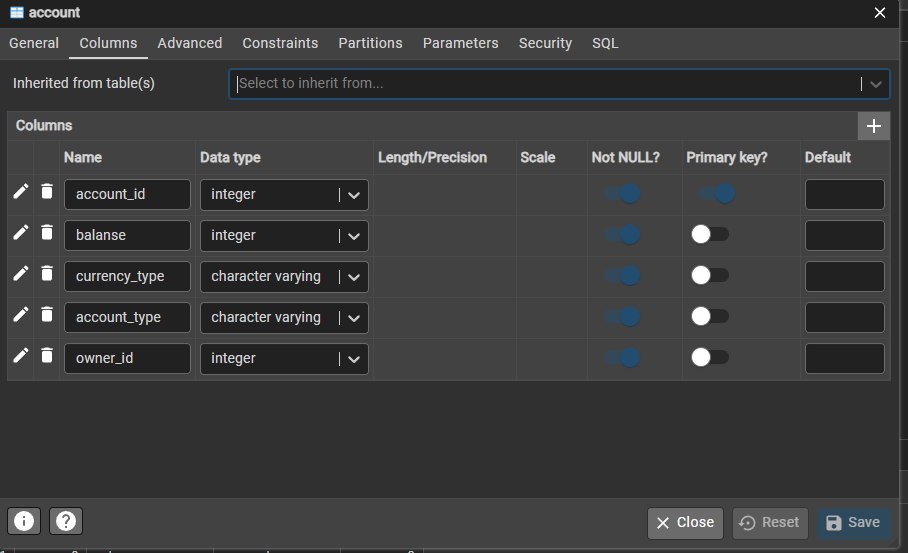
**Таблиці бази даних у pgAdmin4**

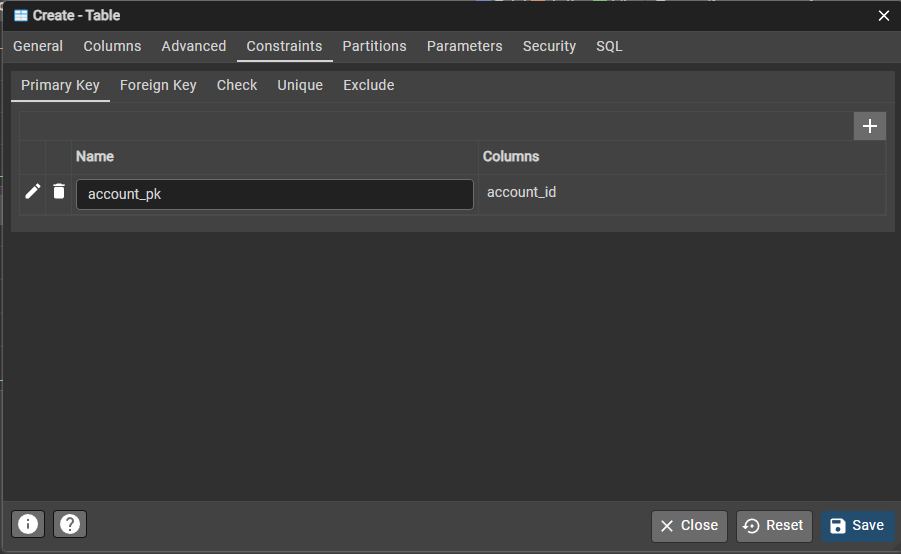
**Client**

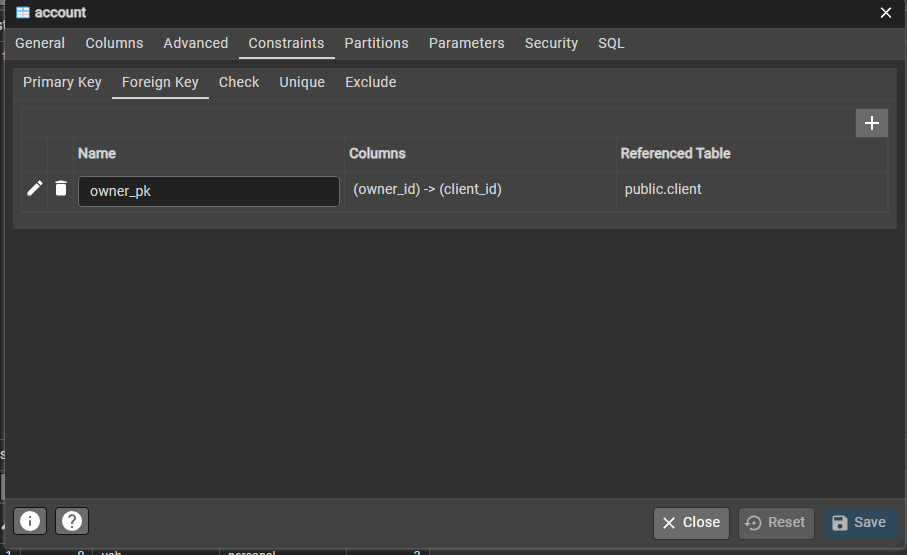
****

****

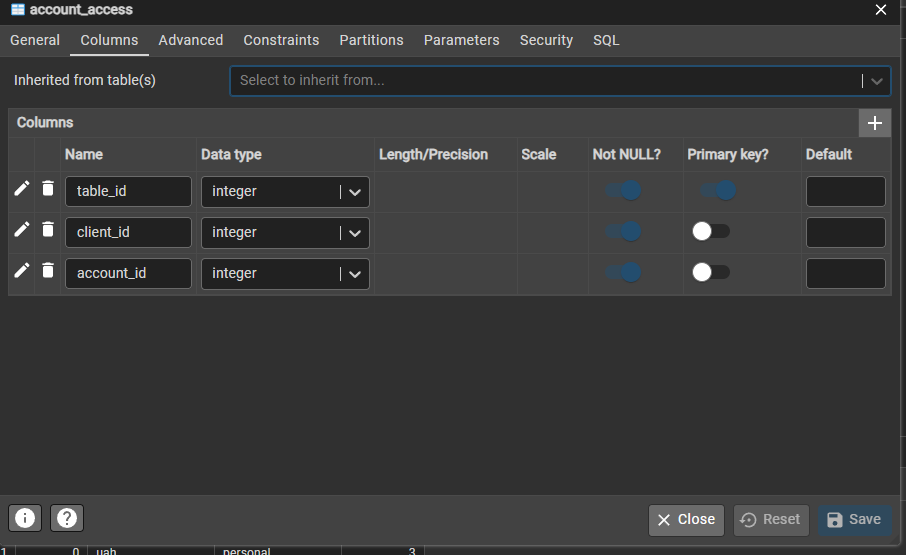
**Account**

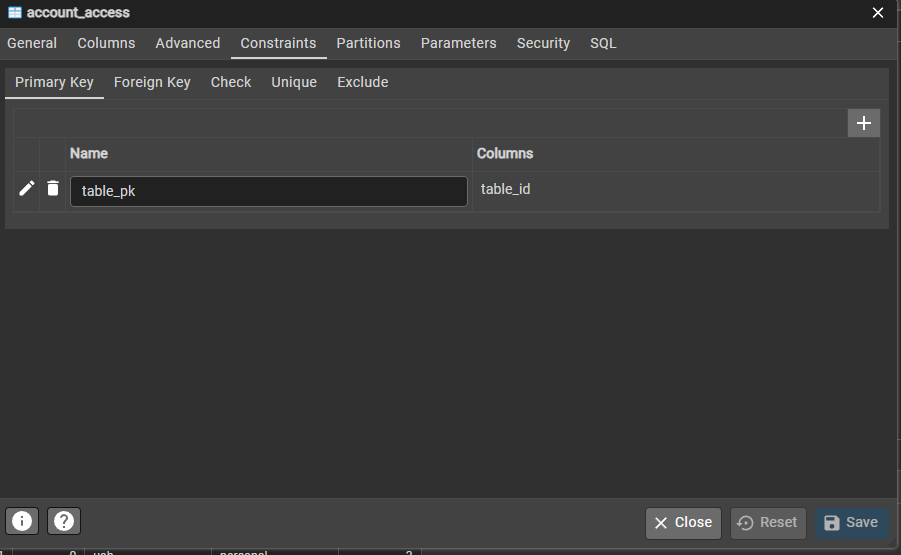
****

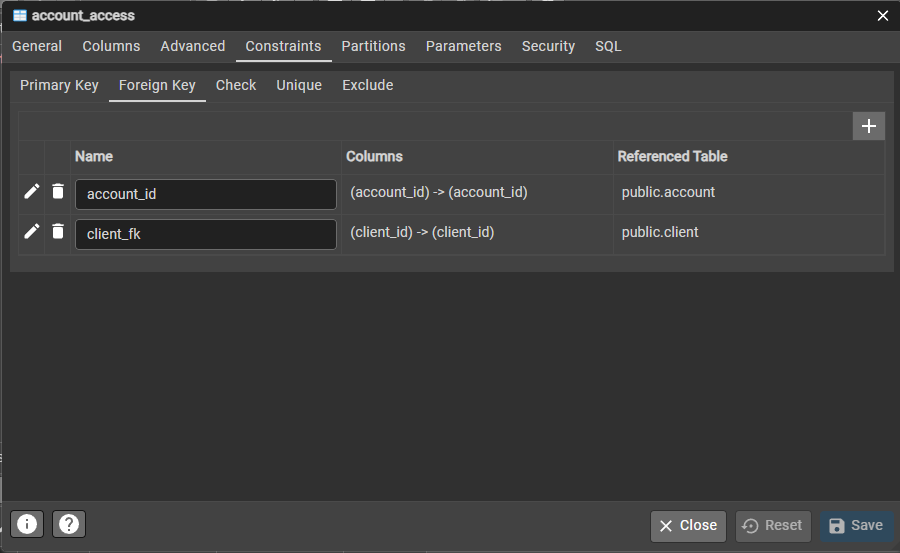
****

****

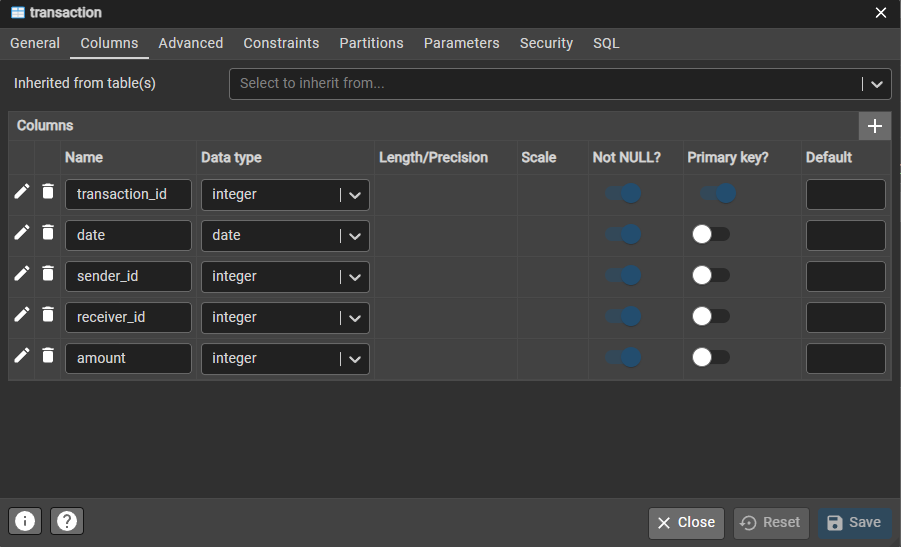
**Account\_Access**

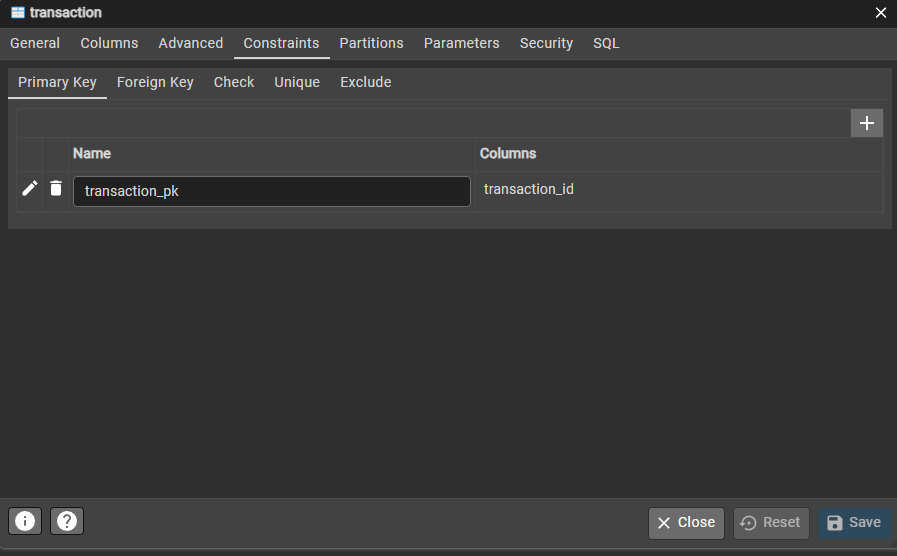
****

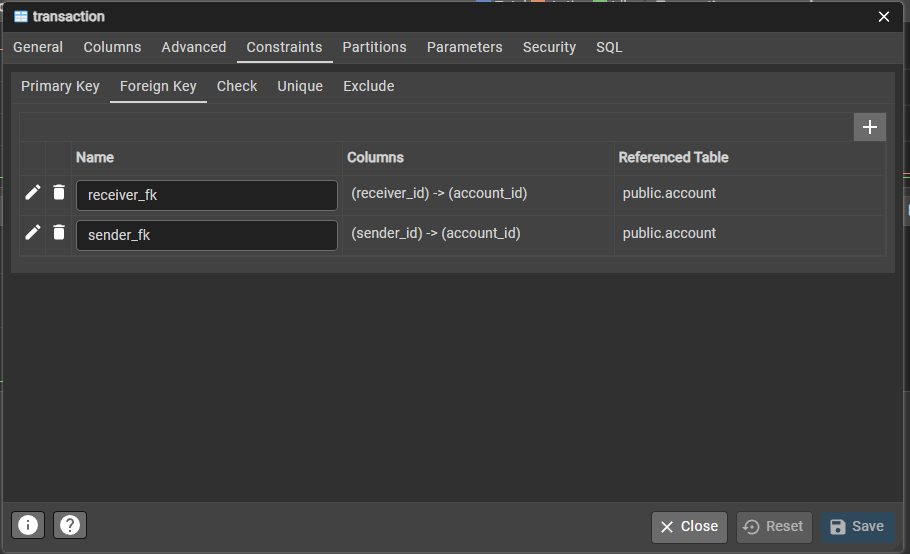
****

****

**Transaction**

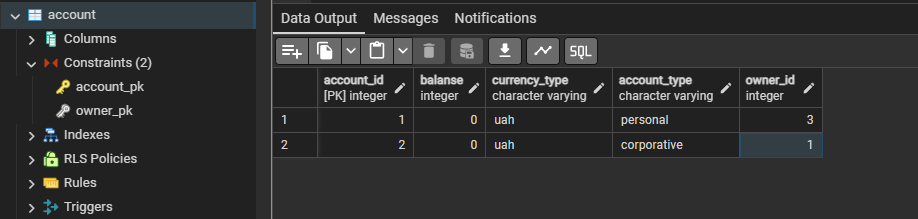
****

****

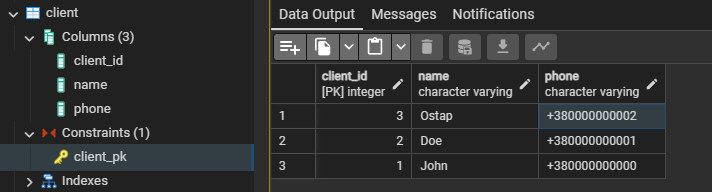
****

**Таблиці заповнені даними**

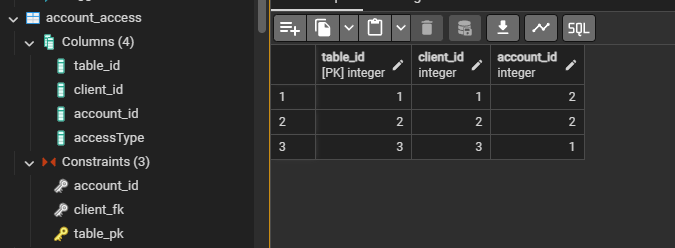
**Account**

****

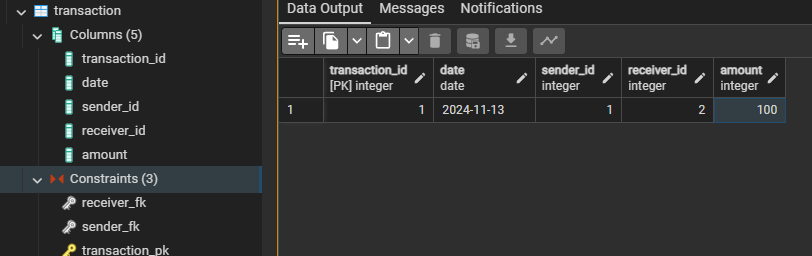
**Client**

****

**Account\_Access**

****

**Transaction**

****

**Висновки**

В ході лабораторної роботи було розроблено ER-модель предметної області, яка відповідає вимогам. Модель включає сутності з атрибутами і зв’язками між ними. Для побудови ER моделі використовувалась нотація Чена. Це дозволило правильно відобразити всі зв’язки між сутностями (зокрема зв'язки типу 1:N та M:N).

Після перетворення ER-моделі на реляційну схему бази даних, було проведено нормалізацію до третьої нормальної форми (3НФ).

За допомогою PostgreSQL та інструмента pgAdmin4 були створені таблиці бази даних, визначені типи даних атрибутів, встановлені первинні та зовнішні ключі, а також обмеження (NOT NULL, UNIQUE). Це гарантувало цілісність даних та відповідність заданим обмеженням.

У результаті лабораторної роботи було заповнено таблиці бази даних реальними даними, що дозволило перевірити коректність створеної структури та виконання базових операцій додавання і збереження даних у системі.

Репозиторій Github**:**

[**https://github.com/andriyderkach1/KV22\_Derkach\_SQL\_Lab1/**](https://github.com/andriyderkach1/KV22_Derkach_SQL_Lab1/)