

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”*

Виконала:

студентка ІII курсу

групи КВ-12

Павленко Л.П.

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.

*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.

**Модель «сутність-зв’язок» електронного каталогу для зберігання музичних нот та композицій**

Дана предметна галузь передбачає збереження музичних нот і композицій.

**Сутності**

Згідно цієї області для того щоб побудувати бази даних виділено наступні сутності:

Сутність «Електронний каталог» з атрибутами : ID, назва;

Сутність «Ноти» з атрибутами : ID, назва композиції, номер композиції;

Сутність «Композитор» з атрибутами : ID, Ім'я, Прізвище.

Сутність «Користувач» з атрибутами : ID, Ім'я, Прізвище.

**Опис зв'язків**

Користувач завантажує ноти в каталог, отже ці сутності (користувач, ноти та каталог) утворюють зв'язок. Зв'язок передбачає, що електронний каталог може містити багато нот композицій, як і ноти композиції можуть зберігатися в декількох каталогах, отже зв'язок M:N. Користувач може додавати декілька нот композицій в декілька каталогів, отже зв’язок в цьому випадку M:N. Декілька користувачів можуть додати композицію в каталог, отже зв’язок також M:N

Композитор може написати декілька творів, натомість ноти композиції зазвичай мають одного композитора. Отже між сутностями Композитор та Ноти зв’язок 1:N.

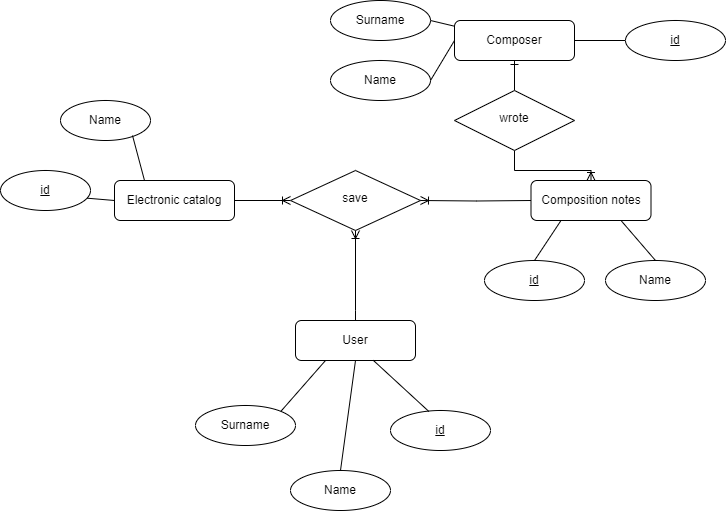


Рисунок 1 – ER-діаграма, побудована за нотацією “Пташиної лапки (Crow’s foot)”

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

**бази даних**

Сутність «Каталог» було перетворено в таблицю «Catalog»

Сутність «Ноти» було перетворено в таблицю «Notes».

Сутність «Композитор» було перетворено в таблицю «Composer».

Сутність «Користувач» було перетворено в твблицю «User».

Опис даних сутностей, їх атрибути та типи атрибутів вказані в таблиці 1.

Таблиця 1 – Опис структури бази даних «Електронний каталог для зберігання музичних нот та композицій»

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| **Catalog** *– містить дані про каталог* | **CatalogID –** *унікальний ідентифікатор каталогу*  **Catalog name** – *назва каталогу* | **integer** *(числовий)*  **character varying** *(рядок)* |
| **Notes –** *містить дані про ноти* | **NotesID –** *унікальний ідентифікатор нот*  **ComposerID –** *унікальний ідентифікатор користувача*  **Composition name –** *назва нот*  **UserID –** *унікальний ідентифікатор користувача* | **integer** *(числовий)*  **character varying** *(рядок)*  **integer** *(числовий)*  **integer** *(числовий)* |
| **User –** *містить дані про користувача, який додає ноти* | **UserID –** *унікальний ідентифікатор користувача*  **Name** – *Ім’я користувача*  **Surname** – *Призвіще користувача* | **integer** *(числовий)*  **character varying** *(рядок)*  **character varying** *(рядок)* |
| **Composer –***містить дані про композитора який створив ноти* | **ComposerID –** *унікальний ідентифікатор користувача*  **Name** – *Ім’я користувача*  **Surname** – *Призвіще користувача* | **integer** *(числовий)*  **character varying** *(рядок)*  **character varying** *(рядок)* |
| **Saving\_notes**  *містить інформацію про взаємозв’язок каталогу, нот та користувача* | **Saving\_notesID –** *унікальний ідентифікатор взаємозв’язку між катологом, нотами та користувачем;*    **CatalogID -** *унікальний ідентифікатор каталогу*  **NotesID –** *унікальний ідентифікатор нот*  **UserID –** *унікальний**ідентифікатор користувача* | **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)*  **integer** *(числовий)* |

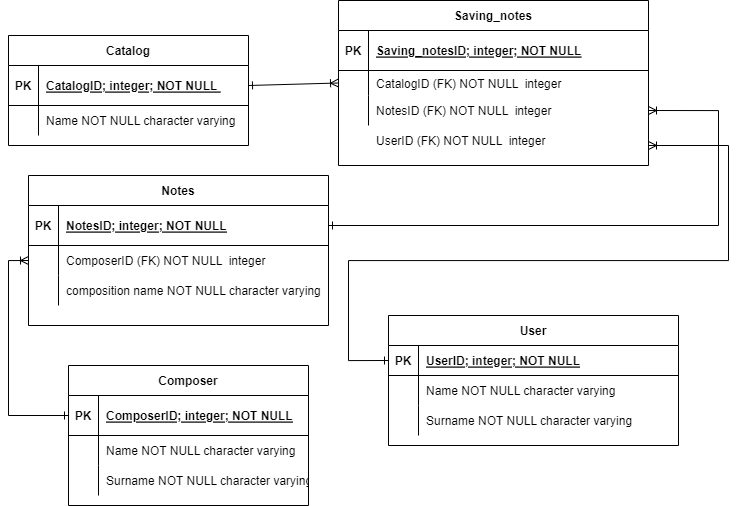


Рисунок 2 – Схема бази даних

**Функціональні залежності**

**Catalog:**

**CatalogID –** *унікальний ідентифікатор каталогу*

**Name –** *назва каталогу*

CatalogID → Name*(назва каталогу залежить від її ідентифікатора)*

**Notes:**

**NotesID –** *унікальний ідентифікатор нот;*

**ComposerID –** *ідентифікатор композитора, який написав цю композицію;*

**Composition name –** *назва композиції;*

NotesID → ComposerID, Composition name

NotesID → Composition name (назва композиції залежить від номеру нот)

NotesID → ComposerID

Присутні транзитивні відношення, але вони включають ключовий атрибут.

**Composer:**

**ComposerID –** *унікальний ідентифікатор композитора;*

**Name –** *Ім’я композитора;*

**Surname –** *Призвіще композитора;*

ComposerID → Name, Surname

ComposerID → Name (ім’я композитора залежить від індивідуального ідентифікатора автора)

ComposerID → Surname (призвіще композитора залежить від індивідуального номера автора)

**User:**

**UserID –** *унікальний ідентифікатор користувача;*

**Name –** *Ім’я користувача;*

**Surname –** *Прізвище користувача;*

UserID → Name, Surname

UserID → Name (ім’я композитора залежить від ідентифікатора користувача)

UserID → Surname (призвіще користувача залежить від ідентифікатора користувача)

**Saving\_notes:**

**Saving\_notesID** **–** *унікальний ідентифікатор взаємозв’язку між катологом, нотами та користувачем (тобто додавання нот користувачем в каталог);*

**CatalogID** **–** *унікальний ідентифікатор каталогу;*

**NotesID** **–** *унікальний ідентифікатор нот;*

**UserID** **–** *унікальний ідентифікатор користувача;*

Saving\_notesID → CatalogID, NotesID, UserID;

NotesID → CatalogID, UserID;

CatalogID → NotesID, UserID;

UserID → CatalogID, NotesID;

Хоч в даному випадку і присутні транзитивні відношення, але вони включають в себе ключовий атрибут.

Схема відповідає 1НФ, тому що кожна комірка містить унікальне та атомарне значення та також кожен запис є унікальним.

Схема відповідає 2НФ, тому що таблиця знаходиться в 1НФ та кожний атрибут який не є ключовим залежить від первинного ключа.

Схема відповідає 3НФ, тому що таблиця знаходиться в 2НФ та кожний неключовий атрибут не є транзитивно залежним від первинного ключа.

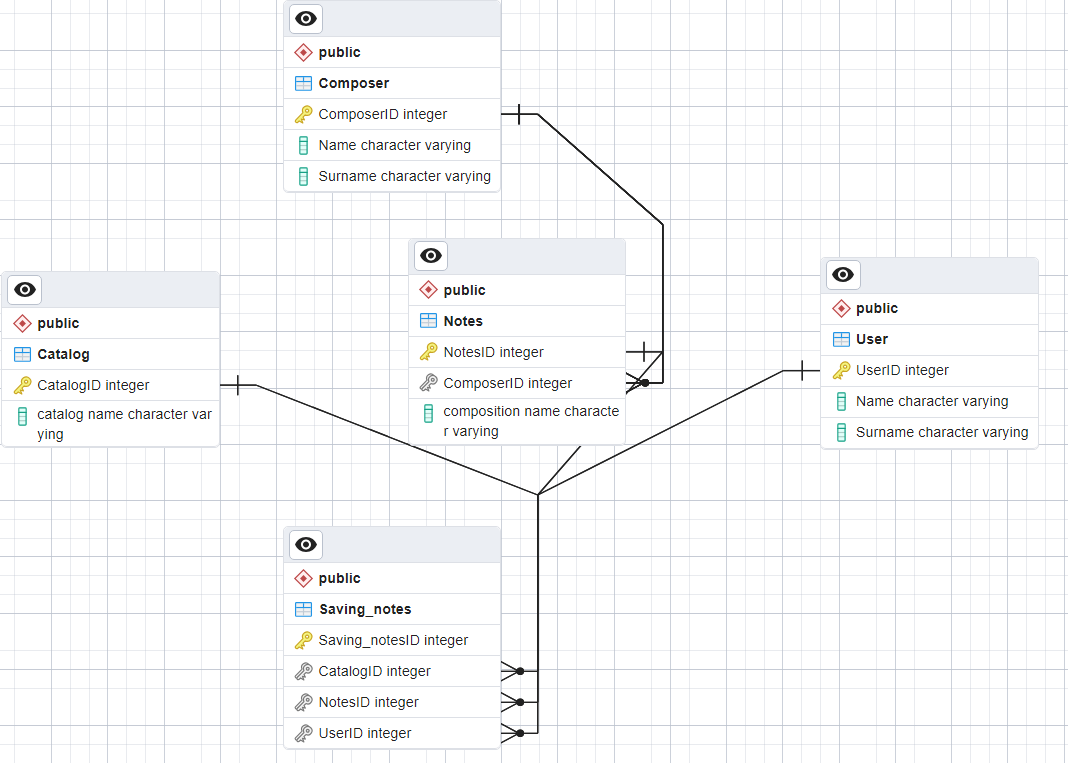
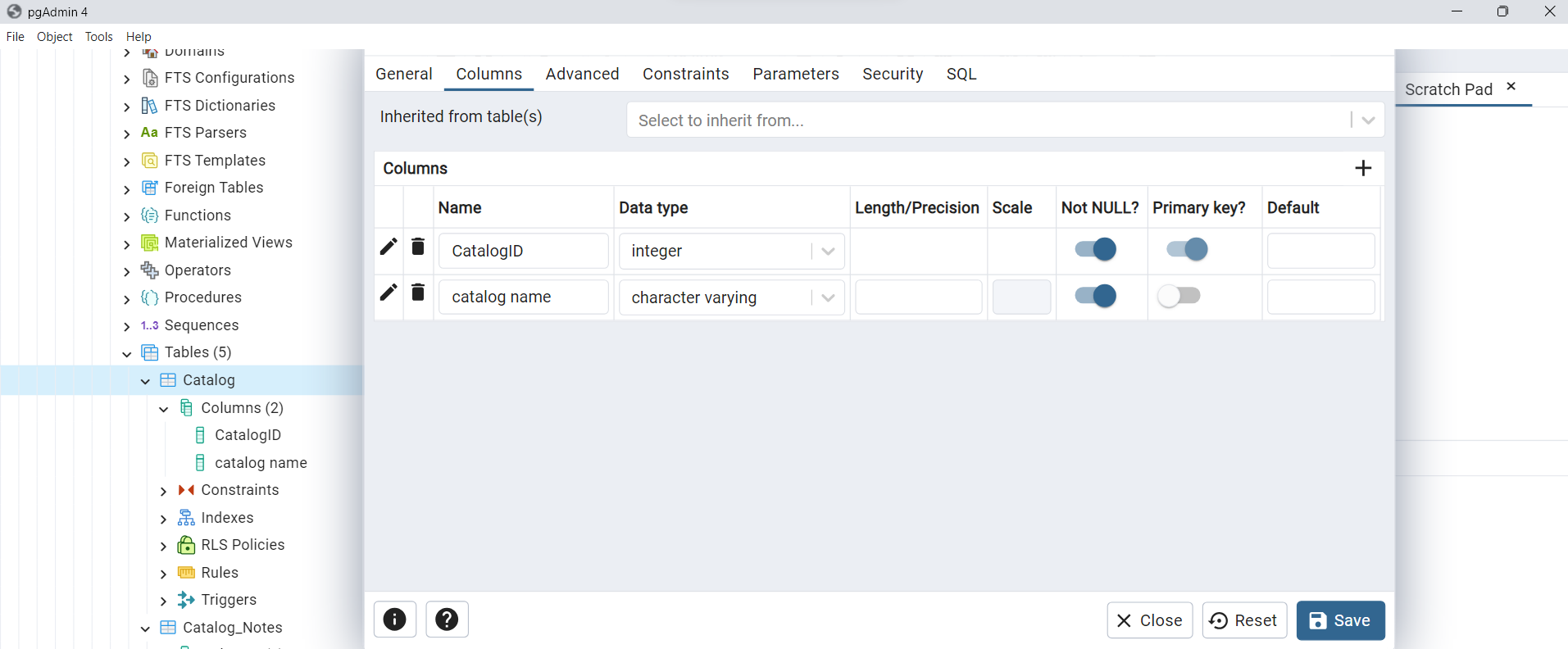
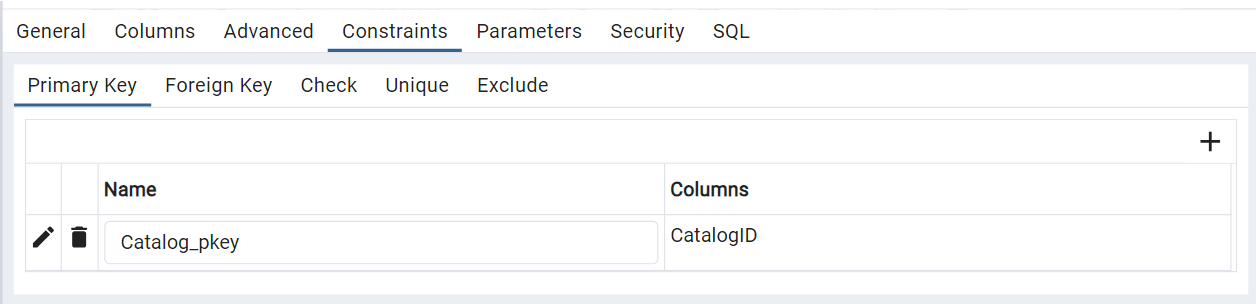


Рисунок 3 – Схема бази даних у pgAdmin 4

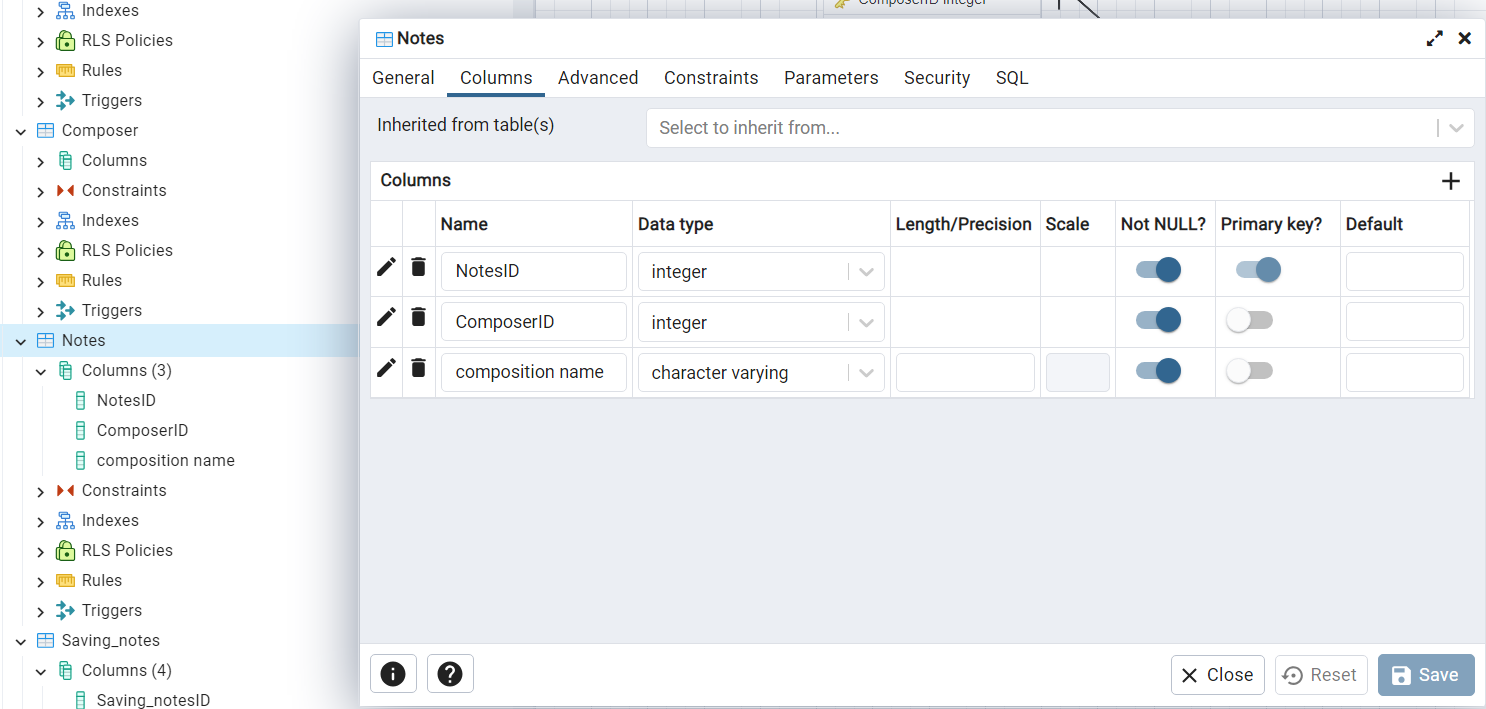
**Таблиці бази даних у pgAdmin4**

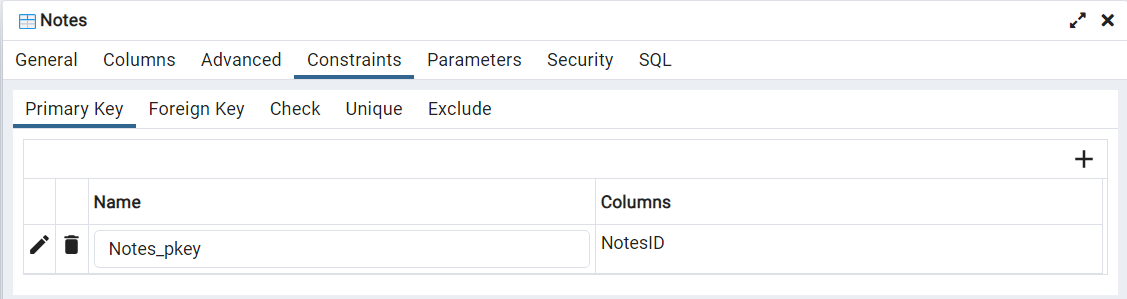
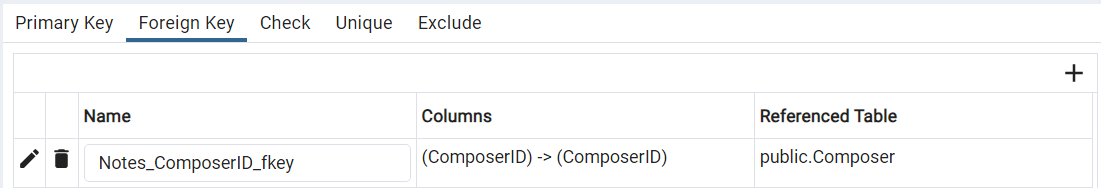
**Catalog**

****

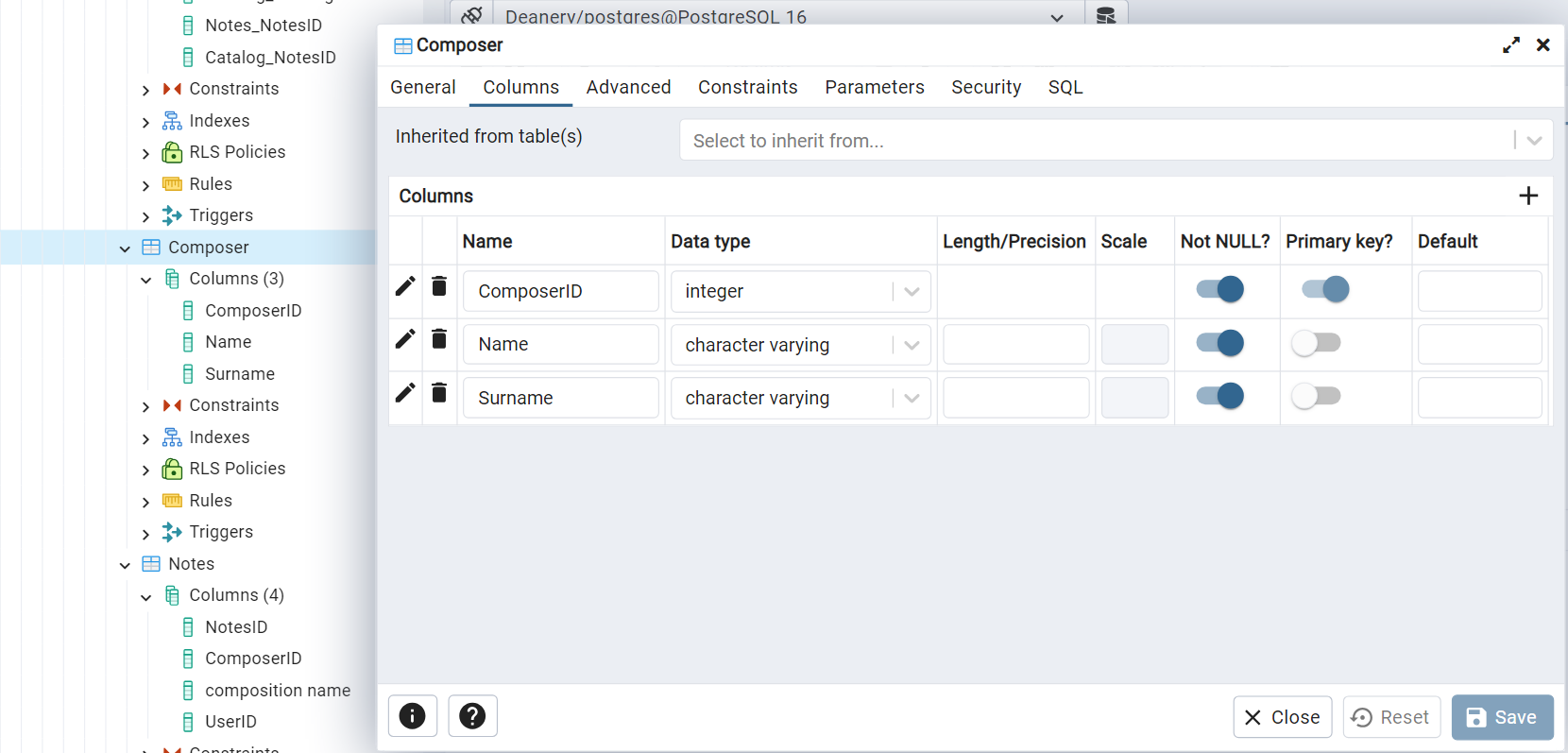
****

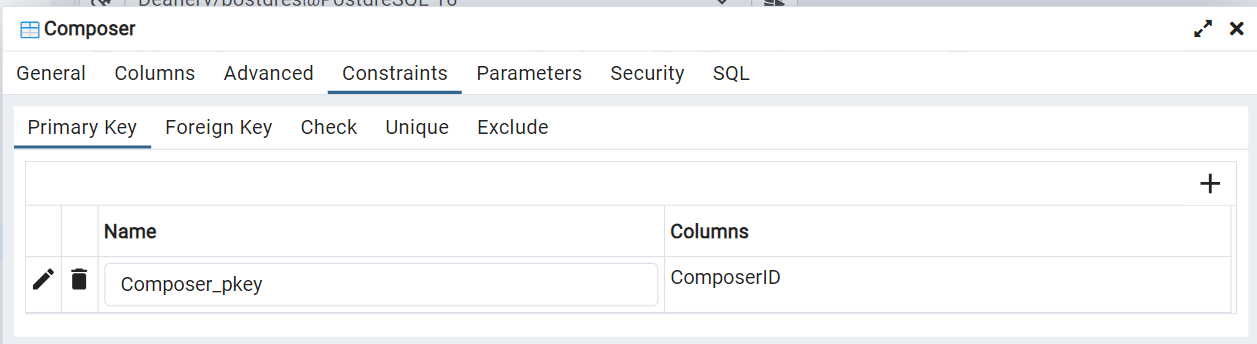
**Notes**

****

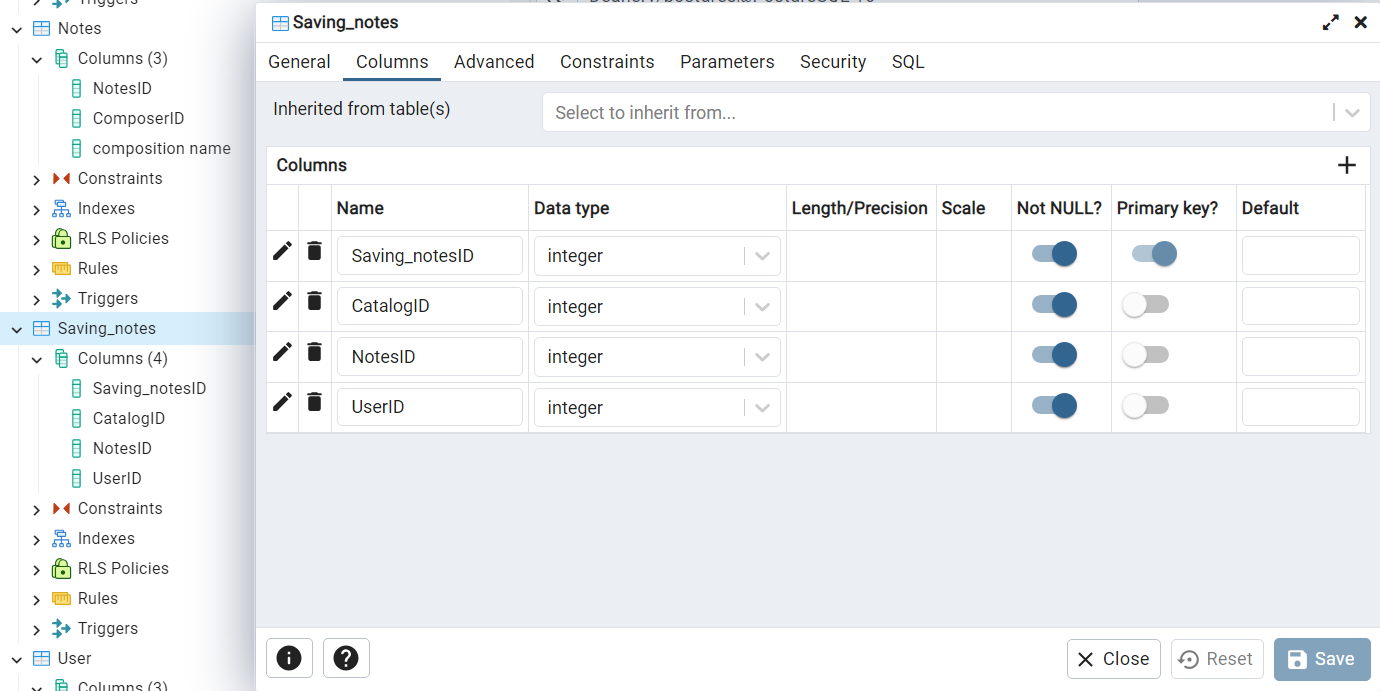
****

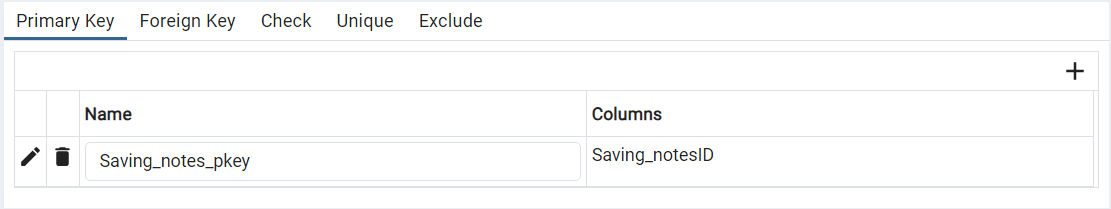
**Composer**

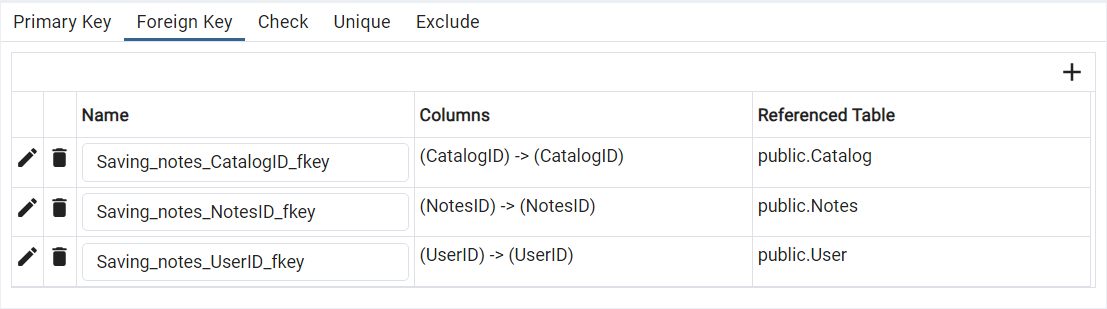
****

****

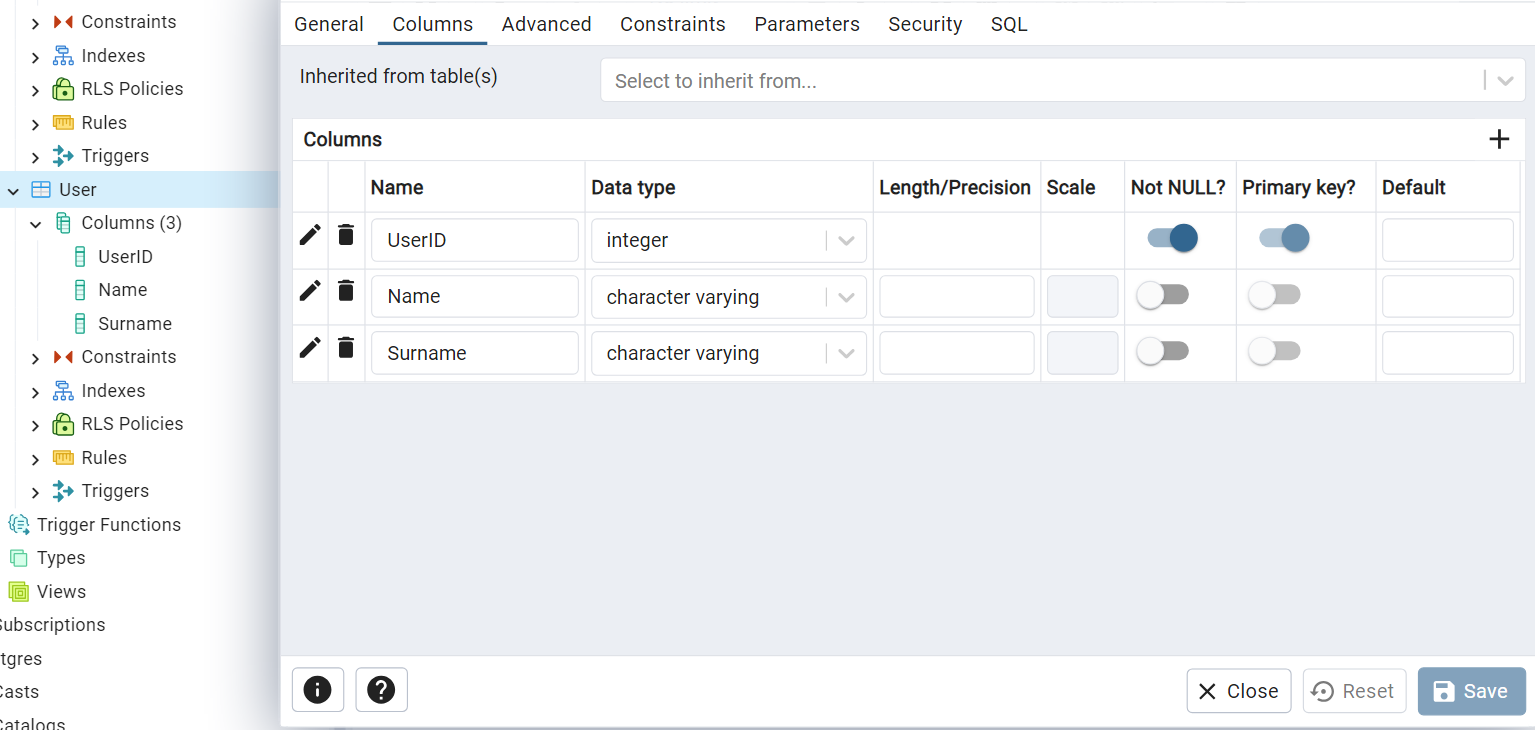
**Saving\_notes**

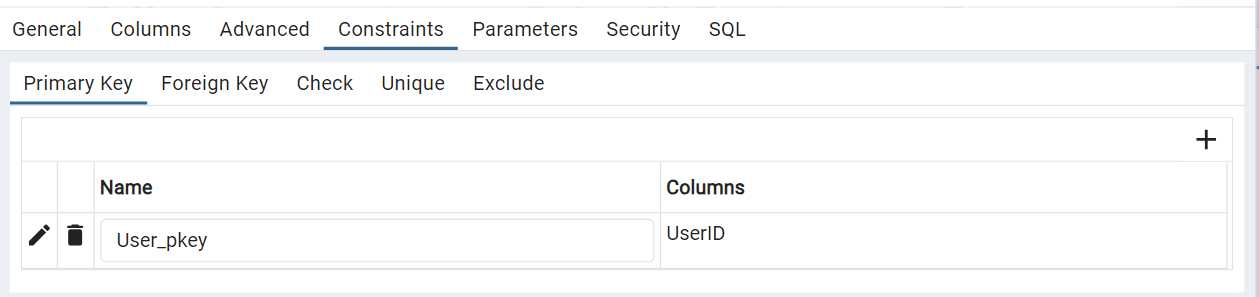
****

****

****

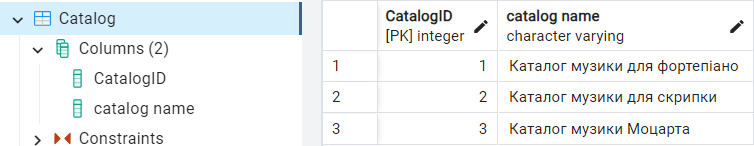
**User**

****

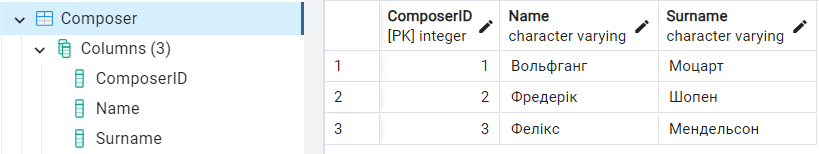


**Таблиці заповнені даними**

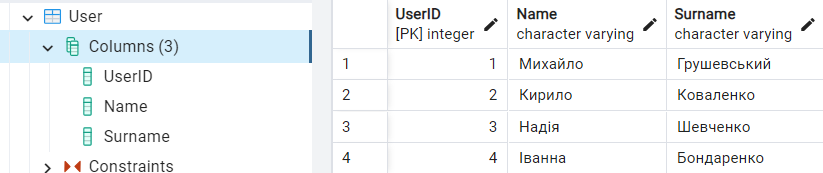
**Таблиця «Catalog»**

****

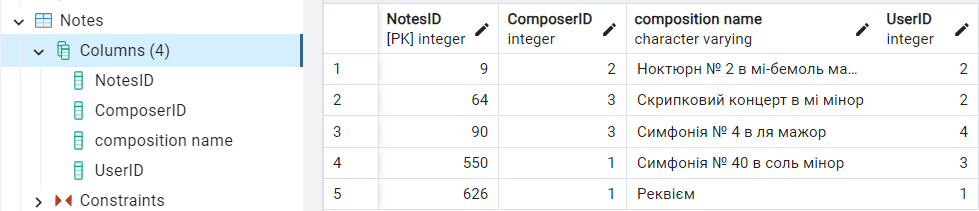
**Таблиця «Composer»**

****

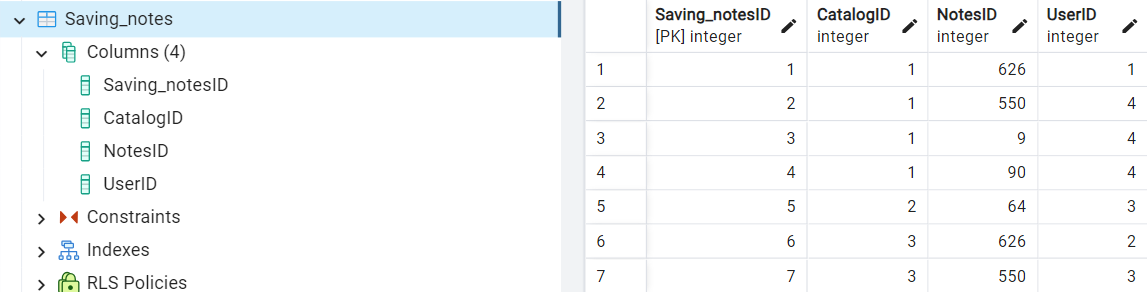
**Таблиця «User»**

****

**Таблиця «Notes»**

****

**Таблиця «Saving\_notes»**

****

Telegram: <https://t.me/lumi_789>

GitHub: https://github.com/liudapavlenko/lab\_1.git