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”*

Виконала:

студентка ІІI курсу

групи КВ-22

Гарматюк К.О.

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 управління завданнями та проектами фрілансерів. Вона охоплює сукупність інструментів і процесів, які спрямовані на організацію, моніторинг і виконання робочих завдань. Система забезпечує облік виконаних проектів, контроль термінів, розподіл ресурсів між учасниками.

**Опис сутностей**

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

1. Фрілансер (Freelancer)

Атрибути: ідентифікатор фрілансера, ім’я, електронна пошта, рейтинг.

Призначення: збереження даних щодо фрілансерів.

1. Проєкт (Project)

Атрибути: ідентифікатор проєкту, назва, дата створення, статус (активний, завершений).

Призначення: збереження даних щодо проєктів.

1. Завдання (Task)

Атрибути: ідентифікатор завдання, опис, термін виконання (Deadline), ідентифіктор фрілансера який над ним працює.

Призначення: збереження даних щодо завдань, на які розбиті проєкти.

1. Навичка (Skill)

Атрибути: ідентифікатор навички, назва, галузь.

Призначення: збереження даних щодо навичок, якими володіють фрілансери.

**Опис зв’язків між сутностями**

Зв’язок «Фрілансер» - «Завдання» є зв’язком 1:N». Один фрілансер може виконувати багато завдань, але одне завдання може бути виконане лише одним фрілансером.

Зв’язок «Проєкт» - «Завдання» є зв’язком 1:N. Один проєкт може включати багато завдань, але кожне завдання належить лише одному проєкту.

Зв’язок «Фрілансер» - «Навичка» є зв’язком N:M. Один фрілансер може володіти багатьма навичками, і одна навичка може бути притаманна багатьом фрілансерам.

Зв’язок «Завдання» - «Навичка» є зв’язком N:M. Одне завдання може вимагати від фрілансера володіння багатьма навичками, і одна навичка може бути потрібна для багатьох завданнь.

Графічне подання концептуальної моделі «Сутність-зв’язок» зображено на рисунку 1.

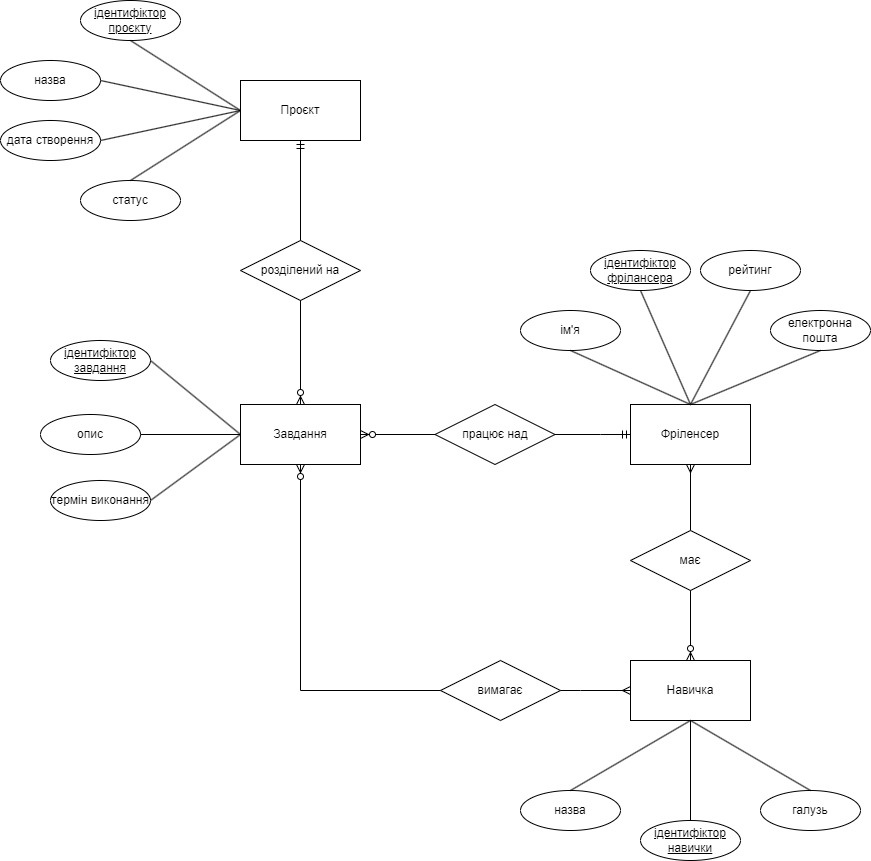


Рисунок 1 – ER-діаграма, побудована за нотацією Чена

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

Сутність Фрілансер перетворено в таблицю Freelancer з первинним ключем freelancer\_id та атрибутами name, email, rating.

Сутність Проєкт перетворено в таблицю Project з первинним ключем project\_id та атрибутами title, creation\_date, status.

Сутність Завдання перетворено в таблицю Task з первинним ключем task\_id та атрибутами description, deadline, а також зовнішнім ключем project\_id, що вказує на таблицю Project, і freelancer\_id, що вказує на таблицю Freelancer.

Сутність Навичка перетворено в таблицю Skill з первинним ключем skill\_id та атрибутами name і field.

Сутність Task-Skill Requirement створено як окрему таблицю Task\_Skill для реалізації зв’язку N:M між таблицями Task та Skill. Таблиця містить первинний ключ task\_skill\_id та зовнішні ключі task\_id і skill\_id, а також атрибут required\_level (basic, intermediate, advanced).

Сутність Freelancer-Skill створено як окрему таблицю Freelancer\_Skill для реалізації зв’язку N:M між таблицями Freelancer та Skill. Таблиця містить первинний ключ freelancer\_skill\_id та зовнішні ключі freelancer\_id і skill\_id, а також атрибут skill\_level.

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

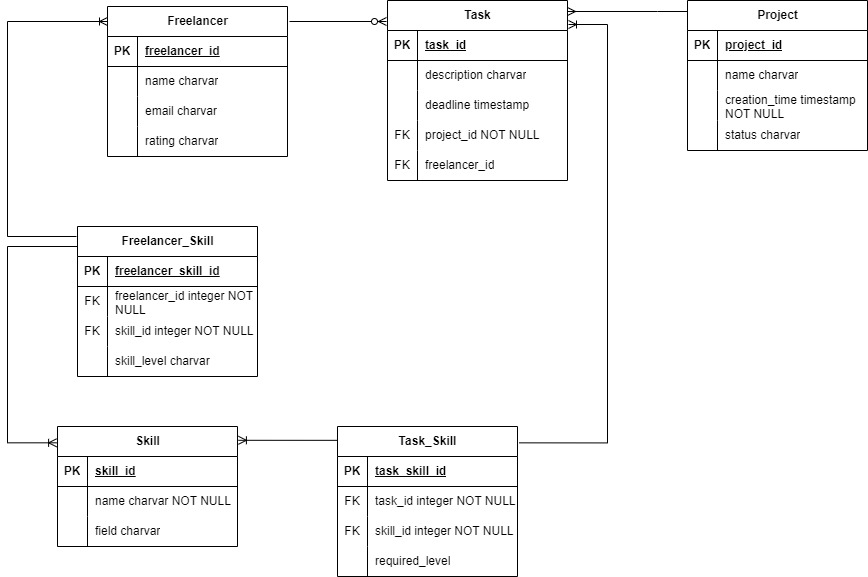
****

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** |
| **Freelancer** – містить дані про фрілансерів | freelancer\_id – унікальний ідентифікатор фрілансера | *integer* (число) |
|  | name – ім’я фрілансера | *character varying* (рядок) |
|  | email – електронна пошта фрілансера | *character varying* (рядок) |
|  | rating – рейтинг фрілансера | *character varying* (рядок) |
| **Project** – містить дані про проєкти | project\_id – унікальний ідентифікатор проєкту | *integer* (число) |
|  | name – назва проєкту | *character varying* (рядок) |
|  | creation\_time – дата створення проєкту | *timestamp* (дата та час) |
|  | status – статус проєкту | *character varying* (рядок) |
| **Task** – містить дані про завдання | task\_id – унікальний ідентифікатор завдання | *integer* (число) |
|  | description – опис завдання | *character varying* (рядок) |
|  | deadline – термін виконання завдання | *timestamp* (дата та час) |
|  | project\_id – ідентифікатор проєкту | *integer* (число) |
|  | freelancer\_id – ідентифікатор фрілансера | *integer* (число) |
| **Skill** – містить дані про навички | skill\_id – унікальний ідентифікатор навички | *integer* (число) |
|  | name – назва навички | *character varying* (рядок) |
|  | field – галузь, до якої належить навичка | *character varying* (рядок) |
| **Task\_Skill** – вимоги до навичок для завдання | task\_skill\_id – унікальний ідентифікатор | *integer* (число) |
|  | task\_id – ідентифікатор завдання | *integer* (число) |
|  | skill\_id – ідентифікатор навички | *integer* (число) |
|  | required\_level – необхідний рівень | character varying (рядок) |
|  | freelancer\_skill\_id – унікальний ідентифікатор |  |
| **Freelancer\_Skill** – навички фрілансера | |  | | --- | | freelancer\_id – ідентифікатор фрілансера |  |  | | --- | |  | | |  | | --- | | *integer (число)* |  |  | | --- | |  | |
|  | |  | | --- | | skill\_id – ідентифікатор навички |  |  | | --- | |  | | |  | | --- | | *integer (число)* |  |  | | --- | |  | |
|  | |  | | --- | | skill\_level – рівень володіння |  |  | | --- | |  | | *character varying (рядок)* |

**Функціональні залежності для кожної таблиці**

1. **Freelancer**:

* freelancer\_id -> { name, email, rating }
* email -> { freelancer\_id, name, rating }

1. **Project**:

* project\_id -> { name, creation\_time, status }

1. **Task**:

* task\_id -> { description, deadline, project\_id, freelancer\_id }
* project\_id -> { task\_id, description, deadline, freelancer\_id }
* freelancer\_id -> { task\_id, description, deadline, project\_id }

1. **Skill**:

* skill\_id -> { name, field }
* name, field -> { skill\_id }

1. **Task\_Skill**:

* task\_skill\_id -> { task\_id, skill\_id, required\_level }
* task\_id -> { task\_skill\_id, skill\_id, required\_level }
* skill\_id -> { task\_skill\_id, task\_id, required\_level }

1. **Freelancer\_Skill**:

* freelancer\_skill\_id -> { freelancer\_id, skill\_id, skill\_level }
* freelancer\_id -> { freelancer\_skill\_id, skill\_id, skill\_level }
* skill\_id -> { freelancer\_skill\_id, freelancer\_id, skill\_level }

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**

**Перша нормальна форма (1НФ):**  
Для відповідності 1НФ необхідно, щоб:

* У кожній клітинці таблиці зберігалося лише одне атомарне значення.
* Кожен запис був унікальним і не містив дублікатів.  
  У запропонованій моделі всі таблиці відповідають цим вимогам: кожен атрибут містить атомарні значення, а записи унікальні завдяки використанню первинних ключів. Отже, схема перебуває у першій нормальній формі.

**Друга нормальна форма (2НФ):**  
Для відповідності 2НФ необхідно:

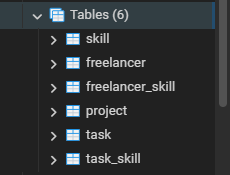
* Схема повинна відповідати 1НФ.
* Усі неключові атрибути повинні функціонально залежати від усього первинного ключа, а не його частини (для складених ключів).  
  У цій схемі всі таблиці мають чітко визначений первинний ключ, і всі неключові атрибути функціонально залежні від цього ключа. Жодна таблиця не містить складених ключів, де частина ключа могла б впливати на залежність атрибутів. Тому схема відповідає другій нормальній формі.

**Третя нормальна форма (3НФ):**  
Для відповідності 3НФ необхідно:

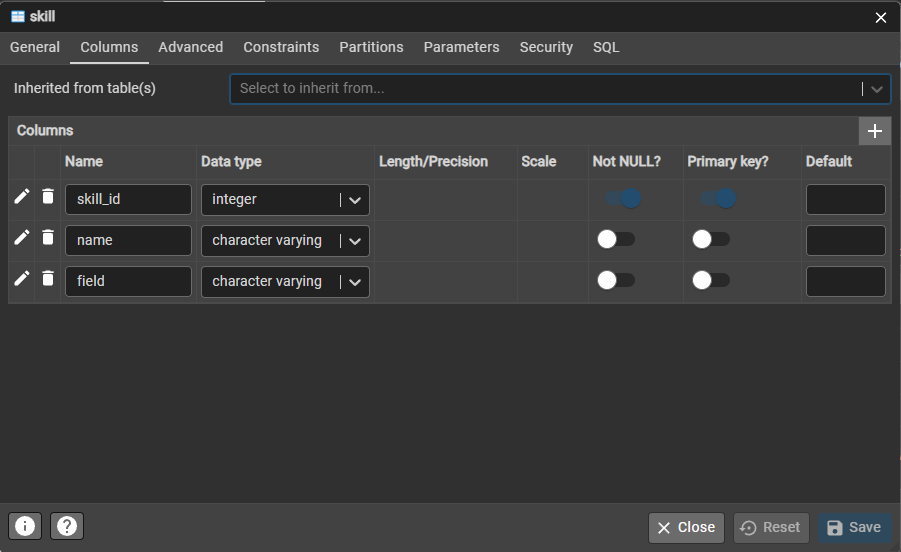
* Схема повинна відповідати 2НФ.
* Жоден неключовий атрибут не має функціонально залежати від іншого неключового атрибута (виключення транзитивних залежностей).  
  У цій моделі атрибути в таблицях залежать лише від первинних ключів і не мають транзитивних залежностей. Отже, схема відповідає третій нормальній формі.

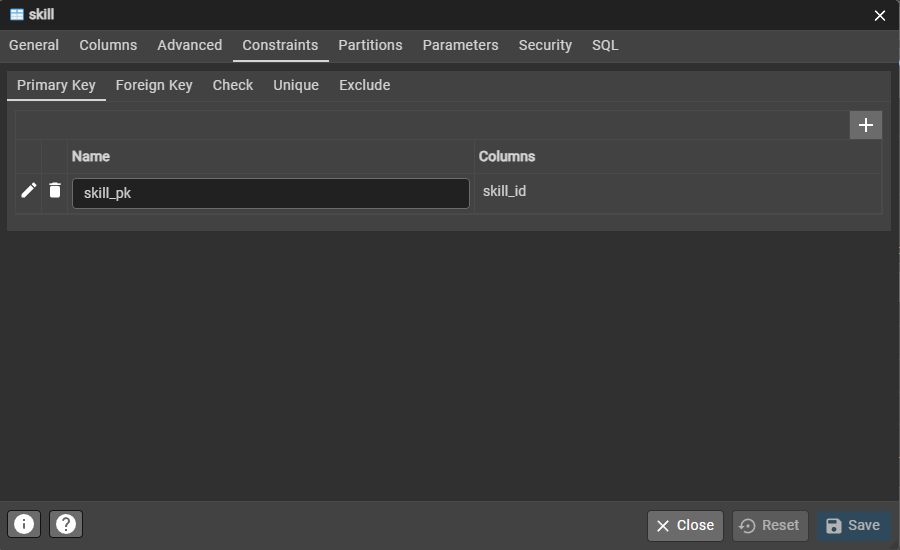
**Висновок:**  
Розроблена схема бази даних відповідає нормалізаційним вимогам 1НФ, 2НФ та 3НФ. Вона є структурованою та оптимізованою для зберігання й обробки даних, що забезпечує ефективність та мінімізує надмірність.

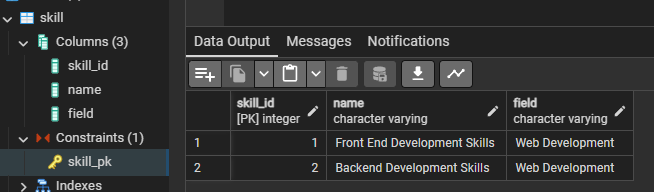
**Таблиці бази даних у pgAdmin4**

****

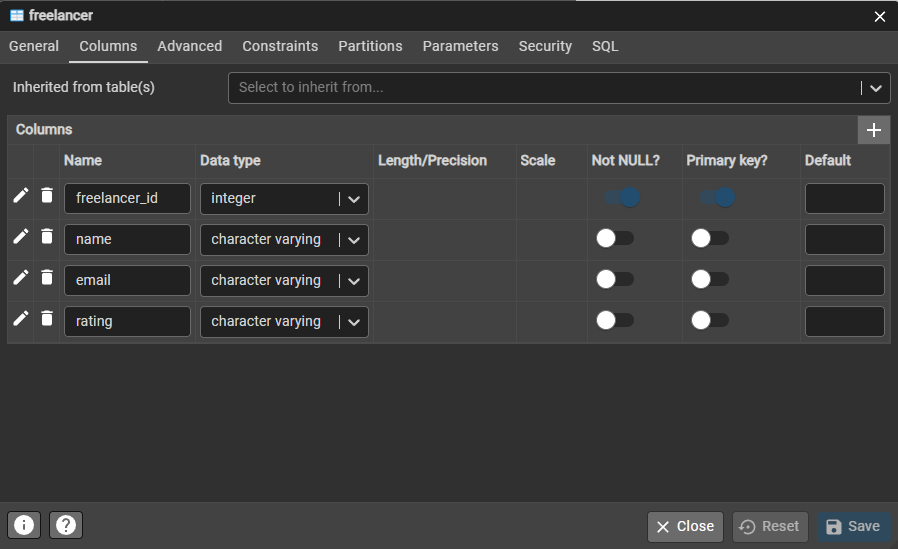
Skill

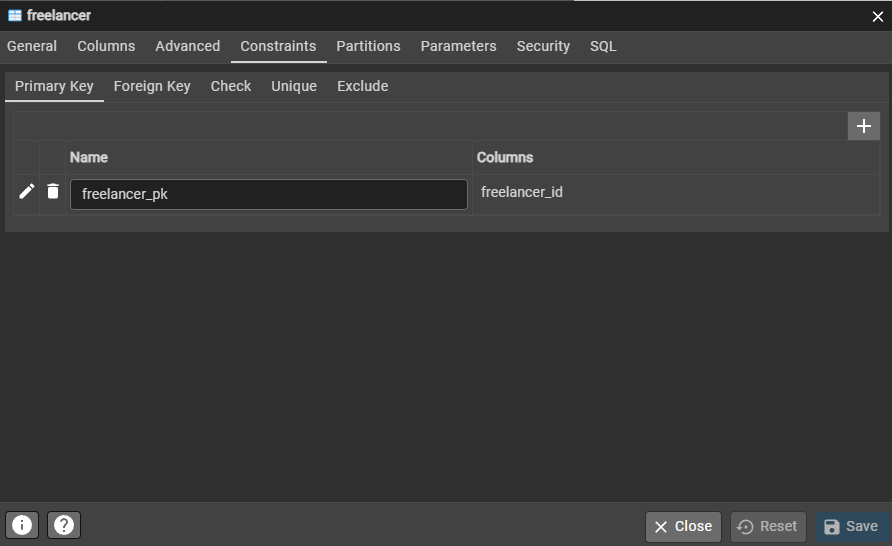
****

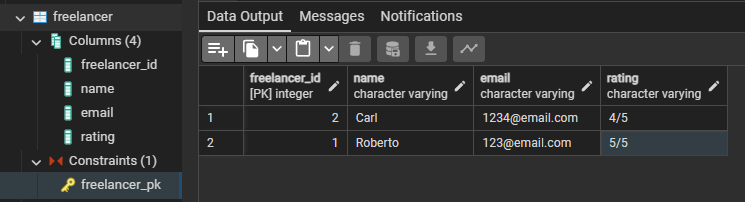
****



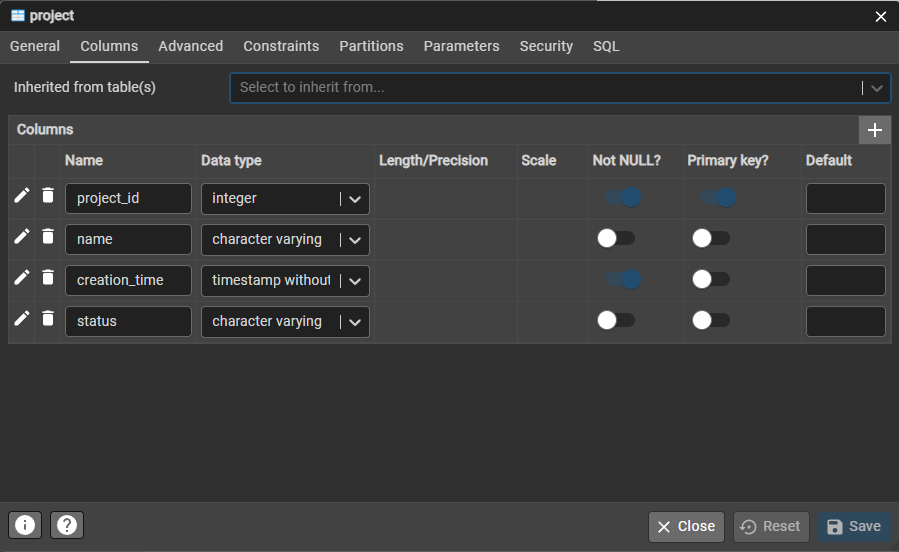
Freelancer

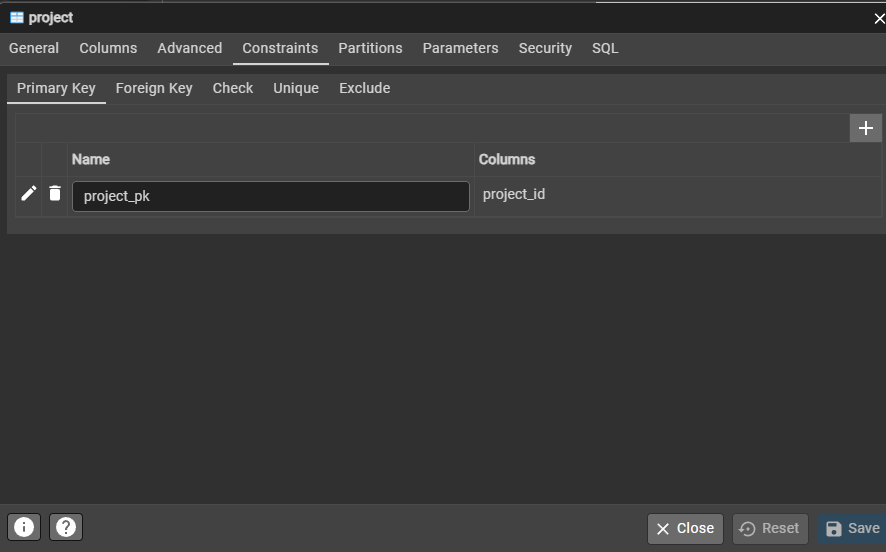
****

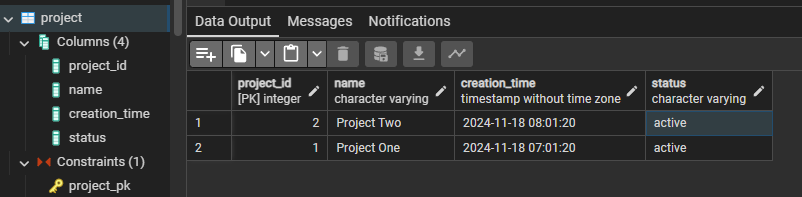
****

****

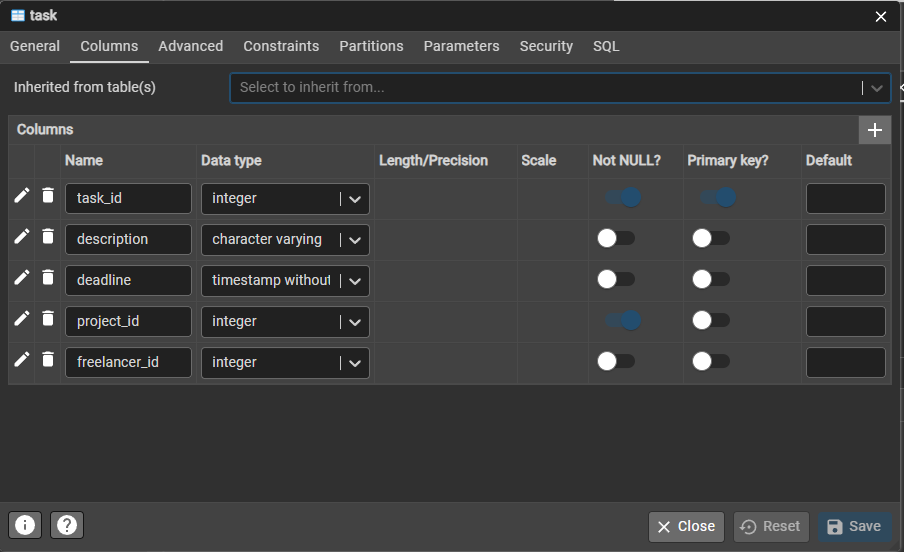
Project

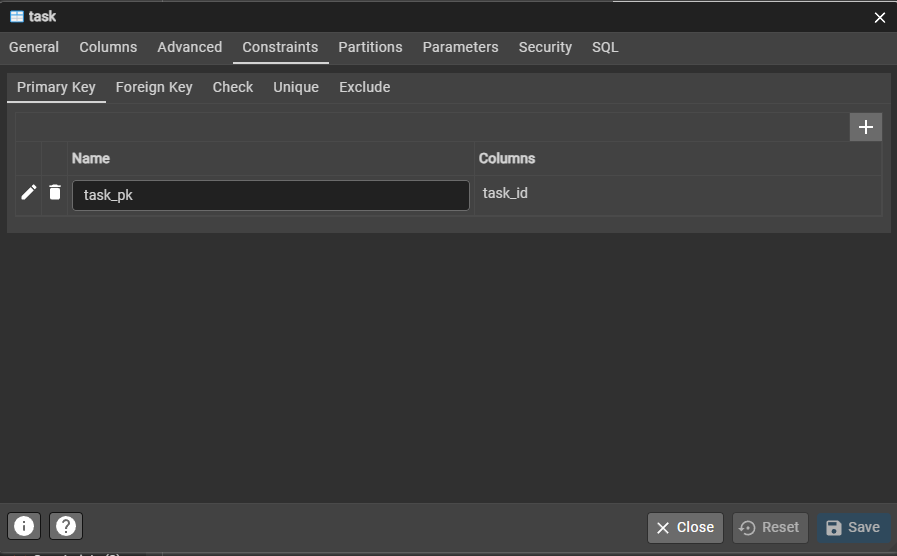
****

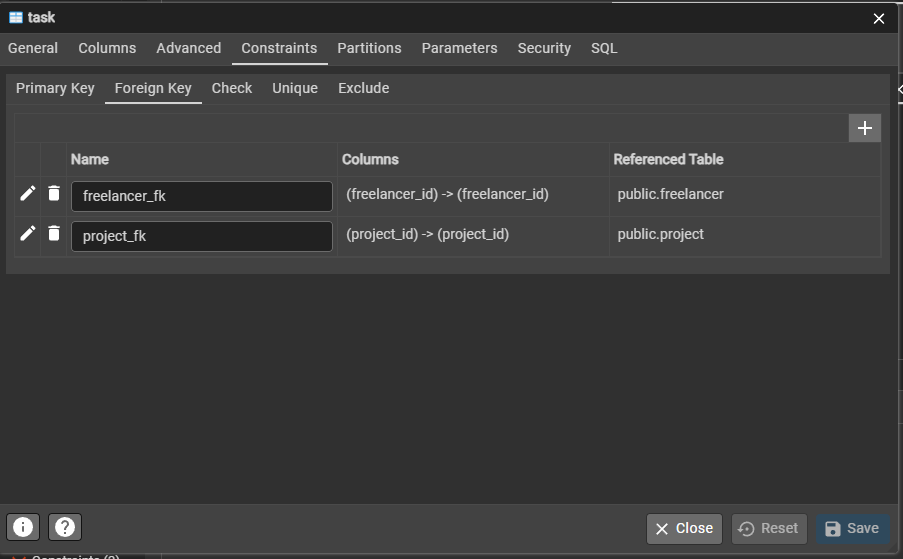
****

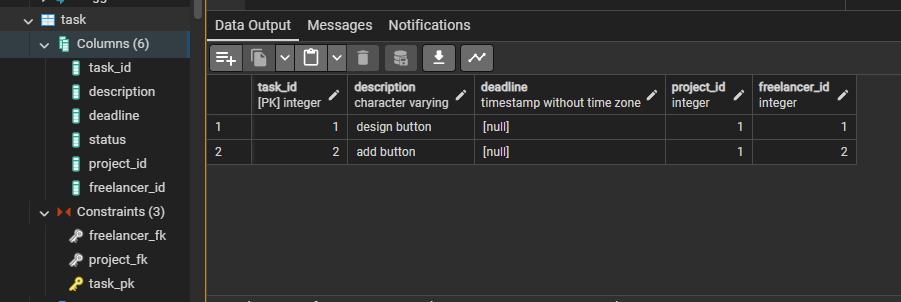
****

Task

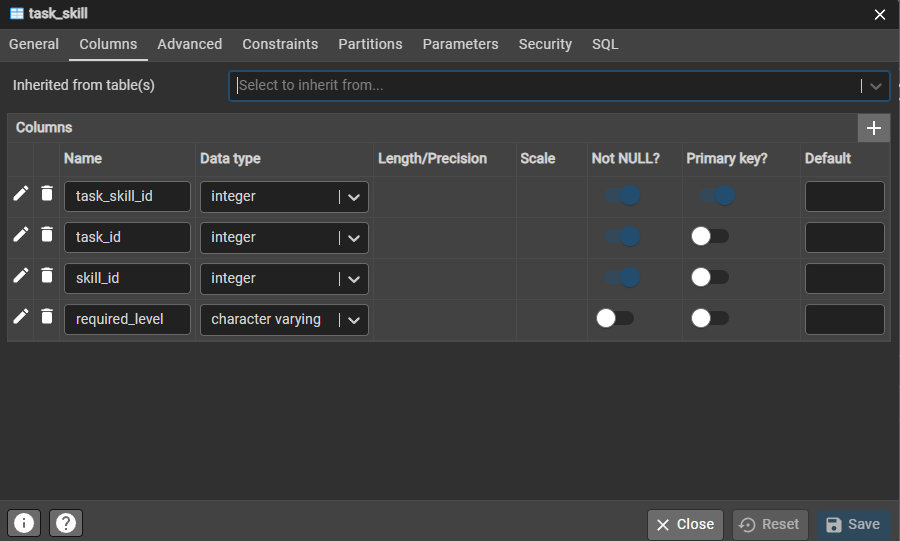
****

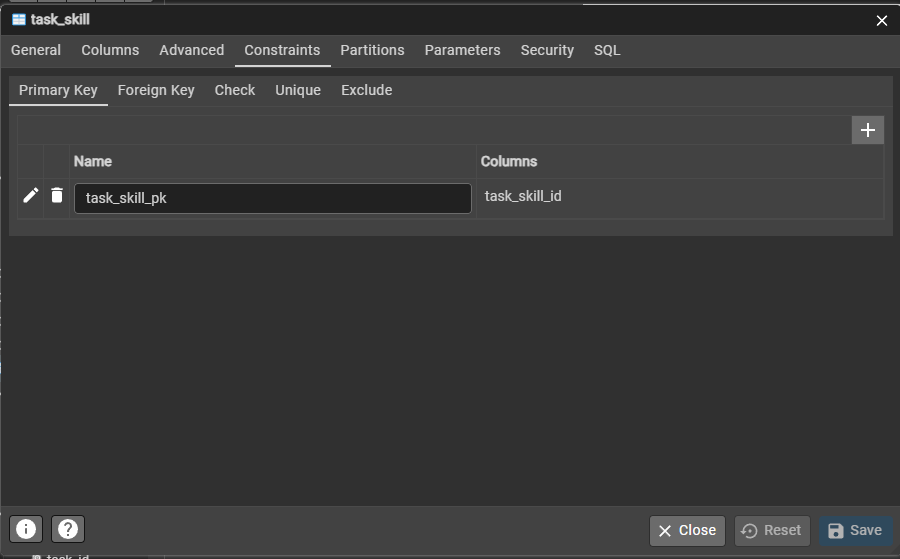
****

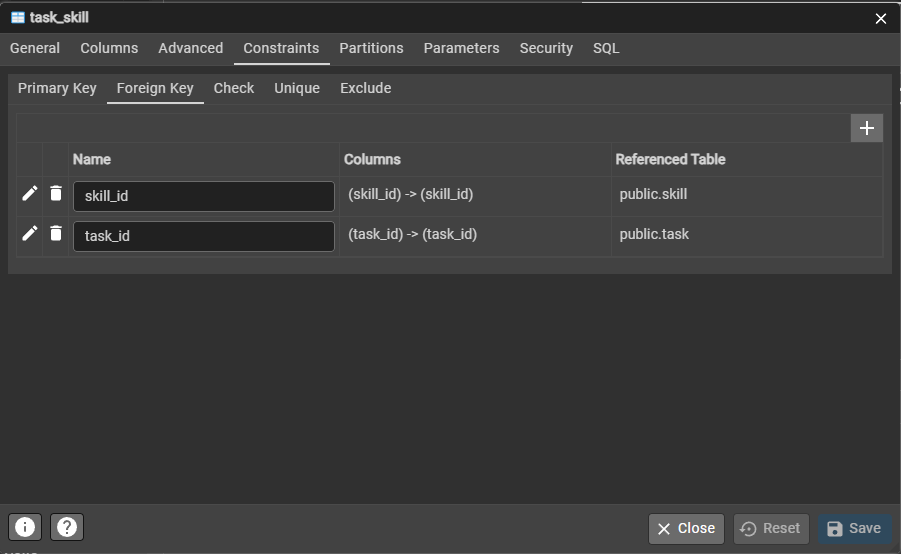
****

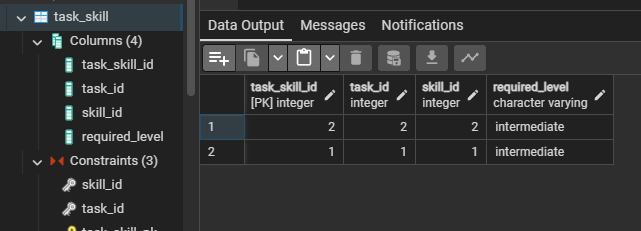
****

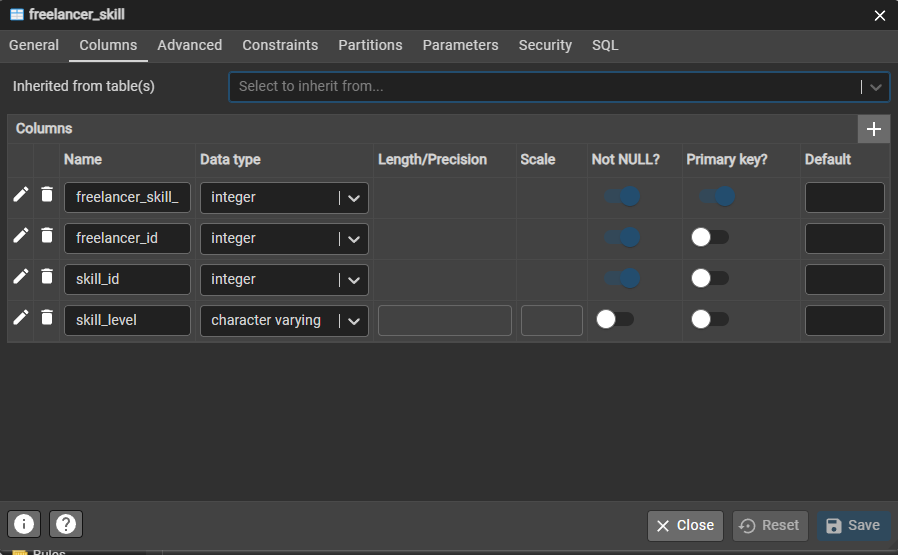
Task\_Skill

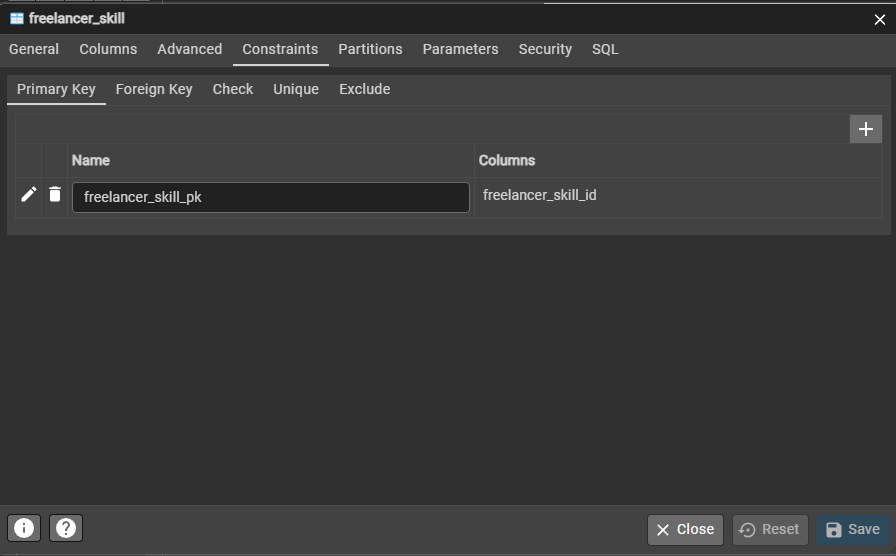
****

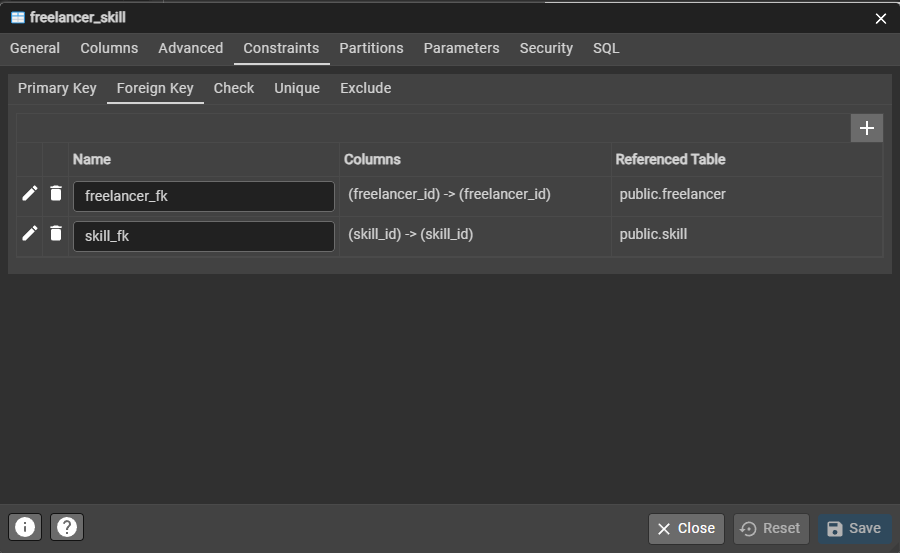


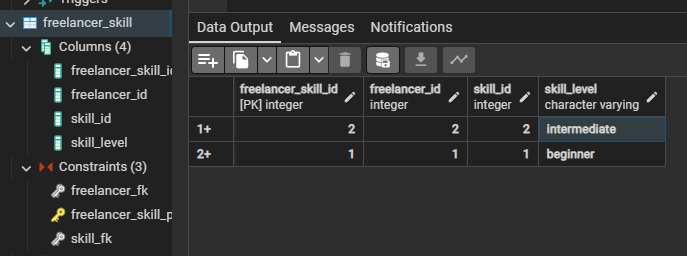


Teacher\_Subject







****

**Висновки**

У ході виконання лабораторної роботи була розроблена база даних для системи управління завданнями та проектами фрілансерів. База даних включає такі сутності: Freelancer, Project, Task, Skill, Task\_Skill, та Freelancer\_Skill. Кожна з цих сутностей має відповідні атрибути, що дозволяють зберігати і керувати інформацією про фрілансерів, проєкти, завдання та навички.

ER-діаграма бази даних була створена з використанням нотації Чена для відображення взаємозв’язків між сутностями та атрибутами. Схема бази даних була перетворена в таблиці PostgreSQL із забезпеченням нормалізації до третьої нормальної форми (3НФ). Усі таблиці відповідають вимогам НФ1, НФ2 та НФ3, що забезпечує ефективне та структуроване зберігання даних. Реалізовано зв’язки типу N:M для сутностей Skill та Task, а також Skill та Freelancer через окремі таблиці Task\_Skill та Freelancer\_Skill. Це дозволяє точно відображати багатозв’язкові відношення між завданнями, фрілансерами та їхніми навичками.

На завершальному етапі були надані скриншоти з pgAdmin4, які демонструють структуру таблиць, їхні атрибути, обмеження, а також зразки даних. Розроблена база даних відповідає поставленим вимогам і є функціональною для ефективного зберігання, обробки та управління інформацією предметної області.

**Посилання на репозиторій Github:**