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”*

Виконав:

студент ІІI курсу

групи КВ-23

Перетятько Б.В.

Перевірив:

Київ – 2024

**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**

Предметна область – це система обліку тренувань і виконання вправ користувачами. Ця база даних охоплює дані про користувачів, їхні тренування та вправи, що виконуються під час тренувань. Вона дозволяє вести облік фізичних характеристик користувачів, таких як зріст та вага, а також зберігати інформацію про кількість виконаних повторень і підходів під час тренувань. Ця система допомагає відстежувати прогрес кожного користувача і аналізувати результати його тренувань.

**Опис сутностей**

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

1. **Користувач (User)**

**Атрибути**:

* 1. user\_id: Ідентифікатор користувача.
  2. user\_firstname: Ім'я користувача.
  3. user\_lastname: Прізвище користувача.
  4. user\_weight: Вага користувача.
  5. user\_height: Зріст користувача.

**Призначення**: Збереження інформації про користувачів системи (спортсменів).

1. **Тренування (Training)**

**Атрибути**:

* 1. training\_id: Ідентифікатор тренування.
  2. user\_id: Ідентифікатор користувача, який проходить тренування.
  3. start\_date\_time: Дата і час початку тренування.
  4. end\_date\_time: Дата і час завершення тренування.

**Призначення**: Збереження даних про тренування користувачів, включаючи їх тривалість і користувача, який тренується.

1. **Вправа (Exercise)**

**Атрибути**:

* 1. exercise\_id: Ідентифікатор вправи.
  2. exercise\_name: Назва вправи.
  3. difficulty: Рівень складності вправи.
  4. description: Опис вправи.

**Призначення**: Збереження інформації про різні вправи, які можуть виконуватися під час тренування.

1. **Тренування-Вправа (Workout)**

**Атрибути**:

* 1. training\_id: Ідентифікатор тренування.
  2. exercise\_id: Ідентифікатор вправи.
  3. number\_of\_sets: Кількість підходів.
  4. number\_of\_repetitions: Кількість повторень у кожному підході.

**Призначення**: Збереження інформації про вправи, які виконувалися під час конкретного тренування, включаючи кількість підходів та повторень.

**Опис зв’язків між сутностями**

Кожен користувач може брати участь у кількох тренуваннях 1:N. Тренування прив'язане до одного користувача через поле user\_id.

Під час кожного тренування користувач може виконувати кілька різних вправ 1:N. Ці вправи фіксуються у таблиці workout, яка зберігає інформацію про те, скільки підходів і повторень було виконано для кожної вправи.

Таблиця exercises містить інформацію про всі можливі вправи, а таблиця workout пов'язує їх з конкретними тренуваннями через training\_id і exercise\_id.

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

Сутність users перетворено в таблицю users з первинним ключем user\_id та атрибутами user\_firstname, user\_lastname, user\_weight та user\_height

Сутність training перетворено в таблицю training з первинним ключем training\_id та атрибутами start\_date\_time, end\_date\_time, та зовнішнім ключем user\_id.

Сутність exercises перетворено в таблицю exercises з первинним ключем exercise\_id та атрибутами exercise\_name, difficulty, description.number\_of\_sets, number\_of\_repetitions.

Сутність workout перетворено в таблицю workout з атрибутами number\_of\_sets, number\_of\_repetitions, та зовнішніми ключами training\_id, exercise\_id.

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

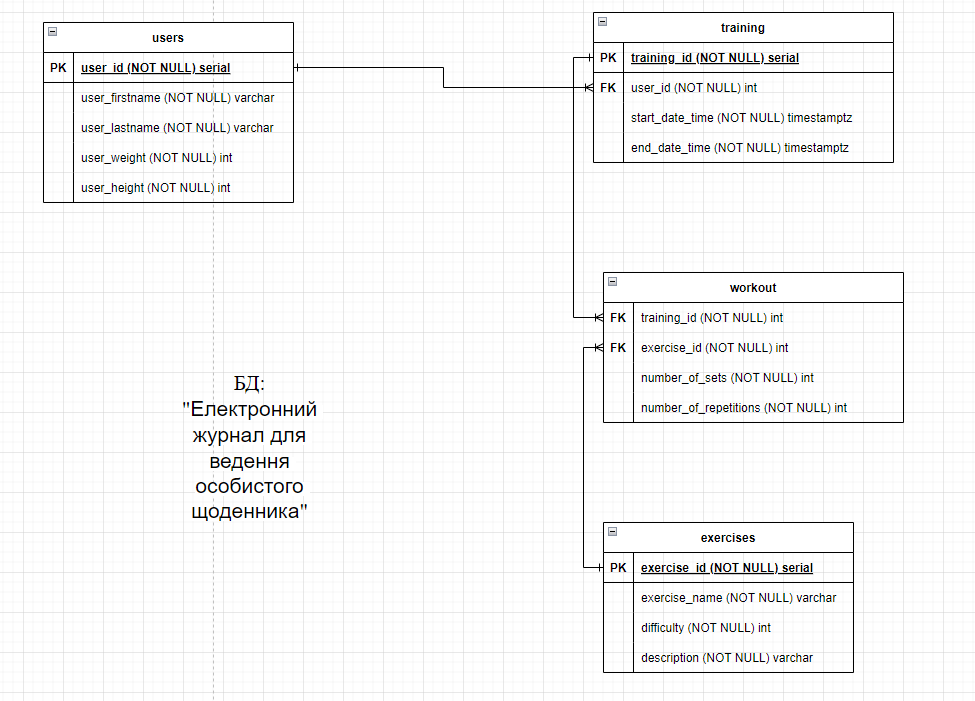


Рисунок 2 – Схема бази даних

Таблиця 1 ілюструє детальний перехід від однієї моделі до іншої.

Таблиця 1 – Опис об’єктів бази даних

|  |  |  |
| --- | --- | --- |
| Сутність | Атрибут | Тип атрибуту |
| *users* – містить дані про користувачів | *user\_id* – унікальний ідентифікатор користувача | *serial* |
| *user\_firstname* – ім’я користувача | *character varying* (рядок) |
| *user\_lastname* – прізвище користувача | *character varying* (рядок) |
| *user\_weight* – вага користувача | *integer* (число) |
| *User\_height* – зріст студента | *integer* (число) |
| *training* – містить дані про тренування | *training\_id* – унікальний ідентифікатор тренування | *serial* |
| *user\_id -* унікальний ідентифікатор користувача | *integer* (число) |
| start\_date\_time – дата і час початку тренування | *timestamptz* |
| start\_date\_time – дата і час кінця тренування | *timestamptz* |
| *exercises* – містить дані вправи | *exercise\_id* – унікальний ідентифікатор вправи | *serial* |
| *exercise\_name* – назва вправи | *character varying* (рядок) |
| *difficulty* – складність справи | *integer* (число) |
| *description –* опис вправи | *character varying* (рядок) |
| *workout* – містить дані вправи які робилися в певні тренування | *training\_id* – унікальний ідентифікатор тренування | *integer* (число) |
| *exercise\_id* – унікальний ідентифікатор вправи | *integer* (число) |
| *number\_of\_sets* – кількість підходів | *integer* (число) |
| *number\_of\_repetition* – кількість повтерень в одному підході | *integer* (число) |

**Функціональні залежності для кожної таблиці**

 **Користувач (User)**

* user\_id → {user\_firstname, user\_lastname, user\_weight, user\_height}
* Первинний ключ (user\_id) унікально ідентифікує записи в цій таблиці.

 **Тренування (Training)**

* training\_id → {user\_id, start\_date\_time, end\_date\_time}
* Первинний ключ (training\_id) визначає всі інші атрибути тренування.
* user\_id → {user\_firstname, user\_lastname, user\_weight, user\_height}
  + Ця залежність існує через зв'язок з таблицею User.

 **Вправа (Exercise)**

* exercise\_id → {exercise\_name, difficulty, description}
* Первинний ключ (exercise\_id) унікально ідентифікує записи в цій таблиці.

 **Тренування-Вправа (Workout)**

* training\_id, exercise\_id → {number\_of\_sets, number\_of\_repetitions}
* Первинний ключ (training\_id, exercise\_id) визначає всі інші атрибути тренування-вправи.
* training\_id → {user\_id, start\_date\_time, end\_date\_time}
  + Ця залежність існує через зв'язок з таблицею Training.
* exercise\_id → {exercise\_name, difficulty, description}
  + Ця залежність існує через зв'язок з таблицею Exercise.

Ці функціональні залежності вказують на те, які атрибути в кожній таблиці визначаються від інших атрибутів. Це важливо для нормалізації та управління базою даних.

**Відповідність схеми нормальним формам**

1. Щоб задовільнити умови 1НФ кожен атрибут в таблиці має бути атомарним, тобто:

* Кожна клітинка містить єдине значення;
* Кожен запис є унікальним.

Дана схема відповідає 1НФ.

1. Щоб схема відповідала 2НФ повинні виконуватись умови:

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

У даній схемі кожна таблиця має власний унікальний ідентифікатор (ключ). Кожний атрибут у кожній таблиці залежить від цього унікального ідентифікатора. Тобто, схема також в НФ2.

1. Щоб схема відповідала 3НФ повинні виконуватись умови:

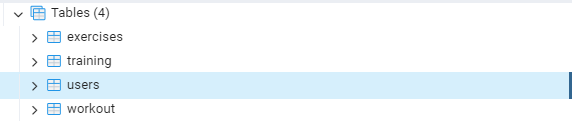
* Схема перебуває в 2НФ;
* Кожен не первинний атрибут має бути не транзитивно залежним від кожного ключа.

Оскільки дана схема в НФ2 та неключові атрибути не транзитивно залежать від інших неключових атрибутів, схема також в НФ3.

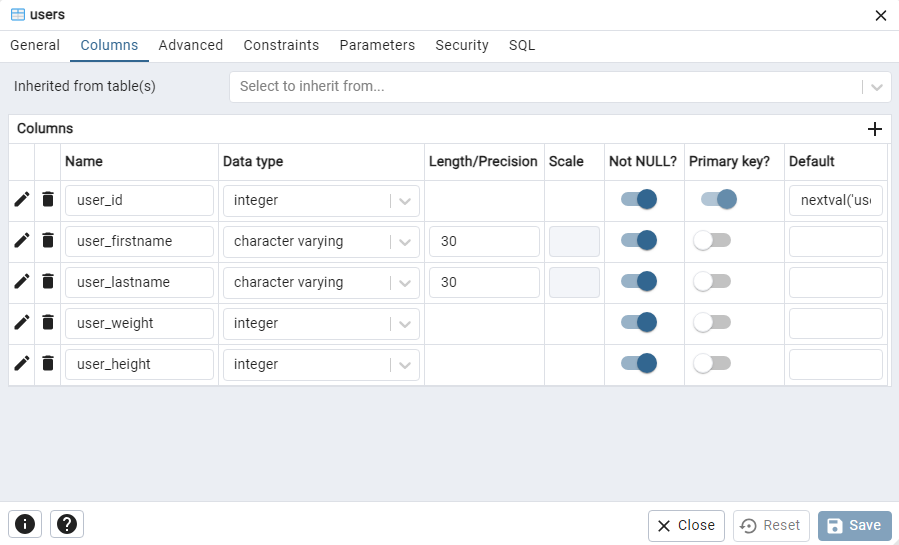
Висновок:

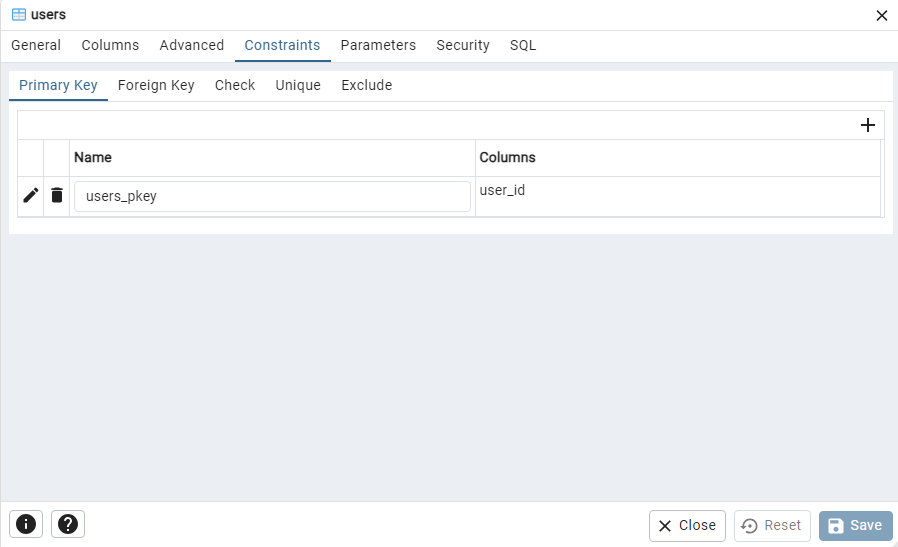
Схема бази даних відповідає нормальним формам НФ1, НФ2 та НФ3. Вона добре структурована і нормалізована, що сприяє ефективному та надійному зберіганню та обробці даних.

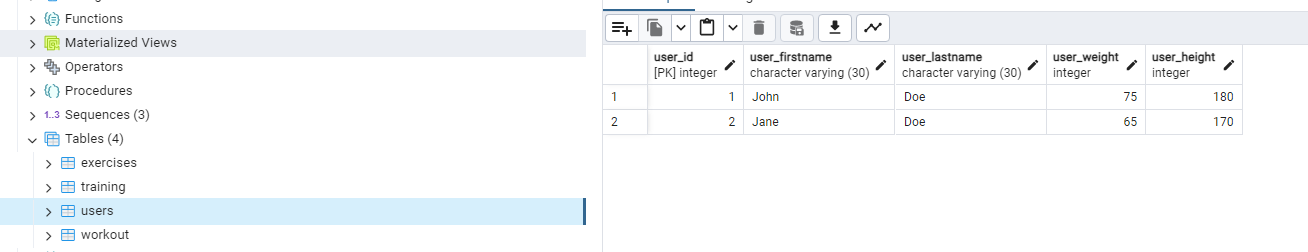
**Таблиці бази даних у pgAdmin4**

****

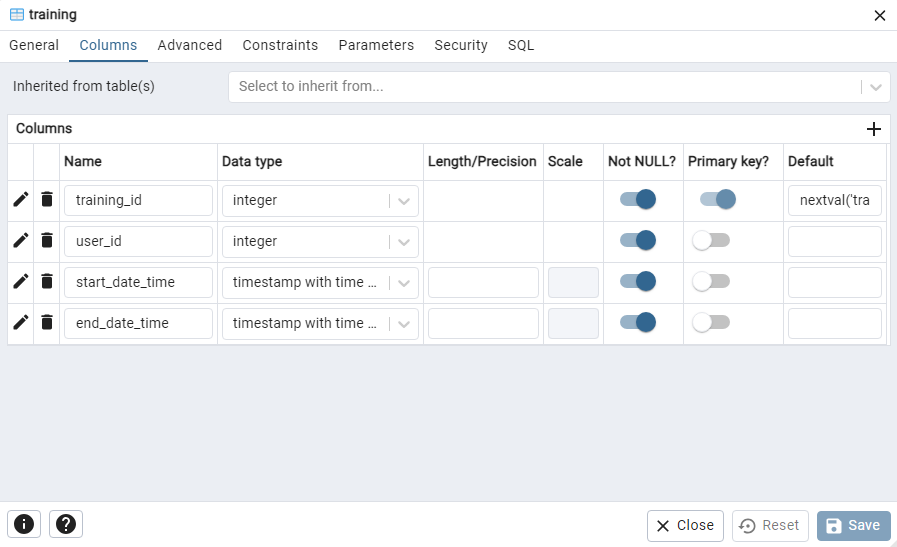
users

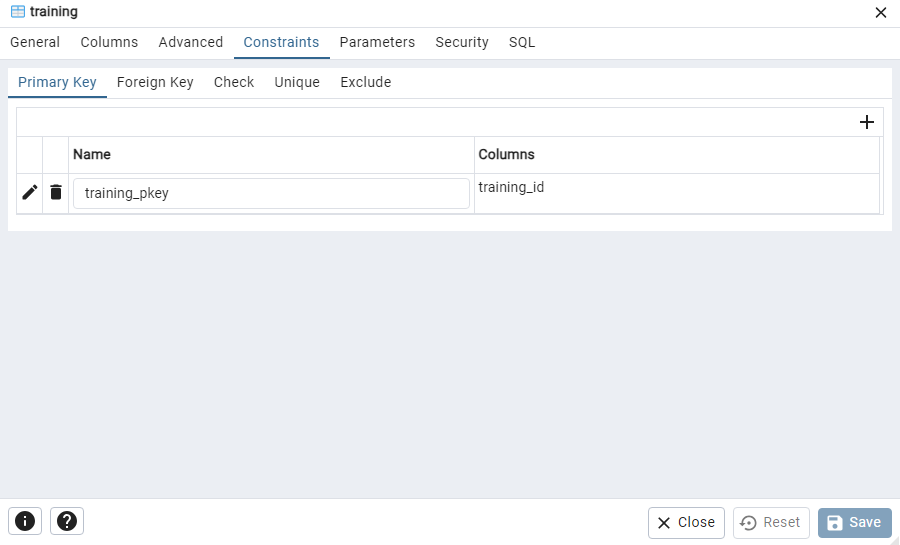
****

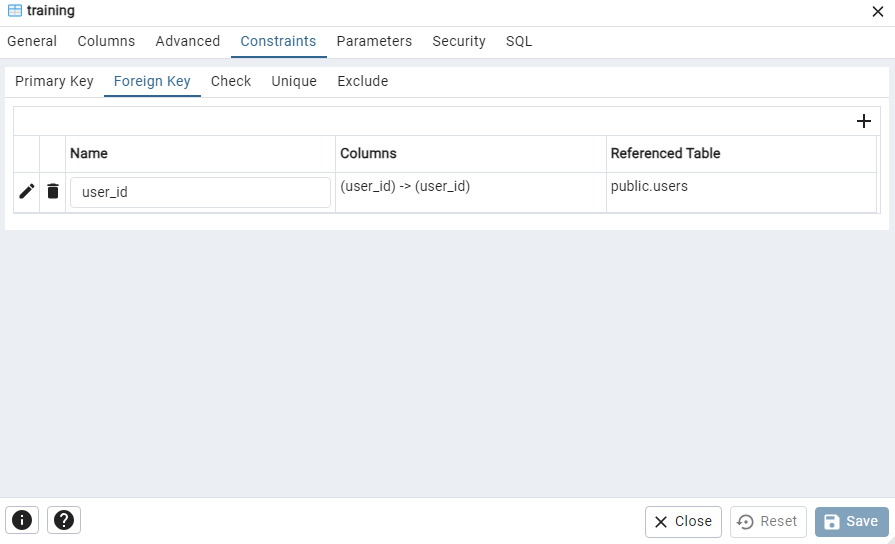
****

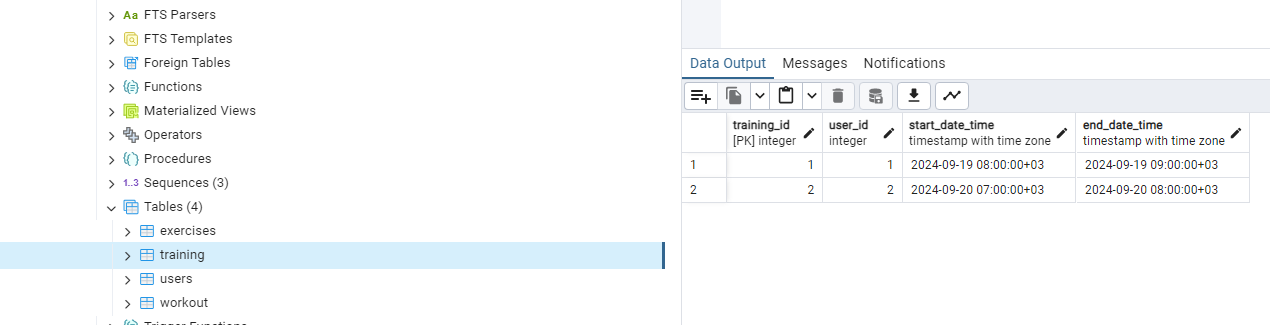


training

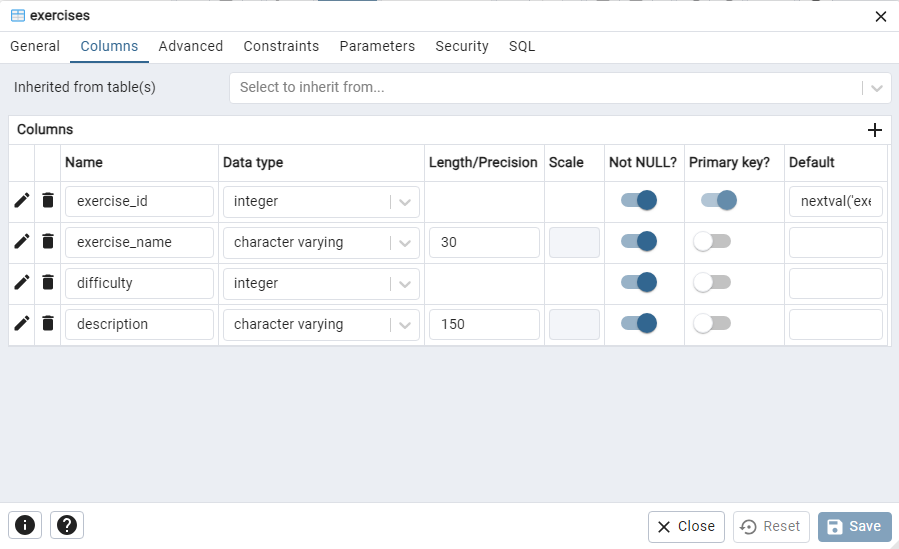
****

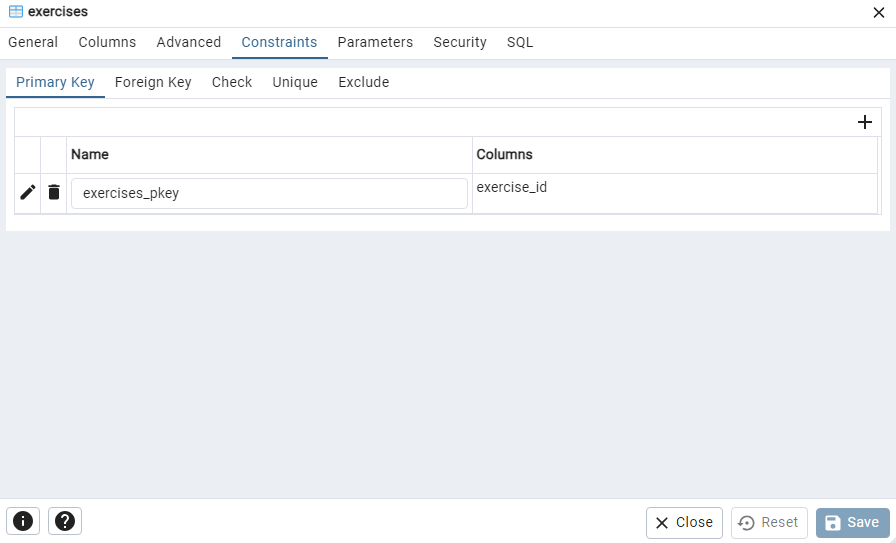
****

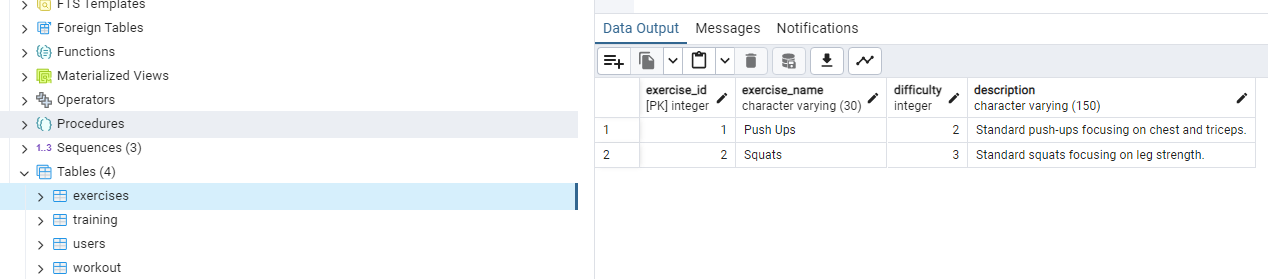
****

****

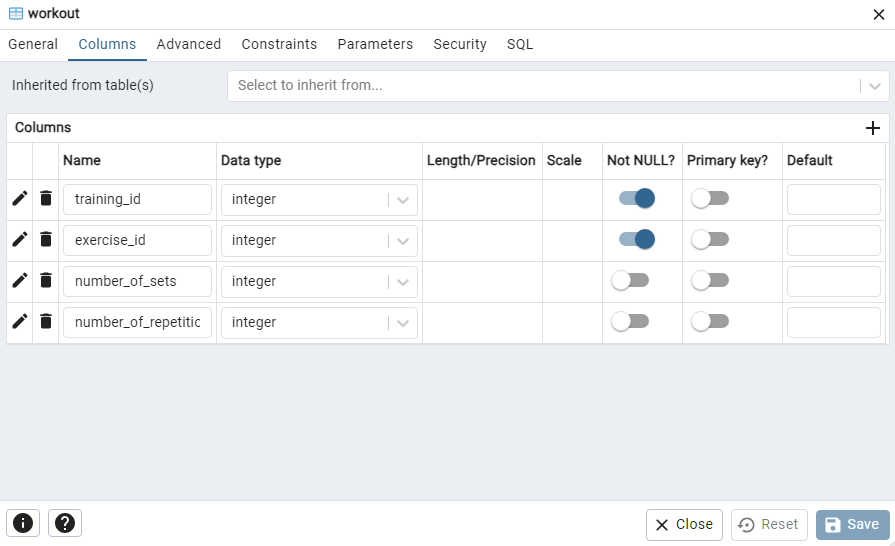
exercises

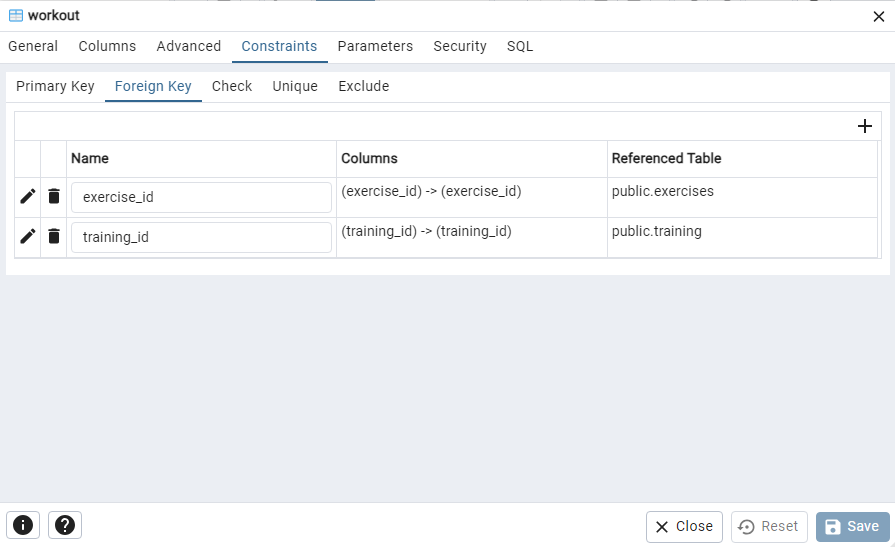
****

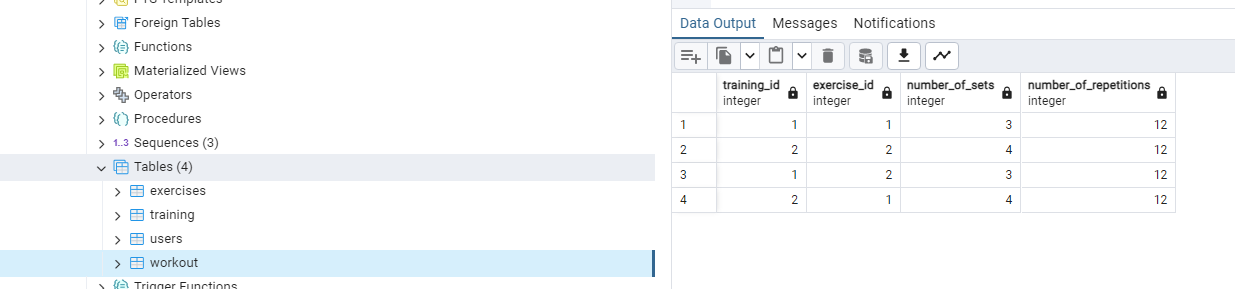
****

****

workout

****

****

****

**Таблиці в коді SQL**

CREATE TABLE public.exercises (

exercise\_id integer NOT NULL,

exercise\_name character varying(30) NOT NULL,

difficulty integer NOT NULL,

description character varying(150) NOT NULL

);

ALTER TABLE public.exercises OWNER TO postgres;

CREATE SEQUENCE public.exercises\_exercise\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER SEQUENCE public.exercises\_exercise\_id\_seq OWNER TO postgres;

ALTER SEQUENCE public.exercises\_exercise\_id\_seq OWNED BY public.exercises.exercise\_id;

CREATE TABLE public.training (

training\_id integer NOT NULL,

user\_id integer NOT NULL,

start\_date\_time timestamp with time zone NOT NULL,

end\_date\_time timestamp with time zone NOT NULL

);

ALTER TABLE public.training OWNER TO postgres;

CREATE SEQUENCE public.training\_training\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER SEQUENCE public.training\_training\_id\_seq OWNER TO postgres;

ALTER SEQUENCE public.training\_training\_id\_seq OWNED BY public.training.training\_id;

CREATE TABLE public.users (

user\_id integer NOT NULL,

user\_firstname character varying(30) NOT NULL,

user\_lastname character varying(30) NOT NULL,

user\_weight integer NOT NULL,

user\_height integer NOT NULL

);

ALTER TABLE public.users OWNER TO postgres;

CREATE SEQUENCE public.users\_user\_id\_seq

AS integer

START WITH 1

INCREMENT BY 1

NO MINVALUE

NO MAXVALUE

CACHE 1;

ALTER SEQUENCE public.users\_user\_id\_seq OWNER TO postgres;

ALTER SEQUENCE public.users\_user\_id\_seq OWNED BY public.users.user\_id;

CREATE TABLE public.workout (

training\_id integer NOT NULL,

exercise\_id integer NOT NULL,

number\_of\_sets integer,

number\_of\_repetitions integer

);

ALTER TABLE public.workout OWNER TO postgres;

ALTER TABLE ONLY public.exercises ALTER COLUMN exercise\_id SET DEFAULT nextval('public.exercises\_exercise\_id\_seq'::regclass);

ALTER TABLE ONLY public.training ALTER COLUMN training\_id SET DEFAULT nextval('public.training\_training\_id\_seq'::regclass);

ALTER TABLE ONLY public.users ALTER COLUMN user\_id SET DEFAULT nextval('public.users\_user\_id\_seq'::regclass);

ALTER TABLE ONLY public.exercises

ADD CONSTRAINT exercises\_pkey PRIMARY KEY (exercise\_id);

ALTER TABLE ONLY public.training

ADD CONSTRAINT training\_pkey PRIMARY KEY (training\_id);

ALTER TABLE ONLY public.users

ADD CONSTRAINT users\_pkey PRIMARY KEY (user\_id);

ALTER TABLE ONLY public.workout

ADD CONSTRAINT exercise\_id FOREIGN KEY (exercise\_id) REFERENCES public.exercises(exercise\_id);

ALTER TABLE ONLY public.workout

ADD CONSTRAINT training\_id FOREIGN KEY (training\_id) REFERENCES public.training(training\_id);

ALTER TABLE ONLY public.training

ADD CONSTRAINT user\_id FOREIGN KEY (user\_id) REFERENCES public.users(user\_id) NOT VALID;

**Висновок**

У ході виконання лабораторної роботи була розроблена база даних для системи відвідуваності тренувань користувачами. База даних включає такі сутності: "users", "training", "exercises", "workout". Кожна з цих сутностей має відповідні атрибути, які дозволяють зберігати та керувати інформацією про користувачів, тренування, вправи та вправи які робилися в певні тренування.

Схема бази даних пройшла аналіз на відповідність нормальним формам НФ1, НФ2 та НФ3. Всі таблиці відповідають цим нормальним формам, що гарантує надійне та ефективне зберігання даних.

На останньому етапі було надано копії екранів з pgAdmin4, що демонструють властивості стовпців та обмеження, а також вміст таблиць бази даних у PostgreSQL.

У цілому, розроблена база даних відповідає поставленим завданням та вимогам.

ПОСИЛАННЯ НА ГІТ ХАБ <https://github.com/4k3rm4n/DB-KPI>